



EDITED BY

Chiara Renso

Stefano Spaccapietra

Esteban Zimányi

MOBILITY DATA

Modeling, Management,
and Understanding

www.allitebooks.com

CAMBRIDGE

Mobility Data

Mobility of people and goods is essential in the global economy. The ability to track the routes and patterns associated with this mobility offers unprecedented opportunities for developing new, smarter applications in different domains. Much of the current research is devoted to developing concepts, models, and tools to comprehend mobility data and make them manageable for these applications.

This book surveys the myriad facets of mobility data, from spatio-temporal data modeling, to data aggregation and warehousing, to data analysis, with a specific focus on monitoring people in motion (drivers, airplane passengers, crowds, and even animals in the wild). Written by a renowned group of worldwide experts, it presents a consistent framework that facilitates understanding of all these different facets, from basic definitions to state-of-the-art concepts and techniques, offering both researchers and professionals a thorough understanding of the applications and opportunities made possible by the development of mobility data.

Chiara Renso is a permanent researcher at the Institute of Information Science and Technologies at the Italian National Research Council, Italy. Her research interests are related to spatio-temporal data mining, reasoning, data mining query languages, semantic data mining, and trajectory data mining.

Stefano Spaccapietra is an honorary professor at the School of Computer and Communication Sciences, Swiss Federal Institute of Technology, in Lausanne, Switzerland, where he has been chairing the database laboratory for more than twenty years. Together with Christine Parent, he developed MADS, a conceptual spatio-temporal data model equipped with multi-representation support, which gained worldwide renown and has been used in several applications.

Esteban Zimányi is a professor and director of the Department of Computer & Decision Engineering of the Université Libre de Bruxelles. His current research interests include business intelligence, geographic information systems, spatio-temporal databases, data warehouses, and Semantic Web. He has co-authored two books and co-edited four books in the domains of spatio-temporal modeling, spatio-temporal data warehouses, and business intelligence.

Mobility Data

Modeling, Management, and Understanding

Edited by

CHIARA RENSO

ISTI Institute of National Research Council Italy

STEFANO SPACCAPIETRA

Ecole Polytechnique Federale de Lausanne

ESTEBAN ZIMÁNYI

Université Libre de Bruxelles



CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

32 Avenue of the Americas, New York, NY 10013-2473, USA

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning,
and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9781107021716

© Cambridge University Press 2013

This publication is in copyright. Subject to statutory exception
and to the provisions of relevant collective licensing agreements,
no reproduction of any part may take place without the written
permission of Cambridge University Press.

First published 2013

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication data

Mobility data : modeling, management, and understanding / [edited by] Chiara Renso, ISTI Institute
of National Research Council (CNR), Pisa, Italy, Stefano Spaccapietra, Ecole Polytechnique Federale
de Lausanne, Switzerland, Esteban Zimányi, Université Libre de Bruxelles, Belgium.

pages cm

Includes bibliographical references and index.

ISBN 978-1-107-02171-6 (hardback)

1. Mobile computing. I. Renso, Chiara, 1968– editor of compilation. II. Spaccapietra, S., editor of
compilation. III. Zimányi, Esteban, 1964– editor of compilation.

QA76.59.M725 2014

004.16–dc23 2013009544

ISBN 978-1-107-02171-6 Hardback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for
external or third-party Internet Web sites referred to in this publication and does not guarantee that
any content on such Web sites is, or will remain, accurate or appropriate.

CONTENTS

List of Contributors	page xi
Preface	xv
Acknowledgments	xix

PART I MOBILITY DATA MODELING AND REPRESENTATION

1 Trajectories and Their Representations	3
<i>S. Spaccapietra, C. Parent, L. Spinsanti</i>	
1.1 Introduction	3
1.2 Trajectory: Definition and Application Scenario	5
1.3 From Raw Trajectories to Semantic Trajectories	10
1.4 Trajectory Patterns and Behaviors	14
1.5 Individual, Collective, and Sequence Trajectory Behavior	17
1.6 Conclusions	20
1.7 Bibliographic Notes	21
2 Trajectory Collection and Reconstruction	23
<i>G. Marketos, M.L. Damiani, N. Pelekis, Y. Theodoridis, Z. Yan</i>	
2.1 Introduction	23
2.2 Tracking Trajectory Data	24
2.3 Handling Trajectory Data	26
2.4 Reconstructing Trajectories	30
2.5 Protecting the Privacy of Individuals' Positions	34
2.6 Conclusions	40
2.7 Bibliographic Notes	40

3 Trajectory Databases	42
<i>R.H. Güting, T. Behr, C. Düntgen</i>	
3.1 Introduction	42
3.2 Data Model and Query Language	45
3.3 SECONDO	51
3.4 Representations for Sets of Trajectories	56
3.5 Indexing	58
3.6 Hermes	59
3.7 Conclusions	60
3.8 Bibliographic Notes	60
4 Trajectory Data Warehouses	62
<i>A.A. Vaisman, E. Zimányi</i>	
4.1 Introduction	62
4.2 Data Warehousing	63
4.3 Running Example	65
4.4 Querying Trajectory Data Warehouses	68
4.5 Continuous Fields	73
4.6 An Example Trajectory DW: GeoPKDD	76
4.7 Conclusions	81
4.8 Bibliographic Notes	81
5 Mobility and Uncertainty	83
<i>C. Silvestri, A.A. Vaisman</i>	
5.1 Introduction	83
5.2 Causes of Uncertainty in Mobility Data	85
5.3 Uncertainty Models for Spatio-Temporal Data	88
5.4 Conclusions	100
5.5 Bibliographic Notes	100
PART II MOBILITY DATA UNDERSTANDING	
6 Mobility Data Mining	105
<i>M. Nanni</i>	
6.1 Introduction	105
6.2 Local Trajectory Patterns/Behaviors	107
6.3 Global Trajectory Models	113
6.4 Conclusions	123
6.5 Bibliographic Notes	125

7 Understanding Human Mobility Using Mobility Data Mining	127
<i>C. Renso, R. Trasarti</i>	
7.1 The Mobility Knowledge Discovery Process	127
7.2 The M-Atlas System	129
7.3 Finding Behavior from Trajectory Data	140
7.4 Conclusions	146
7.5 Bibliographic Notes	147
8 Visual Analytics of Movement: A Rich Palette of Techniques to Enable Understanding	149
<i>N. Andrienko, G. Andrienko</i>	
8.1 Introduction	149
8.2 Looking at Trajectories	150
8.3 Looking inside Trajectories: Attributes, Events, and Patterns	156
8.4 Bird's Eye on Movement: Generalization and Aggregation	159
8.5 Investigation of Movement in Context	167
8.6 Conclusions	171
8.7 Bibliographic Notes	171
9 Mobility Data and Privacy	174
<i>F. Giannotti, A. Monreale, D. Pedreschi</i>	
9.1 Introduction	174
9.2 Basic Concepts for Data Privacy	175
9.3 Privacy in Offline Mobility Data Analysis	178
9.4 Privacy by Design in Data Mining	184
9.5 Conclusions	192
9.6 Bibliographic Notes	193
PART III MOBILITY APPLICATIONS	
10 Car Traffic Monitoring	197
<i>D. Janssens, M. Nanni, S. Rinzivillo</i>	
10.1 Traffic Modeling and Transportation Science	197
10.2 Data-Driven Traffic Models	199
10.3 Data Understanding	201
10.4 Analysis of Movement Behavior	206
10.5 Conclusions	219
10.6 Bibliographic Notes	220

11 Maritime Monitoring	221
<i>T. Devogele, L. Etienne, C. Ray</i>	
11.1 Maritime Context	221
11.2 A Monitoring System Based on Data-Mining Processes	226
11.3 Conclusions	238
11.4 Bibliographic Notes	239
12 Air Traffic Analysis	240
<i>C. Hurter, G. Andrienko, N. Andrienko, R.H. Güting, M. Sakr</i>	
12.1 Introduction	240
12.2 Motivation	241
12.3 Data Set Description	243
12.4 Direct Manipulation of Trajectories	244
12.5 Event Extraction	249
12.6 Complex Pattern Extraction Using a Moving Object Database System	253
12.7 Conclusions	257
12.8 Bibliographic Notes	258
13 Animal Movement	259
<i>S. Focardi, F. Cagnacci</i>	
13.1 Introduction	259
13.2 The Study of Animal Movement	264
13.3 Conclusions	274
13.4 Bibliographic Notes	275
14 Person Monitoring with Bluetooth Tracking	277
<i>M. Versichele, T. Neutens, N. Van de Weghe</i>	
14.1 The Difficult Nature of Measuring Human Mobility	277
14.2 How Bluetooth Offers an Alternative Solution	279
14.3 Case Studies	281
14.4 Conclusions	292
14.5 Bibliographic Notes	293
PART IV FUTURE CHALLENGES AND CONCLUSIONS	
15 A Complexity Science Perspective on Human Mobility	297
<i>F. Giannotti, L. Pappalardo, D. Pedreschi, D. Wang</i>	
15.1 Models of Human Mobility	298
15.2 Social Networks and Human Mobility	306
15.3 Conclusions	312
15.4 Bibliographic Notes	313

16 Mobility and Geo-Social Networks	315
<i>L. Spinsanti, M. Berlingerio, L. Pappalardo</i>	
16.1 Introduction	315
16.2 Geo-Social Data and Mobility	317
16.3 Trajectory from Geo-Social Web	321
16.4 Geographic Information in Geo-Social Web	322
16.5 Open Issues	330
16.6 Conclusions	332
16.7 Bibliographic Notes	332
17 Conclusions	334
<i>C. Renso, S. Spaccapietra, E. Zimányi</i>	
Bibliography	341
Glossary	351
Author Index	357
Subject Index	358

Color plates section appears between pages 258 and 259

CONTRIBUTORS

Gennady Andrienko Fraunhofer Institute IAIS, Schloss Birlinghoven, Sankt-Augustin, Germany

Natalia Andrienko Fraunhofer Institute IAIS, Schloss Birlinghoven, Sankt-Augustin, Germany

Thomas Behr FernUniversität in Hagen, Hagen, Germany

Michele Berlingerio IBM Technology Campus (Building 3), Damastown Industrial Estate, Dublin, Ireland

Francesca Cagnacci Department of Biodiversity and Molecular Ecology, Research and Innovation Centre, Fondazione Edmund Mach, Trento, Italy

Maria Luisa Damiani Dipartimento di Informatica, Università degli Studi di Milano, Milano, Italy

Thomas Devoгле Laboratoire d'informatique, Université François Rabelais de Tours, Blois, France

Christian Düntgen FernUniversität in Hagen, Hagen, Germany

Laurent Etienne Department of Industrial Engineering, Dalhousie University, Halifax, NS, Canada

Stefano Focardi Istituto Dei Sistemi Complessi, ISC-CNR, Sesto Fiorentino, Italy

Fosca Giannotti KDD Lab, ISTI-CNR, Pisa, Italy

Ralf Hartmut Güting FernUniversität in Hagen, Hagen, Germany

Christophe Hurter Ecole Nationale de l'Aviation Civile (ENAC), Toulouse Cedex, France

Davy Janssens Transportation Research Institute (IMOB), Hasselt University, Diepenbeek, Belgium

Gerasimos Marketos Department of Informatics, University of Piraeus, Piraeus, Greece

Anna Monreale Computer Science Department, University of Pisa, Pisa, Italy

Mirco Nanni KDD Lab, ISTI-CNR, Pisa, Italy

Tijs Neutens Department of Geography, Ghent University, Gent, Belgium

Luca Pappalardo KDD Lab, ISTI-CNR, Pisa, Italy

Christine Parent ISI-HEC, Université de Lausanne, Lausanne, Switzerland

Dino Pedreschi KDD-Lab, Computer Science Department, University of Pisa, Pisa, Italy

Nikos Pelekis Department of Statistics and Insurance Science, University of Piraeus, Piraeus, Greece

Cyril Ray Naval Academy Research Institute, Brest Cedex 9, France

Chiara Renso KDD Lab, ISTI-CNR, Pisa, Italy

Salvatore Rinzivillo KDD Lab, ISTI-CNR, Pisa, Italy

Mahmoud Sakr FernUniversität in Hagen, Hagen, Germany

Claudio Silvestri Università Ca' Foscari di Venezia, Mestre (VE), Italy

Stefano Spaccapietra Faculté IC, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Laura Spinsanti JRC-TP262, Ispra (VA), Italy

Yannis Theodoridis Department of Informatics, University of Piraeus, Piraeus, Greece

Roberto Trasarti KDD Lab, ISTI-CNR, Pisa, Italy

Alejandro A. Vaisman Department of Computer and Decision Engineering (CoDE), Université Libre de Bruxelles, Bruxelles, Belgium

Nico Van de Weghe Department of Geography, Ghent University, Gent, Belgium

Mathias Versichele Department of Geography, Ghent University, Gent, Belgium

Dashun Wang Center for Complex Network Research, Department of Physics,
Northeastern University, Boston, MA, USA

Zhixian Yan Samsung R&D Center, San Jose, CA, USA

Esteban Zimányi Department of Computer and Decision Engineering (CoDE),
Université Libre de Bruxelles, Bruxelles, Belgium

PREFACE

From the invention of the wheel to moon-landing rockets, technological progress over thousands of years has produced increasingly powerful and efficient transportation means, thus making moving easier and easier. Most recent progress in telecommunications has added new facets to mobility. We have now the ability to automatically keep track of our travel routes and even document them with information such as photos about the places we have been. This prompted the surge of small to huge databases holding mobility data, that is, the data about where and when we have been all over the world as well as during our daily trips to reach our workplace. Complementarily, more and more applications in a great variety of domains have been or are being developed to make intelligent use of mobility data.

While most of us are aware that our cellphones and cars equipped with a GPS facility do regularly generate signals conveying their geographical position (plus other data characterizing movement, e.g., acceleration and instant speed), not everybody is aware of what may happen later to this data, that is, how it can be used, by whom, and for what purpose. This book aims to introduce the potential answers to this question. The presentation of the material aims to make the book an easy read for all professionals (students included) in computer sciences and geoinformatics. Special attention has been given to show enough examples to optimize the understanding of the discussions. Moreover, application-oriented chapters have been included to illustrate a number of existing application domains that already benefit from using mobility data. All topics are covered to the level of detail that is compatible with a reasonable length of the book.

While the ultimate goal in mobility data processing is to solve high-level issues such as understanding how, when, where, and ultimately why objects (including persons and animals) move, elaborating the answer to these questions relies on a complex, multistep process, where the data sent by the data acquisition

device (e.g., a GPS/GSM device) are analyzed and transformed to be gradually turned into something readily meaningful for the targeted application. This process is sometimes referred to as the *Knowledge Discovery* (KD) process: from raw data to knowledge.

This book first offers an overview of the KD process as applied to mobility data. Each chapter from 1 to 8 discusses one of the issues involved.

Chapter 1 introduces the reader to the basic concepts and terms to deal with mobility data. Namely, the concept of trajectory is defined together with the various ways to approach this fundamental concept. Chapter 2 explains the most important techniques that can be used to collect the raw data from the acquisition devices and transform, homogenize, and prepare it for efficient use by applications, consistently with the application requirements. This includes potential modification of the raw data (e.g., anonymization and obfuscation) to meet privacy requirements. Chapter 3 focuses on how to store the mobility data in a database so that users can benefit from the existing know-how in database management. This is extremely important for this data to become operational with no delay. Chapter 4 similarly investigates the issues for storing mobility data in a data warehouse, opening to its use for decision-making applications interested in aggregated levels of knowledge rather than the detailed level of individual trajectories. Chapter 5 is specifically devoted to addressing the uncertainty issues that are inherent to mobility data, given that position measurements are affected by observational error and thus not necessarily as precise as applications would like them to be. The chapter closes the review of the basic data processing techniques needed for mobility data management.

With Chapter 6 the reader fully enters into the core of the knowledge management process (Part II of the book), that is, how to analyze the collected data to find its aggregated characteristics that can be of interest to the applications at hand. Movement patterns or trajectory behaviors are the core concern of the chapter. However, the lack of semantics of the extracted patterns makes the interpretation task far from obvious. To solve the mismatch following, Chapter 7 introduces the semantic dimension, thus closing the gap between the application quest for mobility information and the knowledge extracted from the data. The identification of semantic behaviors of the moving objects holds the final result of the KD process. Chapter 7 also presents a system, M-Atlas, which supports the whole mobility knowledge discovery process. Chapter 8 closes the knowledge extraction part of the book by showing through many illustrations how visualization of mobility data can be a very effective analysis tool to detect trends as well as singularities.

The last chapter in Part II of the book, Chapter 9, addresses the privacy issue, that is, how to ensure that mobility data do not violate the privacy regulations and constraints that aim at protecting individuals from the undue disclosure of personal data. This represents a very important concern as mobility data related

to moving persons can reveal details of the person's life that nobody wants to see exposed to public view. Moreover, as users of cellphones and computers we have very little control over what happens to the data that these electronic systems collect, most frequently without making us aware of the hidden data collection routines.

Part III of the book details a number of application examples that show the reader concrete uses of mobility data in a variety of domains. This part starts with the most frequently quoted application domain: car traffic. Obviously the popularity of this application domain is due to the relative ease of getting massive volumes of data from the GPS-equipped cars that have become available in recent years. Chapter 10 shows traffic application results from a variety of data repositories.

Chapter 11 is also devoted to traffic analyses, but its moving objects are boats and the moving space is the sea. This leads to a context that is quite different from cars moving in a city, as navigation rules and paths for boats are different from those of cars. The environmental data are quite different: cities show plenty of landmarks to which a human trajectory can be linked, and the same landmark (e.g., a commercial centre) can host a multiplicity of facilities that can be targeted by a moving person. Instead, the destination of a boat can usually be recognized without ambiguity, while its path is not arbitrary and has to avoid potentially hidden obstacles.

Chapter 12 closes the analysis of transportation means showing an air traffic control application, in which a variety of data sources, for example, meteorological data, have to be combined with trajectories of planes with very strong security constraints. The interesting feature of this application domain is its use of visualization tools that play an essential role in facilitating faster decision making.

Ecology is another very popular application domain that largely benefits from the availability of movement data. Chapter 13 discusses the evolution of scientific approaches to modeling animals' movement, from the formulation of the first hypotheses to modern mathematical models supporting statistical studies. It also discusses the devices that are used today for data acquisition of animals' movement.

The next application chapter, Chapter 14, covers aspects of human movement. Human movement has several unique features, such as unconstrained routes, unpredictability and sudden changes, variety of transportation means, and a richer variety of reasons for moving than animals have. In some contexts (e.g., large pedestrian crowds), traditional means of measuring mobility will not suffice for quantitative analyses. The chapter introduces the Bluetooth tracking methodology and some of its benefits in comparison with other methodologies. Despite the coarse nature of the data, exciting analyses such as crowd size estimations, flow analysis, pattern discovery, and profiling are possible.

Part IV concludes the book with three more chapters. The aim of this part is to introduce the newest developments that call for new forms and new uses of mobility data. Chapter 15 explores how the recent developments of network sciences can be applied to enriching mobility analysis approaches. This is a recent combination of scientific domains that together can significantly enhance our ability to understand movement. The second prospective chapter, Chapter 16, explores the peculiar forms of mobility data that can be gathered thanks to the popularity of social networks. Social network data is not necessarily in terms of trajectories, yet it implicitly conveys data about the movement of people. How to intelligently extract these data and analyze them is a new and exciting challenge. At last, the concluding Chapter 17 outlines some directions for future research in view of future applications. Obviously these are just a few examples; the real potential is huge.

ACKNOWLEDGMENTS

We would like to express our deepest gratitude to the many persons without whom this book would not exist.

We are obviously grateful to all the authors for their excellent contributions, as well as their availability for and commitment to writing and rewriting the several revisions of their chapters. We know how painful it may be to develop material that well harmonizes with the material provided by other authors, in particular authors from different disciplines and with different views. As editors we tried to help but also added our own constraints and guidelines, complementing those provided by the publisher. We do appreciate the cooperation efforts of the authors. We also are pleased to further thank those authors who accepted the extra task to review the first draft of a chapter written by other authors. They are listed below as internal reviewers.

We would like to stress the special role that Christine Parent generously accepted to play during all the steps of the process of developing the book. Thanks to her contributions, from many reviews to detailed suggestions for revising several chapters, as well as the glossary and the index, the final version of this book has reached the level of quality at which we were aiming.

We also owe special thanks to those colleagues who carefully reviewed the second version of the chapters, thus providing external insight and substantial comments to help the authors improve their chapters. Our gratitude is but a minimal compensation for their hard work. They are listed below as external reviewers.

We would also like to thank the European Project FP7-FET MODAP N. 245410 (<http://www.modap.org/>) and the COST Action MOVE N. IC0903 (<http://www.move-cost.info/>) for partially supporting the work of the authors and the editors of this book.

Finally, we would like to warmly thank Lauren Cowles, from Cambridge University Press, for her continued support of this book. Since the day she came to us with the book proposal, her enthusiasm and encouragement throughout its writing helped significantly in giving us impetus to pursue our project to its end.

Internal Reviewers

- Gennady Andrienko
- Natalia Andrienko
- Christophe Claramunt
- Maria Luisa Damiani
- Thomas Devogele
- Fosca Giannotti
- Ralf Hartmut Güting
- Anna Monreale
- Mirco Nanni
- Christine Parent
- Nikos Pelekis
- Salvatore Rinzivillo
- Claudio Silvestri
- Laura Spinsanti
- Yannis Theodoridis
- Alejandro A. Vaisman
- Nico Van de Weghe

External Reviewers

- Josep Domingo-Ferrer, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
- Eric Feron, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA
- Georg Gartner, Institut für Geoinformation und Kartographie, Technischen Universität Wien, Austria
- Leticia Gómez, Instituto Tecnológico de Buenos Aires, Argentina
- Michael Goodchild, University of California, Santa Barbara, CA, USA
- Patrick Laube, Department of Geography, University of Zurich, Switzerland
- Michal May, Fraunhofer Institute for Intelligent Analysis and Informations Systems IAIS, Sankt Augustin, Germany
- Gavin McArdle, School of Computer Science and Informatics, University College Dublin, Ireland
- Rosa Meo, Dipartimento di Informatica, Università degli Studi di Torino, Italy
- Mohamed Mokbel, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA
- Alejandro Pauly, Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, USA
- Peter Smouse, Department of Ecology, Evolution & Natural Resources, Rutgers University, New Brunswick, NJ, USA
- Chaoming Song, Department of Physics, Northeastern University, Boston, MA, USA

- Kathleen Stewart, Department of Geography, The University of Iowa, Ames, IA, USA
- Jack van Wijk, Department of Mathematics and Computer Science, Eindhoven University of Technology
- Vassilios Verykios, Hellenic Open University, Greece

PART I

**MOBILITY DATA MODELING
AND REPRESENTATION**

1

Trajectories and Their Representations

Stefano Spaccapietra, Christine Parent, and Laura Spinsanti

1.1 Introduction

For a long time, applications have been using data about the positions of the moving objects they are interested in. For example, city planning applications, in particular in the transportation and traffic management domains, have been observing and monitoring traffic flows to capture their characteristics, namely their importance and localization, with the aim to build better models for traffic regulation and to identify solutions for future development of the existing road network. Sociologists have also been examining the movement of cars equipped with GPS, focusing on individual cars rather than traffic flows, to understand the habits of their drivers. In the logistics domain, applications have been monitoring the localization of the parcels during their transportation from their source locations to their destinations. These applications use the data both to be able to locate a parcel at any time and to optimize the performance of the transportation and distribution strategy. Similar concerns rule the management of data tracking airline passengers and their luggage. Ecologists have been observing animals and, whenever possible, tracking them via transmitters and satellites, mainly to understand animals' individual and group behaviors. Nowadays many enterprises are looking to extract information about their potential consumers out of the tracks left by their smartphones, electronic tablets, or access to social networks such as Flickr and Foursquare that record the geographic position of their users.

Traditionally, data about movement have been captured using static facilities, for example, sensors producing traffic flow measures or detecting an animal's presence. Data acquisition facilities changed drastically with the availability of embedded positioning devices (e.g., GPS). Traffic data, for example, can now be captured as the sequences of positioning signals transmitted by the cars' GPS all along their itineraries.

These sequences may be very long, far longer than the ideal unit of processing of the application. Often the processing unit is some segment of the movement of the object instead of the whole movement itself. For instance, for animals' study the segments may correspond to the daylight time; for employees of an enterprise the segments are defined by working hours, for example, 8 A.M.–6 P.M.; for hikers in a natural park segments may be defined as going from one camp site to another camp site. These segments of movement are nowadays called *trajectories*. They are the unit of interest in applications' processing of movement data. They are the focus of this chapter.

While movement is inherently continuous, it cannot be captured as such in computers where stored data is by definition discrete. The movement track that stores movement data consists of a discrete sequence of records (transmitted by the acquisition device or input by humans) containing the position in space and time of the moving object. Movement tracks are application independent; their precise format and content depend on the device. Movement tracks are analyzed and transformed to produce application-dependent representations of trajectories. Because applications can require very different representations of trajectories (with differences in their structure as well as differences in their content) we define in this chapter three main kinds of trajectory representations that we identified as particularly significant and useful: continuous, discrete, and segmented.

Yet trajectories are not the only way to represent movement. Other representations have been designed to suit applications that need some global view of movement, resulting from the aggregation of the data about movement of individual moving objects. For example, movement can be represented as a *field of vectors* within a given space perceived as a continuous field. The vectors aggregate data from the individual tracks to represent, for a given instant, some characteristics – usually speed and direction – of the movements at every position in space. Similarly, applications willing to globally analyze the flow of objects moving among a discrete set of points (e.g., popular places within a city) will aggregate individual movement tracks into edges between nodes of a *flow network* as described in Chapter 15 on network systems. Various representations of aggregated movements in a continuous field are presented in Chapter 8. In this chapter we deal with trajectories only.

Furthermore, movement data is inherently uncertain, because of imprecision of the data sensing and data transmission devices, or because of human inaccuracy and data entry errors if a position is manually acquired. This chapter does not address this issue, but Chapter 5 discusses uncertainty issues and approaches in detail.

Application users rarely reason about locations expressed as geographical coordinates: “I am at the Eiffel Tower” is easier to understand than “I am at 48°51'29" North, 2°17'40" East.” To enable easier and richer use of movement

data, recent research has been investigating ways to reformulate and enrich movement data to make them better correspond to application requirements and scenarios. This is done by adding to the movement data contextual data that describe where the object moved (e.g., the roads it followed, the places where it stopped), when (e.g., during which time period, during which event), how (e.g., using which transportation means), what for (e.g., which activity it performed when it stopped). Enriched movement tracks are nowadays referred to as *semantic trajectories*. Chapters 6 and 7 in this book discuss how to build and use semantic trajectories.

This initial chapter introduces the reader to a global understanding of the trajectory domain. It spans from raw data to data transformation and enrichment, to end up with the analysis tasks needed to fulfill application requirements. The chapter covers both the static representation of the domain (what a trajectory is and how it can be represented) and its behavioral aspects (how to understand and characterize mobility in terms of why things move, what they do while moving, which are meaningful movement sequences, etc.). Given the diversity of application requirements, several representations of trajectories are considered. Basic concepts and terminology are defined, explained, and documented via examples.

1.2 Trajectory: Definition and Application Scenario

Mobility is a recent domain where people use diverse terminologies and concepts, without much consensus on choices and definitions. To limit misunderstanding and confusion, this section defines a set of concepts and vocabulary that together form a consistent framework for discussing trajectories and their analysis as understood in this book.

At the source of our movement data processing concerns there is a moving object, that is, an object that can over time change its position in space (its spatial coordinates). In this book, we don't address deformation issues raised when considering moving objects, such as hurricanes and oil spills, that span over a changing area or volume. We focus instead on moving objects represented as points. Keeping movement data about a moving object consists in keeping the history of its successive positions, that is, creating a record that holds, for this object, all past, present, and sometimes future positions and the associated instants. We will not discuss future positioning at this point, and call this record the movement track of the object. The sequence can be unbounded. The time intervals between successive positions may have the same duration or different durations.

Definition 1.1. The *movement track* of a moving object is the temporally ordered sequence of spatio-temporal position records captured by a positioning device

during the whole lifespan of the object. Each record (instant, point, features) contains the instant of the capture, the 2D or 3D point of the object, and possibly other features captured by the device (e.g., the instantaneous speed, acceleration, direction, and rotation). There are no two records with the same instant value. \square

Before going into a detailed analysis of what trajectories are and how they can be tailored into useful information for the targeted applications, we informally sketch an example application scenario that uses trajectories to describe the movement of tourists visiting Paris.

1.2.1 Tourists Application Scenario

Tourism represents an important source of revenue for many countries, regions, and cities. Its promotion has become a critical business. The efficiency of promotion activities can be boosted by the acquisition of knowledge about the habits of tourists, their preferences, and the local features that are likely to attract them in large numbers. Part of this knowledge can nowadays be extracted from the analysis of on-site movements of tourists, collected via their smartphones equipped with GPS and connected to social networks.

From a promoter's viewpoint, a tourist destination is a geographical area that offers tourists the opportunity of visiting a variety of places (e.g., museums, parks, monuments, and attractions) while using many services (e.g., restaurants, accommodations, shops, and travel agencies). All these tourist places and services are collectively referred to as points of interest (POIs), chosen from a tourist perspective. A tourist day consists in moving from one POI to another one, and so on, while stopping for some time in each one of the visited POIs for eating, resting, shopping, visiting, sleeping, attending a show, or meeting other people, as shown in Figure 1.1.

The oriented line in Figure 1.1 shows the spatial route of the trajectory made by a tourist during one day while visiting Paris. Very often, applications use only this spatial representation of movement on a background map. It is very intuitive, yet it provides very little temporal information. Time is only implicitly conveyed by the fact that the sequence of points forming the line is a temporally ordered sequence. In other words, going further down the line (from its beginning to its end) corresponds to moving later in time. In Figure 1.2, part of this trajectory is shown with a volume (x, y, t) visualization. The trajectory is represented by the thick line in the upper part of the figure, and its projection on the (x, y) plane shows its spatial route as a line lying on the map. As time never stops and always flows on, no two points can have the same time value, and the thick 3D line always moves further on the time axis. When a moving object stops, its position in the (x, y) plan does not change. In the (x, y, t)

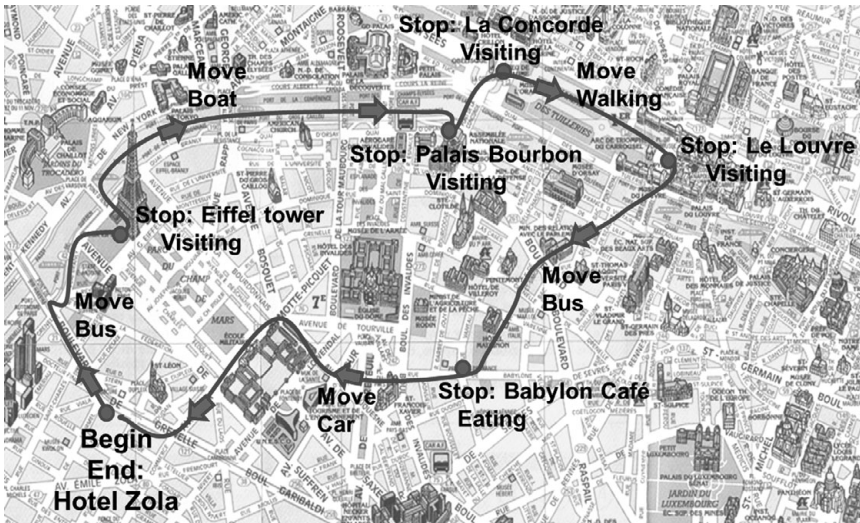


Figure 1.1 A daily trip of a tourist in Paris, visiting several tourist attractions.

visualization an object stopping results in a vertical segment whose length corresponds to the duration of the stop. Figure 1.2 shows three vertical segments that represent stops at Place de la Concorde, Le Louvre museum, and the Babylon café.

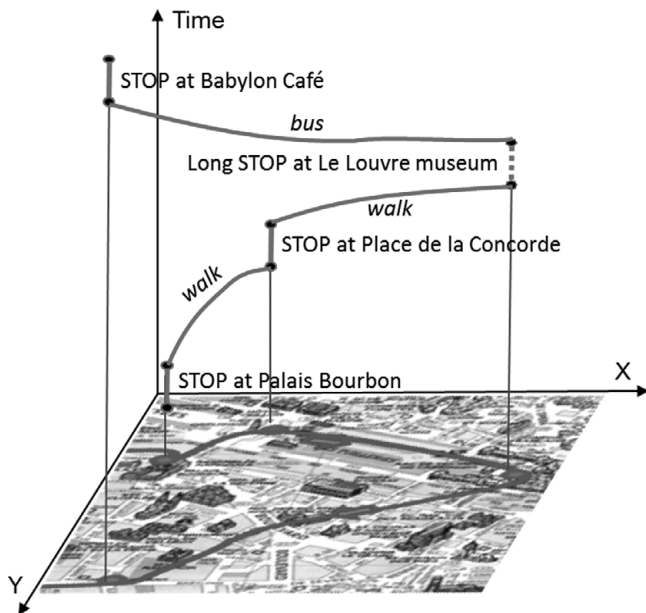


Figure 1.2 A volumetric representation of part of the tourist's daily trip of Figure 1.1.

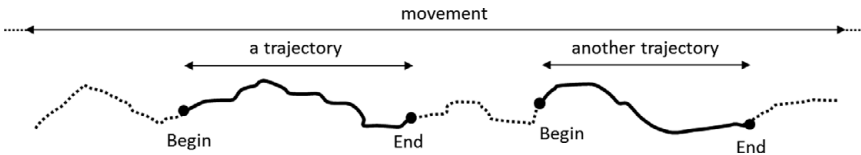


Figure 1.3 Two trajectories extracted from the movement of an object.

Collecting information on daily travels of tourists enables extracting knowledge on their favorite places, in which order places are visited, how much time tourists spend at each attraction, etc. This can be used to tune the facilities to better match tourist expectations and regulate the flow of tourists to avoid large waiting lines. It can also be used to build tourist profiles, propose personalized tours and services, and suggest to tourists on the move their next preferred destination. Similar kinds of moving persons’ scenarios are used in many research papers to illustrate various kinds of analysis. We will use it throughout this chapter for illustrating the concepts.

1.2.2 Trajectory Definition

As stated in Section 1.1, while some applications keep and analyze whole movement tracks, many other applications are interested in specific segments of the movement. We call trajectories the segments of the object’s movement that are of interest for a given application. Obviously the whole movement is a particular case of trajectory.

Definition 1.2. A *trajectory* is the part of the movement of an object that is delimited by a given time interval $[t_{\text{Begin}}, t_{\text{End}}]$. It is a continuous function from the time interval $[t_{\text{Begin}}, t_{\text{End}}]$ to Space. The spatio-temporal position of the object at t_{Begin} (resp. t_{End}) is called the *Begin* (resp. *End*) of the trajectory. \square

Figure 1.3 shows (as a dotted line) a section of the movement of an object and, superimposed as continuous lines, two segments identified as relevant trajectories.

The criterion to identify trajectories within movement is application dependent. For instance, in the tourists scenario, to globally analyze the activities performed by a tourist during his/her stay in Paris, the whole track left by the tourist will generate a single trajectory (spatial criterion “inside Paris”). On the other hand, in order to analyze what tourists do in one day in Paris (whatever the length of their stay), or what they do on specific days (e.g., on Sundays, on December 25), each daily track of each tourist in Paris will generate a separate trajectory as in Figure 1.1.

In the real world, time, movements, and trajectories are continuous, but in the digital world, where applications are implemented, we can only store discrete

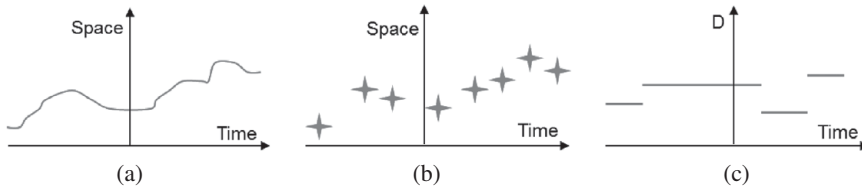


Figure 1.4 The three kinds of representations of movement: continuous, discrete, and stepwise.

implementations, such as the movement track. In order to satisfy applications that need a continuous view of trajectories, the discrete implementation may be enriched with interpolation functions that allow dynamically reconstructing a continuous representation of the discretized trajectory.

Definition 1.3. A *continuous representation of a trajectory* (or *continuous trajectory* in short) is a trajectory representation that describes in a continuous way the movement of the object for the time interval $[t_{\text{Begin}}, t_{\text{End}}]$ of the trajectory. It usually consists of a finite sequence of spatio-temporal positions, and the interpolation functions that enable the computation of the spatio-temporal position of the moving object for any instant in $[t_{\text{Begin}}, t_{\text{End}}]$. \square

Whenever the movement track is too sparse for inferring the original continuous movement of the object, or the applications do not need the continuous movement, the finite sequence of spatio-temporal positions is used as a discrete representation of a trajectory. Currently this is the case, for example, of the applications that use the movement tracks generated by social networks (see Chapter 16).

Definition 1.4. A *discrete representation of a trajectory* (or *discrete trajectory* in short) is a trajectory representation that is made up of the finite list of spatio-temporal positions for the time interval $[t_{\text{Begin}}, t_{\text{End}}]$ of the trajectory, but not providing the continuity of the movement of the object. \square

Figure 1.4a visualizes (as a line) a continuous representation of a trajectory. Figure 1.4b visualizes (as a set of points) a discrete representation. Figure 1.4c visualizes a stepwise (segmented) representation (see Section 1.3).

To complete the basic picture we briefly introduce two trajectory concepts, holes and semantics gaps, which address the understanding of missing points at the conceptual level. These concepts contribute to a more complete vision of trajectories. The reader has to be aware that they only play an important role in a limited number of application cases, which explains why researchers rarely take these concepts into account.

The term *missing point* denotes the existence, within a movement track, of an abnormal (longer than the sampling rate) temporal gap between two consecutive

recorded positions: the information on the movement of the object is missing. If this is accidental (e.g., because of a device malfunction) we say there is a *hole* in the track. The typical case where this still happens is when a GPS is taken through a tunnel. The connection is cut as long as the GPS doesn't get out of the tunnel. Short-duration holes may sometimes be "filled," using, for example, linear interpolation algorithms that compute the missing positions. In this case the hole disappears.

If missing points are not due to some data acquisition accident (whatever the cause), it follows that their absence is due to a decision by the application designer to interrupt data acquisition during some specific periods. For example, a company running daily tourist tours in Paris may decide to track tourists' positions during its hours of operation (say from 8 A.M. to 6 P.M.) but not during lunchtime (say from 12:30 P.M. to 2 P.M.) when tourists on a tour are free to do whatever they want. Consequently, tourists' daily trajectory tracks will be filled with positions from 8 A.M. to 12:30 P.M. and from 2 P.M. to 6 P.M., and no positions during the lunchtime break. This lunchtime break is not an accidental hole in the trajectory; we call it a *semantic gap* (its semantic in this case is that of the lunch period).

A trajectory with semantic gaps is defined for a set of disjoint time intervals instead of a unique time interval. For the sake of simplicity, in the rest of the chapter we will deal only with trajectories defined on a single interval (i.e., without semantic gaps).

1.3 From Raw Trajectories to Semantic Trajectories

The two representations of trajectories defined above come directly from the movement track. It is why they are often called *raw trajectories*. They are well fitted if, for example, the aim of the application reduces to locating some moving objects (e.g., where was Mr. Smith on the evening of June 12, 2012?) or computing statistics on the spatio-temporal characteristics of the trajectory (e.g., which percentage of daily tourist trajectories show a global speed over 7 km/h?). On the other hand, many applications need more informative results, such as those that can be computed by combining raw data with the contextual data (e.g., geo-objects and events that show a spatial or temporal relationship with the trajectory data), and with the thematic data available for the moving object itself (e.g., age, gender). These applications can reach this goal by following one of two approaches:

1. The application dynamically accesses the contextual data during its computations.
2. The application first preprocesses the trajectory representations, enriching them with contextual data and appropriate restructurings, and after that it computes its results by using the enriched trajectories.

Both Chapter 3 (trajectory databases) Chapter 8 (visual analytics for trajectories) of this book follow the first approach, with two different methods. They allow their users to analyze the trajectories in context, that is, by analyzing the relations between the moving object and the environment. On the other hand, Chapters 6 and 7 on trajectory analysis by data mining follow the second approach: The trajectories are first enriched and then mined. In contrast to raw trajectory representations, we qualify as *semantic* any trajectory representation that has been created by enriching and transforming a raw trajectory in order to add more meaning. In this chapter, we focus on the second approach and therefore explore which kinds of semantic trajectory representations are needed.

To create semantic representations of trajectories, the trajectory management system needs to have access to application contextual data, which typically includes knowledge of geographical objects (i.e., objects that have a known position in geographical space) and events in the region and time period traversed by the trajectories. For example, in the tourists scenario we will naturally assume that the knowledge about the city map is available so that, for example, trajectory paths can be described in terms of streets and crossroads and the points where people stop can be identified with places of interest, such as landmarks, significant buildings, monuments, museums, shops, restaurants, cafes, and sports centers. Information about ongoing events, for example, shows, fairs, concerts, and football games, is to be collected too, as it may influence the organization of tourist tours. We use the term *contextual data repository* to generically refer to whatever external source that can be used by the application to enrich trajectory data.

All the following kinds of information can be used together with raw trajectories in order to get semantic representations:

- The geo-objects representing the places of interest, roads, regions where the trajectory passed;
- The events related to the movement of the object;
- The transportation means used by the person for moving;
- The activities performed by the person or animal when (s)he stopped.

The geo-objects corresponding to the positions of a trajectory can be found by a process called *geo-localization*, which is a usual technique for adding semantics to spatial data. It maps spatial coordinates (x, y) to the corresponding geo-objects of the contextual data repository. For instance, in Figure 1.1, the coordinates of the spatio-temporal positions where the tourist stopped for a while have been mapped to the corresponding POIs: Eiffel tower, Palais Bourbon, Le Louvre museum, and so on. The positions where (s)he was moving have also been mapped to the corresponding street segments. Chapter 2 presents in detail this geo-localization process. Similarly, it is possible to find which events correspond


























































MOVE annotation transport means																					
STOP annotation activities																					
STOP annotation POI type																					
STOP annotation POI																					
Stop/move episodes																					
TIME line		8.30	9.00	10.00	10.30	10.45	11.00	11.15	12.00	18.30	19.00	21.00	21.30								

Figure 1.5 A Stop/Move segmented representation of the tourist’s trajectory of Figure 1.1.

to the temporal data of the raw trajectories. For instance, June 17, 2012, was Ascension Day, a public holiday in France.

All this information is conceptually associated with the spatio-temporal positions of the trajectory, but it would be space and time consuming to actually store it for each position of the trajectory. Indeed, usually one does not characterize a given position but a sequence of positions. For instance, when a tourist visits Le Louvre museum, he or she may stay there for several hours, which may mean thousands of consecutive recorded positions with the same annotation value: “Le Louvre.” Therefore a common method consists in segmenting the trajectory into maximum subsegments of spatio-temporal positions that are all associated with the same value of a given expression whose range of values is a finite set of annotation values. Each change of value of the expression signals the starting of a new segment. The segments are called *episodes* and, instead of storing the information with the position, it is stored with the episode. A common kind of segmentation is the segmentation into episodes of kind *Stop* (segments of the trajectory where the object roughly does not move) and *Move* (segments of the trajectory where the object moves). It is a generic segmentation that relies only on computation of the raw data. It is often based on the instantaneous speed of the object, but the exact expression depends upon the application. For the tourists scenario, the expression could be the Boolean expression: $\text{speed} \leq 1 \text{ km/h}$. This expression defines a Stop episode when the expression value is True, and a move episode when it is False. Chapter 2 provides more details on trajectory segmentation methods.

As shown in Figure 1.5 for the tourist’s trajectory of Figure 1.1, segmentation produces a semantic representation that is more abstract than the continuous representation it comes from. The continuous representation contains the sequence of spatio-temporal positions of the trajectory, while a segmented representation provides a semantic view of the trajectory as a sequence of episodes, each one described by a tuple (time interval, annotation value). A segmented

representation does not implement a continuous function: the moving object jumps (so to speak) from one episode and annotation value to the next. It corresponds to a step function as shown in Figure 1.4c.

According to their needs, applications may use the continuous representation, a segmented one, or both superimposed. For example, Figure 1.1 shows for a tourist's trajectory the superimposition of the continuous representation and a Stop/Move segmented representation.

Definition 1.5. A *segmented representation of a trajectory* (or *segmented trajectory* in short) is the implementation of a step function that maps the time interval $[t_{\text{Begin}}, t_{\text{End}}]$ to a finite set of values, D . Each step of the function is called an episode, and its corresponding D -value the defining annotation of the episode. \square

Practically, a segmented trajectory representation is a temporally ordered subsequence of tuples (time interval, defining annotation value, annotation values), where the time intervals are all disjointed.

Another example of segmentation of human trajectories is the transportation means. Chapter 2 shows how this information can be computed automatically by combining the raw trajectory data with the data on the public transport system and some common sense rules about transportation modes. Often a human trajectory starts with a first “walk” segment (at least to get out of the building and into the first transportation means), and this segment is followed by, say, a “metro” segment, then again a “walk” segment, and so on. In this case the segmenting expression is a procedure call whose result is the corresponding defining annotation, for example, “walk,” “metro,” “bus,” “car,” or “boat.”

A given trajectory may be structured into episodes in many different ways, that is, using different expressions. For example, the tourists' trajectories may be alternatively segmented into episodes based on (1) stops and moves, (2) the time period corresponding to the instant of the spatio-temporal position (e.g., morning, noon, afternoon, evening, night), and (3) the category of the area of the city corresponding to the location of the spatio-temporal position (e.g., residential, touristic, commercial, recreational, services, special). There is no limit to the number of episode segmentations that can be applied to a set of trajectories. Each segmentation into episodes provides a new interpretation of the trajectory that can be superimposed as needed.

Moreover, while episodes are created via their defining annotation, episodes, like every other component (especially spatio-temporal positions) of a trajectory, can be further annotated using other annotations. For instance, assuming that the tourists' trajectories have been segmented into Stop and Move episodes, the Stop episodes may be further annotated with the nearest point of interest that is the most likely to have been visited by the tourist during this stop. The Move episodes may be annotated with the transportation means. It is the case

in Figures 1.1 and 1.5: The trajectories (1) have first been segmented into Stop and Move episodes, (2) their Move episodes have been annotated with a new annotation, the main transportation means for this move segment, and (3) their Stop episodes have been annotated with two new annotations: the geo-object (POI) where the stop took place and (POI) the activity of the tourist during this stop. Figure 1.5 provides also an alternative annotation of the Stop episodes: the types of the POI associated to the stop (hotel, museum, restaurant, etc.) instead of the POI itself (the hotel Zola, the museum Le Louvre, the restaurant Babylone, etc.).

Once the best-fitting representations of the trajectories have been built, application analysts can use them to extract all kinds of statistical and higher-level knowledge useful to the application.

1.4 Trajectory Patterns and Behaviors

Section 1.3 discussed how to enrich the raw trajectories with the related contextual data to come up with semantically rich trajectories. This section discusses the concepts involved in the process that extracts relevant semantic knowledge from the trajectories. Since long ago, researchers have developed novel techniques to extract knowledge taking into account the spatio-temporal specificity of movement data. These techniques support learning from trajectories far beyond retrieving factual data about specific moving objects (e.g., where was the car 345FT92 at time t ?) and computing statistics about populations of moving objects (e.g., how many cars per hour travel this road on weekdays?).

Of vital importance for a large number of applications is the identification of the significant trends shown by a population of moving objects. Sociological studies, for example, may aim at comparing commuters' shopping habits versus shopping habits of noncommuters. Trajectory analysis reveals which persons qualify as commuters and identifies their favorite shopping places. Similarly, analysis of tourists' trajectories may detect trends in tourist behavior that provide important information to tourist agencies to optimize their offers.

A significant trend can be identified as a set of trajectory characteristics that repeatedly appear in the set of trajectories under consideration. Most frequently, trends are “found” using a knowledge extraction tool, usually applying data mining techniques. The data mining community uses the term “pattern” to denote the findings from the extraction, and “frequent pattern” to denote those patterns that appear frequently enough in the source data to be considered potentially interesting for the application at hand. For example, “The trajectory ends at the same place it began” is a trajectory characteristic that can be denoted as a *Loop* pattern. The pattern identifies trajectories whose spatial trace, as a whole, forms a loop. Its definition relies on the spatio-temporal positions Begin and End, and nothing else. We call it a *spatio-temporal pattern*.

When the trajectory analysis uses only raw trajectories and no contextual data, the analysis can produce only spatio-temporal patterns, as the Loop pattern above. Since semantic trajectories and contextual data have gained attention, several research groups have been trying to identify patterns that would have a more semantic flavor than the traditional spatio-temporal patterns. For example, a frequently discussed pattern in more recent works is the *HomeToWork* pattern that applies to the daily routine trips of people going to work. Its definition requires knowledge of the home-place and the work-place of the moving person, which is contextual knowledge that is likely to be available from public people and company repertoires or inferable from the analysis of people's trajectories. As the definition of these patterns relies on the semantic and contextual information associated with the trajectories, we refer to them as *semantic patterns*.

It is worth noting that currently there is no consensus on the terminology, and recent research tends to use either pattern or behavior, as well as behavioral pattern, without a clear distinction among the concepts. This book is no exception, and in the following chapters the reader will find the terms pattern and behavior as denoting the same concept. In this chapter we use behavior, with the following definition that covers both semantic and spatio-temporal behaviors/patterns.

Definition 1.6. A *trajectory behavior* (or *behavior*, in short) is a set of trajectory characteristics that identifies a peculiar bearing of a moving object or of a set of moving objects. The behavior is defined by a predicate that says if a given trajectory (or a given set of trajectories) shows the behavior. Synonym: *trajectory pattern*. □

Definition 1.7. A *trajectory semantic behavior* is a trajectory behavior whose defining predicate includes conditions on some semantic representation of the trajectories and/or conditions on some contextual data that are spatio-temporally related to the trajectories. Synonym: *trajectory semantic pattern*. □

Definition 1.8. A *trajectory spatio-temporal behavior* is a trajectory behavior whose defining predicate bears only on the raw representation of the trajectories, excluding any contextual data. Synonym: *trajectory spatio-temporal pattern*. □

The predicate defining a behavior can rely on any characteristic of the trajectories (e.g., spatio-temporal positions, episodes, annotations), contextual data linked to the trajectories (e.g., the type of the geo-objects linked to some episodes or some of their attribute values), spatial relationships with geo-objects (e.g., stopping near some given geo-object), temporal relationships with events (e.g., moving during some given event), and relationships with other moving objects (e.g., moving ahead of a given group of trajectories).

Considering our tourists scenario and its set of daily trajectories of persons moving in Paris with a GPS, the following behavior definition can be used to separate the trajectories of tourists from the trajectories of other persons:

Tourist behavior: A daily trajectory shows the Tourist behavior if: Its Begin point P1 is a place of kind “Accommodation,” it makes at least one stop in a place of kind “Museum” or “TouristAttraction,” it makes one stop in a place of kind “EatingPlace,” and its End point is in the same P1 place as its Begin point.

This *Tourist behavior* is a semantic behavior. An example of spatio-temporal behavior, always for the tourists scenario, is the *LongTrajectory* behavior defined as: the duration of the trajectory is greater than 14 hours. The number of behaviors that can be defined is unlimited. For example, “Going from the Place de la Concorde to the Champs Elysées” and “Going from the Place de la Concorde to Place de la Madeleine” could be semantic behaviors of interest for travel agencies organizing tourist tours in Paris. Trajectories showing these behaviors would also qualify as showing the more generic semantic behavior “Going from a tourist spot to a commercial area.”

Interesting behaviors can be inferred using various methods for extracting useful knowledge from trajectory data sets: data mining methods are discussed in Chapters 6 and 7 of this book, and visual analytics methods are discussed in Chapter 8. The most common outputs of these methods are clustering (trajectories are grouped into classes that share some common characteristics) and behaviors/patterns (describing the characteristics shared by significant groups of trajectories).

Alternatively, an application manager interested in application-specific trends can manually define behaviors a priori. Back to our tourists scenario, a large number of behaviors can be manually predefined, each one targeting the identification of a subset of the population moving in Paris: *Tourist* behavior, *OfficeWorker* behavior, *Housewife* behavior, and so on. In some smaller-scale applications the number of interesting behaviors may be small enough to be exhaustively defined. Moving object database query languages, like the one presented in Chapters 3 and 12 of this book, can be very effective for searching the trajectories that comply with a given behavior.

Researchers have defined generic families of behaviors that rely on the constancy/variation of some given characteristic of the trajectory (e.g., same direction or speed for a while) or on the similarity or correlation of the values of some characteristic of a group of trajectories (e.g., proximity for a while). For example, potentially interesting features in the shape or combinations of shapes of trajectory traces have led to the definition of a number of spatio-temporal

behaviors. Well-known examples include the *Meet*, *Convergence*, and *Flock* behaviors that are discussed in Chapters 6 and 7. Most spatio-temporal behaviors are generic behaviors, that is, they are supposed to be applicable to any application domain. Semantic behaviors tend to be application-specific as they handle the semantic aspects of trajectories and these semantic aspects are the most frequently application-specific.

Behaviors are defined for trajectories. Still, many behaviors can also qualify parts of trajectories. Most spatio-temporal behaviors that characterize the shape of the trace of the trajectory can qualify whole trajectories as well as parts of trajectories. For instance, the *Straight* behavior that characterizes a straight spatial trace can be used for defining trajectories whose whole trace is straight as well as trajectories that contain at least one straight segment whose length is longer than some given threshold. Another example is the *Flock* behavior that characterizes a group of trajectories that travel together (see Chapter 7). The common travel may last during the whole trajectories or only during some part defined by a time interval. On the other hand, behaviors that rely on some global characteristic of the trajectories (e.g., some aggregation on the whole trajectory, *Begin* and *End*) can apply only to whole trajectories. An example is the *StopMoreThanMove* behavior that characterizes *Stop/Move* segmented trajectories that spend more time during the stops than during the moves.

The number of behaviors that can be defined is unbounded, as any application domain has its own typical requirements and any application adds its specific requirements. We purposely abstain from trying to define a taxonomy of behaviors. Description of some works devoted to building such taxonomy can be found in the Bibliographic Notes section. Of particular importance in clarifying the broad vision of behaviors is the separation between individual and collective behaviors, and the *Sequence* behaviors, both discussed in the next section.

1.5 Individual, Collective, and Sequence Trajectory Behavior

A very important feature of behaviors is whether they apply to single trajectories or to groups of trajectories. The former are called individual behaviors, the latter collective behaviors.

Definition 1.9. A *trajectory individual behavior* is a trajectory behavior that is characterized by a predicate $p(T)$ that bears on a single trajectory T . \square

Definition 1.10. A *trajectory collective behavior* is a trajectory behavior that is characterized by a predicate $p(S)$ that bears on a set of trajectories S . \square

The *Tourist* behavior we have seen in the previous section is an individual behavior: for each single trajectory in the data set we can decide whether

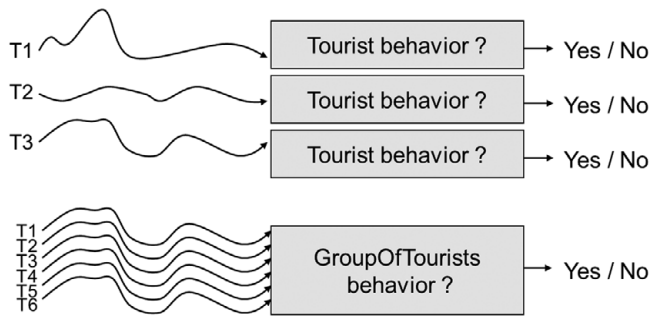


Figure 1.6 Individual (Tourist) versus collective (GroupOfTourists) Behaviors.

it shows this behavior or not. Should a *GroupOfTourists* behavior be defined (as a group of trajectories that represents a group of persons moving together and each of them complies with the *Tourist* behavior), it would be a collective behavior. Well-known collective spatio-temporal behaviors, such as *Meet* (a group of trajectories that simultaneously end up at the same place) and *Flock* (a group of trajectories that travel together), are described in Chapters 6 and 7.

Figure 1.6 illustrates that, for finding which moving objects show a given individual behavior, each trajectory has to be individually checked against the behavior predicate. On the opposite, finding which moving objects show a given collective behavior implies checking a group of trajectories.

Generally, collective behaviors are observed in groups of trajectories that are simultaneously run by various moving objects. But collective behaviors may also be defined for a set of trajectories run by a given moving object at different times. A typical example is the *Commuter* behavior that characterizes a group of trajectories made by the same object on working days and that show the same peculiar trace: they start from a point P1 and go to another one where they stay, then they end by going back to P1.

A case where the classification individual versus collective behavior is not necessarily intuitive is when the behavior involves multiple trajectories with one of them playing a special role in the group. For instance, given a group of tourists, the tourists and their guide move together but the guide's trajectory obeys additional rules: During the stops the guide is in the middle of the group; during the moves, the guide moves a few steps ahead of the other members. The group of tourists (guide included) shows the *GroupOfTourists* collective behavior, yet the guide trajectory complies with the individual *TouristGuide* behavior. Both the group and the guide's trajectory are needed in order to get the *TouristGuide* behavior.

Other cases exist where a fixed number of trajectories is needed for the behavior. For instance the *CourtshipDance* behavior of some birds, such as

cranes, involves two trajectories with the same role. Another example is the *Pursuit* behavior that also involves two trajectories, but with opposite roles.

A trajectory representation is inherently a temporally ordered list of elements, be they raw tuples (spatio-temporal positions) or annotated episodes. The predicate used to define a behavior can involve any number of elements. The simplest behaviors will only require a predicate on a single element. Examples include behaviors such as “starting from a given geo-object” (whose predicate only constrains the Begin element) and “passing by a given geo-object” (whose predicate is satisfied as soon as one of the trajectory elements is located inside or equal to the geo-object). More advanced behaviors rely on complex predicates that involve several elements so that each element has to satisfy the condition associated to it. A simple example is the *HomeToWork* behavior, whose predicate is composed of two component predicates: one on the Begin element and another one on the End element.

Complex predicates may require that their component predicates be satisfied by a sequence of elements that complies with a specified temporal order. Consider, for example, the predicate “starting at a given point P1, later crossing the area A1, 2 hours later crossing the line L1, and ending up inside the area A2.” This predicate on the one hand constrains the Begin and End elements (Begin must be point P1, End must be inside A2), and on the other hand imposes two additional constraints that have to be satisfied by some elements. Which elements satisfy the two constraints is not relevant, but the element crossing the A1 area has to come before the element crossing the L1 line. Complex predicates where a temporal order is specified define behaviors denoted as *Sequence* behaviors.

Definition 1.11. A *trajectory sequence behavior* is a trajectory behavior whose predicate is composed of several conditions, each condition being coupled with a temporal constraint, such that the constraints enforce a specific temporal order on the elements satisfying the conditions. \square

As sequence behaviors may be quite complex, a language is defined for expressing the various sequence operators that link the component conditions. The most usual operators are:

- `AND_THEN_NEXT[N]`: the next element (or the N next elements) of the trajectory must comply with the predicate;
- `AND_THEN_LATER[d]`: there must be later (or at least/exactly some duration *d* later) within the trajectory, an element that complies with the predicate.

The definition of the *Tourist* behavior given in the previous section is a complex behavior, but not a sequence one, because the two component predicates “it makes at least one stop in a place of kind Museum or TouristAttraction” and

“it makes one stop in a place of kind EatingPlace” can be satisfied in any order. On the opposite, the following would be a sequence behavior:

Tourist2 behavior: A daily trajectory shows the Tourist2 behavior if: its Begin point P1 is a place of kind “Accommodation,” it makes at least one stop in a in a place of kind “Museum” or “TouristAttraction,” later it makes one stop in a in a place of kind “EatingPlace,” and its End point is in the same P1 place as its Begin point.

Chapters 6 and 7 present data-mining methods for searching sequence behaviors. Chapter 12 presents a query language for searching complex behaviors containing generic temporal constraints.

1.6 Conclusions

In order to introduce the reader to the broad spectrum of concerns that are discussed in detail in the rest of the book, this chapter has aimed at providing a consistent vision of the trajectory domain. We have defined the basic concepts that underline trajectory management, emphasizing aspects related to various representations of trajectories. Secondly, we have shown how trajectory behaviors can be precisely described by predicates involving movement attributes and/or relationships to the context and/or semantic annotations.

While earlier research mainly focused on processing the raw data received from sensors, GPS devices and the like, recent research rather focuses on methods to enrich a movement track with more semantic, application-oriented information. Semantic additions enable new capabilities of running far-reaching analyses of mobility-related phenomena, thus conveying a huge potential for all kinds of innovative applications. As each application may have its own view of its trajectories, such as a discrete, continuous, or semantic view, we have defined three kinds of trajectory representations that can be superimposed.

In a broader perspective, several complementary types of movement remain to be investigated, including movement of large and deforming objects (e.g., oil spills, diseases), constrained movements (e.g., cars, trains that are constrained by a network), or more aggregated representations of movement, such as flows.

After choosing the representation of the movement best fitted for the application, frequently the major focus is to understand the behaviors of the moving objects. Understanding why and how people and animals move, which places they visit and for which purposes, what their activities are, and which resources they use is of tantamount importance for many kinds of decision makers, in particular public authorities in charge of managing societal resources.

At the core of behavioral analysis is identifying which characteristics of the moving objects define which behaviors. In the simplest case, experts define

the set of behaviors and the problem is to express these behaviors in terms of movement characteristics to be used for searching a trajectory database. Here, database approaches such as the one presented in Chapter 3 are suitable. A more challenging issue arises when no behaviors are known a priori. How can we learn potentially meaningful behaviors from trajectory analyses? Techniques for this kind of research typically include data mining, machine learning, and knowledge extraction in general, as well as visualization.

There are research efforts aiming at defining behaviors in a given domain in a more abstract and generic way, for example, not for the purpose of a specific application. These behaviors stem, for example, from an observation of possible spatio-temporal configurations of moving objects and are assumed to be relevant to a variety of applications. Other research aims at defining an ontology of all the behaviors. We presented a set of basic concepts regarding behaviors. Chapters 6 and 7 develop a more detailed discussion on behaviors (called patterns).

1.7 Bibliographic Notes

Background knowledge on spatial, temporal, and spatio-temporal data description and management is largely covered by the literature. The well-known ChoroChronos book written by Koubarakis et al. (2003) reports the outcomes from an early European project on spatio-temporal databases. Güting and Schneider (2005) is an excellent reference book on a formally sound approach to moving object management. This approach, built on abstract data types, is described in Chapter 3 of the present book. Finally, a conceptual perspective on spatio-temporal data modeling and manipulation is provided in Parent et al. (2006).

Most of the trajectory issues discussed in this chapter were first addressed in Giannotti and Pedreschi (2008), a book produced by the European GeoPKDD project on privacy-preserving techniques for trajectory mining. In this book, the chapters on “Basic Concepts of Mobility Data” and “Trajectory Data Models” nicely complement the content of our chapter.

The conceptual approach that has been very inspirational in writing this chapter was published in a journal by Spaccapietra et al. (2008). This paper develops a comprehensive view on trajectories from a conceptual data modeling perspective. It introduces the concept of semantic trajectories and of segmented trajectories, namely using Stop and Move episodes. Many further papers on trajectory analysis stem from a similar approach.

Trajectory behaviors have been extensively addressed. Dodge et al. (2008) is one of few contributions that aim at proposing a taxonomy of behaviors for raw trajectories. The authors studied the literature on data mining and visual analysis dealing with movement data and they collected definitions of various

movement behaviors, most of which are collective behaviors. The collected types of behaviors have been organized in an informal taxonomy based on the spatial and temporal characteristics of the raw trajectories.

Laube et al. (2005) centered their research on studying relative movement among a set of moving objects. They use a matrix of synchronized raw trajectories that allows an easy comparison of the movement of an object in time or of the movements of several objects at some instant. They analyze the variability of characteristics of the moving objects to characterize a number of behaviors as either individual or collective behaviors. They use the complex behavior concept (as a composition of basic behaviors) in the same way we used it in this chapter.

Wood and Galton (2009) and Wood and Galton (2010) develop a deeper investigation into concepts for collective behaviors. Their ontological approach develops some fundamental questions about definition and properties of groups. These questions are still open for research.

2

Trajectory Collection and Reconstruction

Gerasimos Marketos, Maria Luisa Damiani, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan

2.1 Introduction

The research area of trajectory databases has addressed the need for representing movements of objects (i.e., trajectories) in databases in order to perform ad hoc querying and analysis on them. During the last decade, there has been a lot of research ranging from data models and query languages to implementation aspects, such as efficient indexing, query processing, and optimization techniques.

This chapter covers aspects related to data collection and handling so as to feed trajectory databases with appropriate data. We will also focus on the step *trajectory reconstruction* of the *Geographic Privacy-aware KDD process* (illustrated in Figure 2.1) emerged from the GeoPKDD project which proposed some solid theoretical foundations at an appropriate level of abstraction to deal with traces and trajectories of moving objects aiming at serving real world applications. This process consists of a set of techniques and methodologies that are applicable to mobility data and are organized in some well-defined and individual steps that have a clear target: to extract user-consumable forms of knowledge from large amounts of raw geographic data referenced in space and in time. However, when mobility data are about individuals, data collection is subject to privacy regulations and restrictions. To enable privacy-aware collection of position data, a complementary class of techniques is used, known as *location PETs* (privacy-enhancing technologies).

This KDD process can be applied to heterogeneous sources of mobility data. The cellphone icon that is illustrated in Figure 2.1 could represent various data sets coming from various devices. In Section 2.2, we present such sources.

Before applying trajectory reconstruction techniques we may need to perform some basic trajectory preprocessing. This may include parameterized trajectory compression (so as to discard unnecessary details and concurrently keep

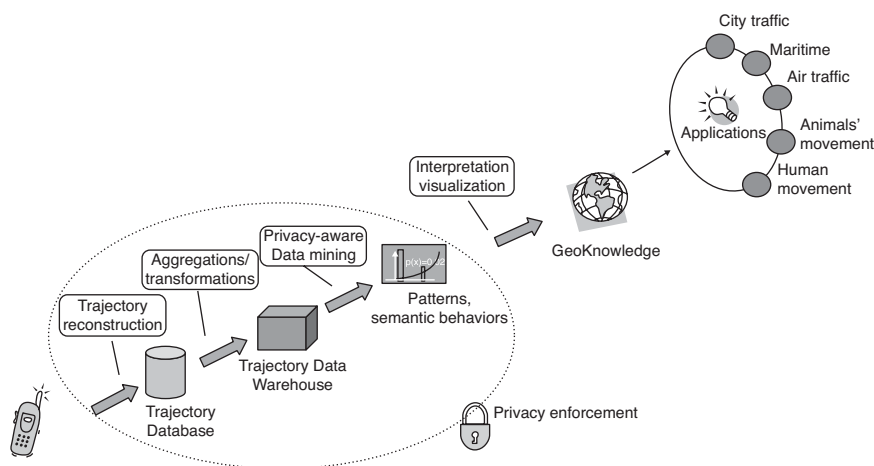


Figure 2.1 The big picture of moving object data management, warehousing, and mining concepts.

informative abstractions of the portions of the trajectories transmitted so far), as well as techniques to handle missing/erroneous values. Moreover, to deal with moving object applications that are restricted to some network, map-matched trajectories may be needed. In other words, we may need the specific trajectory points and portions to correspond to valid network paths. This may include, for example, performing preprocessing or postprocessing tasks that do not violate the validity of trajectories in terms of the real underlying network. We describe these kinds of tasks as trajectory data handling and we present them in Section 2.3.

In Section 2.4, we present trajectory reconstruction techniques for transforming sequences of raw sample points into meaningful trajectories and store them in trajectory databases. The reconstructed trajectories can be either semantic-free (raw trajectories) that just represent the movement of an object or semantically enriched, containing information about the nature of the movement.

Section 2.5 presents techniques for the privacy-preserving collection of trajectory data.

2.2 Tracking Trajectory Data

In this section, we present some technologies that can be used for tracking trajectories of moving objects. More specifically, these technologies provide us access to position data that may represent an incomplete, partial, or vague representation of the real movement of moving objects but with the appropriate handling techniques (Section 2.3) can lead to the reconstruction of trajectories (Section 2.4).

GPS Data

GPS is the fully-functional satellite navigation system that utilizes more than two dozen satellites. It broadcasts precise timing signals by radio to GPS receivers, allowing them to accurately determine their location (longitude, latitude, and altitude) in any weather, day or night, anywhere on Earth. A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include:

- The time the message was transmitted,
- Precise positioning information, and
- The general system health and rough orbits of all GPS satellites.

The receiver computes the distance to each satellite by using the messages it receives to determine the transit time of each message. These distances along with the satellites' locations are used to compute the position of the receiver. This position is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS-enabled devices show derived information such as direction and speed, calculated from position changes. GPS-enabled devices provide us with all the required information for trajectory tracking. They give us access in accurate, time-stamped locations for each tracked moving point.

GSM Data

GSM is the most popular standard for mobile phones in the world, nowadays used by more than 1.5 billion people across more than 210 countries and territories. The ubiquity of the GSM standard makes international roaming very common between mobile phone operators, enabling subscribers to use their phones in many parts of the world. GSM networks consist of a number of base stations, each responsible for a particular spatial area (known as "cell"). Hence, for each GSM-enabled device we can collect information about the base stations it was served by at different timestamps, and as such, assume its movement.

A GSM-enabled device can be tracked by collecting all the communication signals transmitted (cell, signal strength) between this device and the network infrastructure or by studying the log of the outgoing calls (UserID, data and time of the call, duration of the call, the cell where the call began, the cell where the call finished). However, in both levels the accuracy of trajectories that can be collected is very low since the most detailed level of available information is the network cell and not a spatial point.

Bluetooth Data

The movement of a Bluetooth device within an area can be tracked by considering the distance of the device from Bluetooth receivers and using trilateration

approaches. The distance of a Bluetooth device from a specific receiver can be calculated using techniques that consider signal levels.

The disadvantage of this technique is that it can be mainly used for indoor tracking of objects as Bluetooth receivers cover a limited area and they cannot really be used for outdoor object tracking.

RFID Data

The purpose of an RFID system is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate. A typical chip can store as much as 2 kilobytes of data. A reader is needed to retrieve the data stored on an RFID tag. A typical reader is a device that has one or more antennas that emit radio waves and receive signals back from the tag. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, and date of purchase. As in Bluetooth technology, RFID readers can locate tags within a limited area so it is hard to apply this technology for outdoor tracking of moving objects.

2.3 Handling Trajectory Data

Real-life trajectory data, collected using the technologies previously presented, are not really readily used for analysis purposes. In this section, we elaborate on various approaches for handling trajectory as a necessary step for identifying *clean* (i.e., without noise), *accurate* (i.e., map-matched), and *compressed* (i.e., compact) trajectories, from the original sequence of spatio-temporal positions (e.g., GPS records) of the moving objects.

2.3.1 Data Cleaning

Data sets collected by mobile sensors are often imprecise either unintentionally, due to limitations of positioning systems (e.g., inaccurate GPS measurement and sampling errors, signal loss, battery running out), or intentionally, so as to protect individuals' privacy (e.g., people may expose an approximation of their positions).

In case of unintentional (GPS) errors, trajectory cleaning (i.e., removing errors) is an important step in the procedure of constructing meaningful raw trajectories from the GPS feeds. Generally speaking, two types of GPS errors can be identified: *systematic* errors, due to a system's limitations, and *random* errors, due to external reasons. Systematic errors can be caused by horizontal dilution of position (HDOP) due to the low number of available satellites, while

random errors are small errors up to ± 15 meters caused by the satellite orbit, atmospheric and ionospheric effects, and receiver issues. We should note here that errors are related to the spatial positions and not to the temporal aspect of mobility as it is considered highly precise.

In order to identify systematic errors, researchers may resort to visual inspection in case of small data sets. For that reason, we could use a filtering method that filters noisy positions by taking advantage of the maximum allowed speed of a moving object. This threshold/parameter is used in order to determine whether a reported position from the GPS stream must be considered as noise and consequently discarded, or kept as a normal record.

On the other hand, random errors are small distortions from the true values. Their influence is reduced by smoothing methods. In the literature, different approaches can be found based on Gaussian kernels, where a smoothed spatial position is the weighted local regression based on past and future positions within a sliding time window considering the weight as a Gaussian kernel function, and Kalman filter, which uses measurements observed over time (the positions coming in the GPS receiver) and predicts positions that tend to be closer to the true values of the measurements.

2.3.2 Map Matching

The previous trajectory cleaning methods are designed for objects moving without any constraint in their movement. However, real-world applications usually consider objects that are restricted to move within a given spatial network that is represented as a graph (e.g., road/railway network) (you can find more information about this topic on Chapter 3). Other applications may consider spatio-temporal constraints (e.g., a pedestrian cannot walk at a speed above a certain limit, usually bats don't fly during the daytime).

For network-constrained trajectories, the map-matching approach refers to the mapping of a trajectory to the edges and nodes of the network. More precisely, the general idea is the replacement of each position of the original trajectory by the point on the network that is the most likely position of the moving object. From a computational point of view, map-matching methods can be categorized to online (processing streams of new positions in real time) or offline (when all positions are available), while both groups can be further classified as *geometric*, *topological*, or *hybrid* methods.

Geometric methods take into consideration the underlying road network and various distance measures to determine the actual traveled roads. These distance measurements can be point-to-point (e.g., Euclidian distance), point-to-curve (e.g., perpendicular distance), or curve-to-curve (e.g., Fréchet distance). For instance, Dijkstra's shortest path algorithm can be used to determine the distance

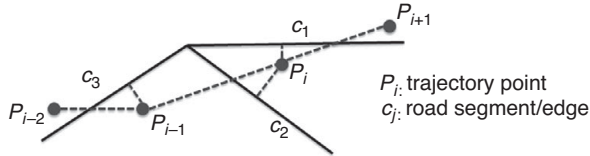


Figure 2.2 Applying map matching.

between a trajectory and a sequence of arcs on a map. The route with the smallest distance from the initial trajectory is taken as the map-matched trajectory. For instance, Figure 2.2 illustrates such a methodology: for every point P_i , given that point P_{i-1} has already been matched to an edge, the adjacent edges to this edge are the candidate edges to be matched to P_i and they are evaluated as illustrated in Figure 2.2. In this example, P_{i-1} is matched to edge c_3 , hence c_1 , c_2 , and c_3 are the candidate edges for point P_i . Two measures are used for choosing among the candidate edges that are based on similarity and orientation criteria. The higher the sum s of these measures is, the better the match to this edge is. If the projection of the current point on the candidate edges does not lie between the end points of any of these edges, the algorithm does not proceed to the next point. Instead, the nearest edge of the candidates is set as part of the trajectory and then the next set of candidate edges is evaluated. On the contrary to geometric approaches, the topological approaches account for the connectivity and contiguity of the road network without assuming any knowledge of the expected traveling route and the speed or heading information supplied by the GPS.

More recent map-matching methods deal with the problematic case where GPS data are arriving with low sampling rate (e.g., one point every two minutes) and high noise. These new methods employ both distance and topology and aim to align an entire trajectory with the road network. In some cases, not only distance and topology are used but also hidden Markov model approaches to find the most likely road route corresponding to a sequence of positions.

The various proposals usually include several postprocessing techniques to calibrate and correct the initial matching results. Obviously this worsens the cost/efficiency of the algorithm. This is an important issue that should be addressed by future research.

2.3.3 Data Compression

Trajectory data in applications grow progressively and intensively as the tracking time goes by. Such huge amounts of data raise storage, transmission, computation, and display challenges. Therefore, trajectory data compression is an essential task of trajectory reconstruction. The research in this area usually assumes that the objectives of trajectory compression are: (1) to reduce the size of the

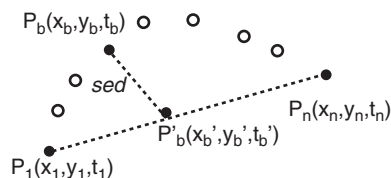


Figure 2.3 Using SED.

data set, (2) to ensure that the reduced data set should allow computations of acceptable/low complexity, and (3) to ensure that a trajectory from the reduced data set should not deviate from the original one by more than a given threshold.

From a geometric perspective, compression techniques exploit online simplification algorithms that remove positions from a trajectory without warping the trend of the trajectory or distorting the database. In general, trajectory compression algorithms can be classified into four categories: *top-down*, *bottom-up*, *sliding window*, and *opening window*. The top-down algorithm recursively splits the sequence of positions and only keeps the key (*representative*) positions in each subsequence, that is, the ones that lie far from the line that would result if these points were removed. A classical top-down method is the Douglas-Peucker (DP) algorithm, with many subsequent extensions. The bottom-up algorithm starts from the finest possible representation, and merges the successive points until some halting conditions are met. Sliding window methods compress data in a fixed window size; open window methods use a dynamic and flexible data segment size.

For instance, the *Top-Down Time Ratio* (TD-TR) and *Open Window Time Ratio* (OPW-TR) algorithms have been proposed for the compression of spatio-temporal data. The TD-TR approach uses the DP algorithm and, moreover, takes the time into account. In particular, it replaces the Euclidean distance used in DP by a time-aware one, called Synchronous Euclidean Distance (SED), as illustrated in Figure 2.3. In this example, let P_b be the currently examined point against line $P_1 P_n$. The DP approach uses the perpendicular distance of P_b to $P_1 P_n$, while the TD-TR uses the distance of P_b to $(P')_b$ (i.e., the SED). The coordinates of point $(P')_b$ are calculated using linear interpolation. The OPW-TR algorithm works as follows. Initially, it defines a line segment between the first and the third data point. If the SED from each internal point to the segment is not greater than a given threshold, the algorithm moves the end point of the segment one position up in the sequence. When the threshold is exceeded, the data point that causes the threshold excess or its precedent is defined as the end position of the current segment and the start position of a new one. As long as new positions arrive, the method continues as described.

Two other interesting algorithms in the literature are the Thresholds and STTrace, appropriate for online trajectory data compression. The algorithms

use the coordinates, speed, and orientation of the current position in order to calculate a safe area where the next position might be located. If the next incoming position lies in the calculated safe area, it can be ignored. There are two options for the definition of the safe area. It is either calculated by using the last position, whether it has been previously ignored or not, or by using the last chosen position. In order to achieve better results, a combination of the two algorithms is also proposed. Both areas are calculated, but only their intersection is defined as the safe area.

These trajectory compression approaches are primarily based on the extension of geometric methods such as the DP algorithm. However, they are not suitable for network-constrained trajectories. Therefore, recent works proposed another kind of trajectory compression model that makes use of the underlying road network. Through map matching, trajectories can be reconstructed (or represented) by only the matched road segments, without the need for keeping the original movement points.

2.4 Reconstructing Trajectories

Chapter 1 introduced the differentiation between raw and semantically enriched trajectories. Here we present reconstruction techniques for both types. Trajectory reconstruction refers to the task of transforming raw spatio-temporal positions into meaningful trajectories. An interesting note here is that different applications may need different trajectories. For instance, there may be a considerable difference between the semantic definitions of a trajectory given by a traffic analyst and, on the other hand, a logistics manager. Let us consider a fleet of trucks moving in a city and delivering goods in various locations. The logistics manager may consider, for each truck, a number of different trajectories (e.g., between the different delivery points) while the traffic analyst may consider a single trajectory for the whole day. Thus, in order to satisfy these two, quite different in semantics, requirements we would have to retrieve raw spatio-temporal position data from a common repository and then execute two different reconstruction tasks so as to produce trajectories that are semantically compliant to each domain. For instance, Figure 2.4a illustrates a raw data set of spatio-temporal positions. Different needs may result in different set of reconstructed trajectories (Figure 2.4b–d, respectively). Recalling the previous example of the truck data set, let us consider Figure 2.4b and c, which illustrate the reconstructed trajectories for the logistics manager and for the traffic manager respectively. Another example of trajectory reconstruction is presented in Figure 2.4d, which considers a compressed trajectory of the movement. The exact number of reconstructed trajectories depends on the different semantic definitions that can be given to a trajectory. In this section, we present reconstruction techniques that can be used to produce either raw or semantically enriched trajectories.

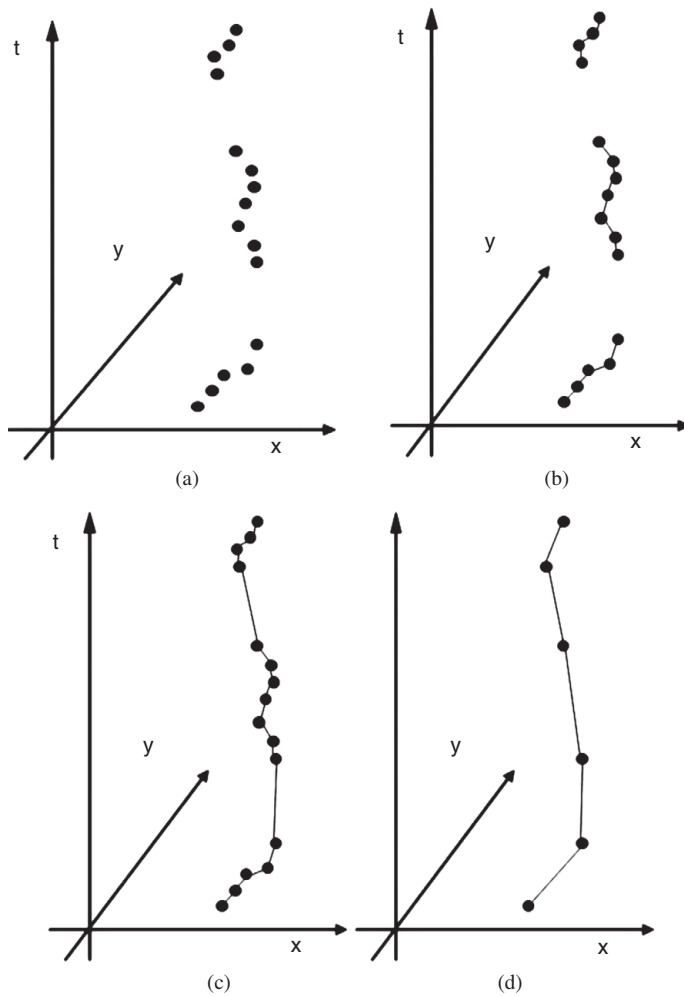


Figure 2.4 Three different trajectory reconstruction approaches (b, c, d) for a raw data set (a).

Reconstructing Raw Trajectories

Collected raw data represent spatio-temporal locations (Figure 2.5a). Apart from storing these raw data, we are also interested in reconstructing trajectories (Figure 2.5b). The so-called *trajectory reconstruction* task is not a straightforward procedure. Having in mind that raw points arrive in bulk sets, we need a filter that decides if the new series of data is to be *appended* to an existing trajectory or not.

The process of algorithm reconstruction needs a method for determining different trajectories, which should be applied to raw positions. Taking into consideration that the notion of trajectory cannot be the same in every application

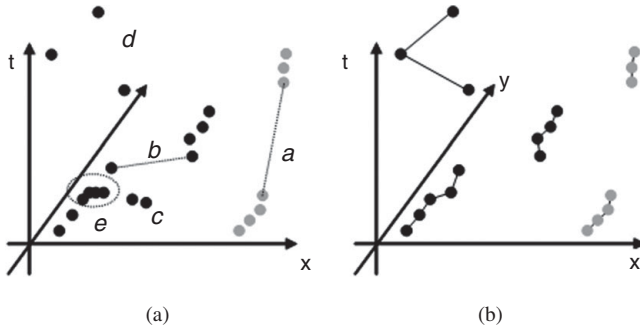


Figure 2.5 (a) Raw locations; (b) reconstructed raw trajectories.

due to the fact that different requirements and semantics arise, some generic trajectory reconstruction parameters can be:

- *Temporal gap between trajectories*: The maximum allowed time interval between two consecutive spatio-temporal positions of the same trajectory for a single moving object (case *a* in Figure 2.5a).
- *Spatial gap between trajectories*: The maximum allowed distance in 2D plane between two consecutive spatio-temporal positions of the same trajectory (case *b* in Figure 2.5a).
- *Maximum speed*: The maximum allowed speed of a moving object, used to determine noisy spatio-temporal positions (case *c* in Figure 2.5a).
- *Maximum noise duration*: The maximum duration of a noisy part of a trajectory so as to consider creating a new trajectory containing this part (case *d* in Figure 2.5a).
- *Tolerance distance*: The maximum distance between two consecutive spatio-temporal positions of the same object in order for the object to be considered as stationary (case *e* in Figure 2.5a).

Reconstructing Semantic Trajectories

Raw trajectories contain only spatio-temporal positions (x, y, t) , which are insufficient for building meaningful trajectory applications. Therefore, researchers have proposed to reconstruct trajectories from the low-level collected data (e.g., GPS records, movement tracks) to high-level data abstractions, thus building semantic trajectories. The idea of semantic trajectories is to encode meaningful geo-locations/geo-objects (e.g., points of interest such as a shopping mall, roads) into the raw spatio-temporal tracks; additional semantic annotations (e.g., trajectory behaviors such as traveling in Paris, walking on Avenue des Champs-Élysées, taking Metro 3, shopping in a supermarket) are attached to the semantic trajectories.

Figure 2.6 briefly presents the main procedure of reconstructing such semantic trajectories from the raw GPS alike mobility records. From the initial GPS

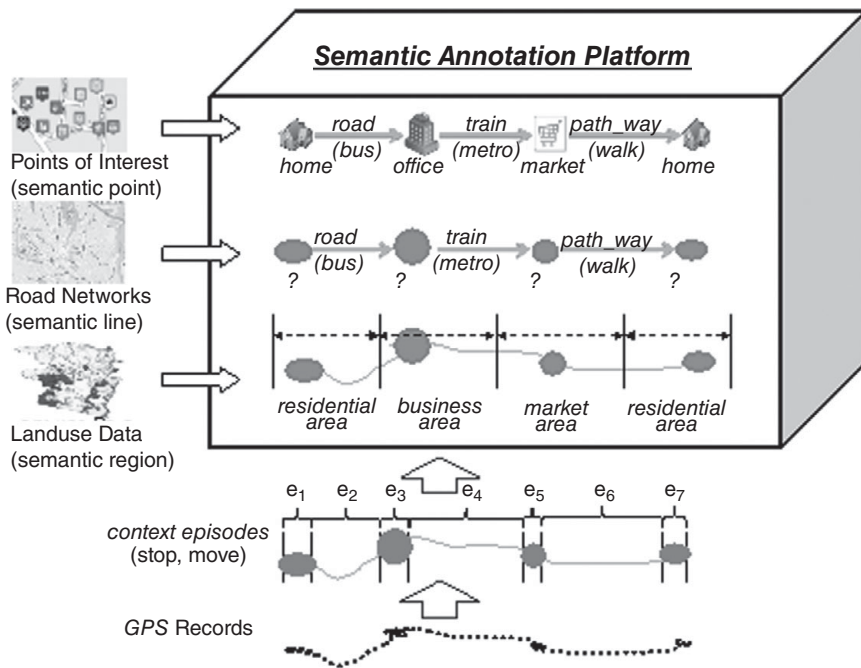


Figure 2.6 Annotation for semantic trajectories.

records, we can compute the trajectory episodes (e.g., stops, moves, which are largely used in the literature to understand the structure of trajectories, presented in Chapter 1); afterward, a couple of dedicated annotation algorithms are provided for enriching trajectories using additional geo-objects and semantic tags. There are four main technical components for constructing such semantic trajectories, as follows:

- **Building trajectory episodes:** The aim is to build trajectory episodes to further understand the inner structure of each individual raw trajectory. Trajectory episode is a subsequence of the raw trajectory. Trajectory data points inside one episode are more or less homogenous (e.g., staying in the same place, having the same travel speed), though data points in two neighboring episodes are unrelated. There are different kinds of episodes, such as Begin, End, Stop, and Move. In addition to these four types of episodes, additional episodes can be further designed according to the application scenarios, for example, specific episodes for representing congestions in traffic. The core issue here is to design efficient and robust trajectory segmentation algorithms to find these meaningful episodes. A couple of trajectory segmentation algorithms are proposed for building trajectory episodes, such as velocity, density, orientation, and even time-series-based segmentation methods.

- *Annotating trajectory with regions*: This component enables annotation of trajectories with meaningful geographic or application domain sources of semantic regions. It does so by computing topological correlations between trajectories and third party data sources containing geo-objects of regions (called regions of interest or ROI). We need to design a spatial join algorithm, which can work for both regular regions (e.g., $100\text{ m} \times 100\text{ m}$ grid-based land use data) and irregular regions (e.g., regions with free-style shapes such as EPFL Rolex Learning Center).
- *Annotating trajectory with lines*: This component annotates trajectories with lines of interest (LOI) such as road networks and considers variations present in heterogeneous trajectories (e.g., vehicles run on road networks, while human trajectories use a combination of transport networks and walkways). Given data sources of different forms of road networks, the purpose is to identify *correct* road segments as well as infer the transportation modes such as “walking,” “cycling,” and “public transportation” such as metro and bus. Thus, the algorithms in this component include two major parts: the first part is designing/reusing a global map-matching algorithm to identify the correct road segments for the move episodes of a trajectory, and the second one is inferring the transportation modes that the moving objects/people used during their moves.
- *Annotating trajectory with points*: This component annotates the Stop episodes in trajectory using information about suitable points of interest (POIs). Examples of POI are “restaurants,” “bars,” “shops,” and “movie theaters.” For scarcely populated landscapes, it is relatively trivial to identify the objective of a stop (e.g., petrol pump on a highway, back home in a very sparse residential area). However, densely populated urban areas bring many different types of candidate POIs for a trajectory stop. The problem of inferring stop behaviors using POIs becomes challenging. Further, low GPS sampling rate due to battery outage and GPS signal losses makes the problem more intricate. Recently, a HMM (hidden Markov model)-based inference algorithm has been designed to extract the underlying stop behaviors in the trajectory. In this algorithm, the location of individual trajectory stop is modeled as a model observation, whilst the POI category is considered as the hidden state that needs to be extracted.

2.5 Protecting the Privacy of Individuals’ Positions

This section overviews techniques that aim at protecting users’ privacy during the data collection process. The concern for privacy stems from the fact that whenever position refers to individuals, position is qualified as personal data, and collecting personal data is restricted by privacy norms and law in several

countries worldwide. In particular, semantic trajectories magnify the risk for privacy because behavior information on individuals is explicitly extracted and represented in a machine-readable form, and therefore can be used within information processing applications and easily unfolded to third parties. Though fundamental, privacy regulations are not capable of preventing malicious and curious parties from improperly accessing and using collected data. This instead is the goal of location PETs (privacy-enhancing technologies). In general, location PETs can be applied at two different stages:

1. Before position data are collected. In this case the goal of location PETs is to prevent mobility data collectors from obtaining the exact location and trace of individuals, everytime and everywhere. Because these techniques are applied on the fly, we refer to this form of protection as *online location privacy*.
2. After position data are collected and trajectories reconstructed. The goal of location PETs is to shape trajectory data in a way that the data set can be published or released to some other party without incurring privacy violations. We refer to this as *offline location privacy*.

Offline and online location privacy present different requirements, which call for different solutions. In particular, the solutions for the online protection of location privacy have to deal with incomplete knowledge of the individuals' trajectories (usually only the current and past positions are known); moreover, techniques must be efficient so as not to compromise the effectiveness of data collection. In what follows, we survey major paradigms supporting online location privacy while techniques for offline location privacy will be presented later on in Chapter 9.

2.5.1 Online Location Privacy

Research on position privacy took off early last decade with the emergence of mobile applications enabling the tracking of moving objects, for example, the vehicles monitored by a fleet management system, and location-based services (LBS), for example, search of points of interests nearby. These applications typically rely on a client-server architecture: the position is collected by mobile devices (the clients) and conveyed to a server handled by a service provider. In this scenario, service providers are in the position of collecting large amounts of position data, therefore, if they are disrespectful of users' rights and requirements or, simply, if the collected data are stolen, users' privacy is at stake. Commonly, location PETs seek to limit the transmission of either accurate or explicit location information to service providers. These techniques can be further classified based on the information to be protected, that is, the privacy goals. In particular,

we distinguish three main goals: *identity privacy*, *location privacy*, and *semantic location privacy*. In what follows we survey representative location PETs addressing these goals.

Identity Privacy

Identity privacy techniques are conceived to forestall the reidentification of seemingly anonymous users based on position information. For example, consider the case in which an LBS is offered to the members of a community potentially subject to discrimination, for example, the gay community, and assumes users will interact with the system through pseudo-identifiers. Unfortunately, simply stripping off users' identifiers is not sufficient to ensure anonymity, because the service provider can draw identities from trajectory information; for example, if a user requests the service from a certain place early in the morning, it is likely that such a place is his or her home and thus the user can be easily reidentified through a white pages service. We refer the reader to the literature for a survey of identity privacy techniques and limit ourselves to consider one of the most popular paradigms, that is, *location k -anonymity*.

Given a population of users, location k -anonymity postulates the following requirement: that the user's position disclosed to the service provider must be indistinguishable from the position of at least $k - 1$ other users. In practice, the exact user's position must be replaced by a coarser position, normally called *cloaked region*, large enough to contain the position of $k - 1$ other users located nearby at the time the online service is requested. Accordingly, the service provider cannot identify the requester of the service based exclusively on the position information. This situation is exemplified in Figure 2.7. For $k = 10$, the position of the single individual is replaced by a larger region (i.e., a cloaked region) containing 10 persons. If the online service is requested from this region, the maximum probability of identifying the requester is $1/10$. Another prominent feature of this privacy mechanism is that it typically requires a dedicated trusted middleware, the *location anonymizer*, between the clients and the service provider. The role of the location anonymizer is to collect the position of all the clients, intercept the individual's requests, replace the user's identifier with a pseudo-identifier, and, finally, replace the true position with the dynamically generated cloaked region.

One representative solution of this class is the Casper system (Figure 2.8). Casper consists of the location anonymizer and the *privacy-aware query processor*, a software component that runs on the server and resolves users' requests with respect to a position that is not a point, as usual, but a region, and returns a set of candidate answers.

A common criticism to location k -anonymity is that it is difficult to gauge which size of k is minimally necessary or sufficient. The higher the value of k , the higher the level of protection but also the loss of position accuracy, that

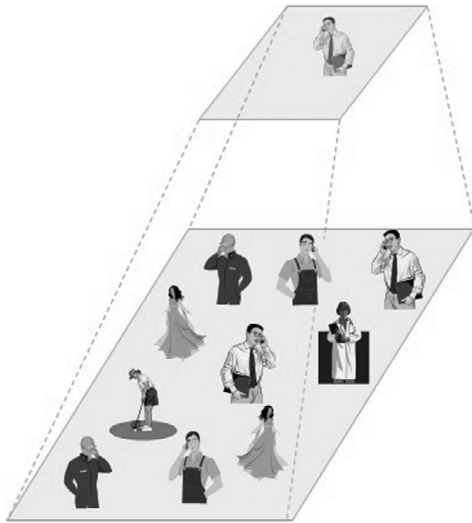


Figure 2.7 A cloaked region for 10-anonymity.

is, the cloaked region is likely larger. Moreover, the position accuracy varies in time and space based on the distribution of people in space, that is, if individuals are sparse then the cloaked regions are larger.

Location Privacy

Unlike identity privacy, location privacy aims at protecting the position information. The protection strategy is to transmit a position that is somewhat different in the content or in the form from the actual position. In particular, the disclosed position can be fake, cloaked, or transmitted using some cryptographic protocol.

- A *fake* position is a position deliberately represented with a wrong value. Privacy is achieved from the fact that the reported position is false. The accuracy and the amount of privacy mainly depend on how far the reported location is from the exact location. For example, the client requesting a service, for example, “where is the closest restaurant?” can transmit to the service provider a fake position in proximity of the actual position and then properly filter out candidate answers.

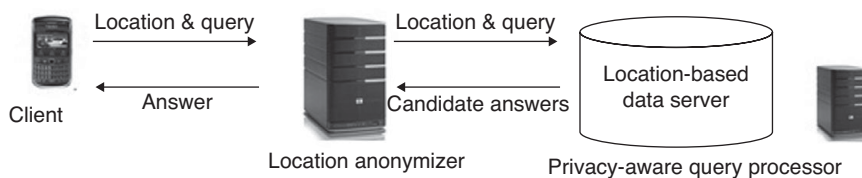


Figure 2.8 The Casper architecture.

- An *obfuscated* position (another term for cloaked region) is a coarse region including the exact user's location. Therefore the service provider does know that the user is located in the cloaked region, but has no clue where exactly the user is located. A popular obfuscation method, which is often used in commercial applications, replaces the actual position with a predefined region chosen in a taxonomy of locations at different granularities, for example, street, zip code area, city. Unfortunately, predefined locations can be too broad to ensure an appropriate quality of service, for example, a zip code region can cover an area of few squared kilometers, or conversely can be too small to provide privacy guarantees, for example, a short street. Another simple method obfuscates the position with a circle of user-defined radius and random center containing the actual position. In other solutions, the size of the obfuscated region can be the result of a trade-off between privacy and position accuracy. Moreover, the transmission of the position can be also delayed a while to cloak the temporal dimension.
- *Cryptographic protocols* define techniques for the secure collaboration of different parties. An example of cryptographic protocol used for privacy protection in LBS is PIR (private information retrieval). This technique allows users to issue a query without disclosing to the LBS provider the information that is requested as well as the information being returned. In this sense this technique protects both the identity and the location. The method ensures the maximum privacy. However, it incurs high computational costs and can be only applied to certain categories of queries, for example, the retrieval of stationary objects (i.e., nonmobile objects).

One specific problem that may arise when the position is obfuscated by a coarse region is that consecutive positions in the user's trajectory are correlated, that is, the presence in one region constrains the position in the subsequent regions. This information can be exploited to prune the obfuscated regions and more precisely delimitate the user's position. To prevent this inference when the maximum speed of the user is known (e.g., the user can be a pedestrian, a car driver, a cyclist, and so on) and the movement is frequently sampled, that is, the position is continuously reported, an approach is to modify the position in space and time before it is released. This form of privacy leak is also called *velocity-based linkage attack*.

Semantic Location Privacy

Semantic location privacy is a form of location privacy that aims at preventing data collectors from identifying the semantic locations in which users stay, for example, hospitals, religious buildings, and so on. Forestalling this type of inference is important for the construction of privacy-aware semantic trajectories.

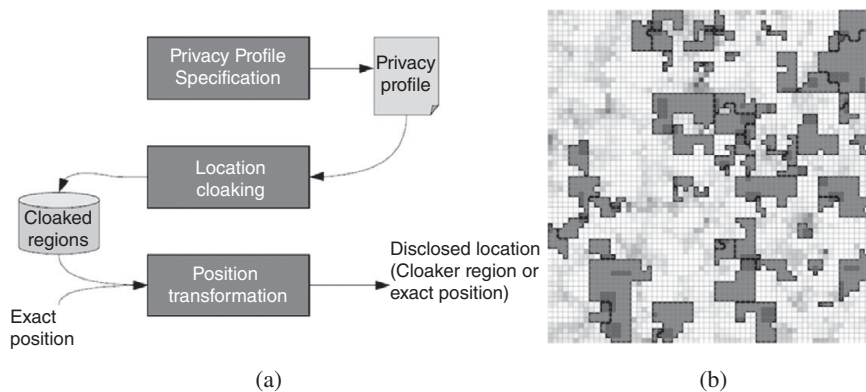


Figure 2.9 The Probe system: (a) The workflow. (b) Obfuscated map: the blue polygons represent cloaked regions, the red rectangles sensitive places, the gray background the distribution of population in space. (See color plate.)

The motivation behind semantic location privacy is that the sensitivity of positions may vary depending on the nature of places; for example, the position of a user staying in an oncological clinic is likely *more sensitive* than the position of a user walking along a street. If all the positions are treated as though they are sensitive, the protection would be excessive. More effective is to obfuscate only those positions that are perceived as sensitive, while disclosing the others with no change. In this way the loss of position accuracy is limited. This form of obfuscation is called semantic location cloaking. A sound semantic cloaking strategy should guarantee:

- *Semantic diversity*: The user's position cannot be blurred exclusively when the user is inside a sensitive place, but also when he or she is outside. That way, the place in which the user is located remains uncertain. An obfuscated region thus must include places of diverse types.
- *Independence* of the position cloaking method from the user's position. This condition prevents the discovery of the correlation between the cloaked region and the true position, which could be exploited to infer where the user is located.

These guidelines have been embodied in the privacy-preserving framework called Probe (Privacy-Aware Obfuscation Environment).

Figure 2.9 illustrates the workflow of the privacy enforcement process in the Probe system. Users first specify in a privacy profile which categories of points of interest are sensitive (selecting, for example, from a pre-defined list, for example, hospitals, religious buildings, and so on) along with the degree of privacy desired for each of those categories. For example, a privacy degree of 0.1 assigned to hospitals means that the (posterior) probability of locating the

user inside a hospital must be less than 0.1. Next, coarse regions are generated satisfying the privacy preferences, independently from the user's position, in order to prevent possible inferences on their reciprocal positions. A sample set of obfuscated regions is shown in Figure 2.9b. Finally, at runtime if the user's position falls inside one of the coarse regions, that region is delivered instead of the exact position. This solution is grounded on a conceptually founded privacy metric. Moreover, an additional metric is defined, the utility metric, providing a measure of the spatial accuracy of the cloaked regions. Unlike more traditional obfuscation techniques, the utility measure can be computed prior to any service request. In this way users can tune and balance the amount of privacy with the quality of service.

2.6 Conclusions

In this chapter, we presented techniques for collecting mobility data and handling them appropriately (applying data cleansing, data compression, and map matching) so as to produce noise-free and meaningful trajectories (trajectory reconstruction). Finally, privacy issues in mobility data collection and handling were discussed.

We outline next a few research directions that originate in the discussion provided in this chapter.

With respect to *trajectory reconstruction*, future work may include the exploration of intelligent ways to automatically extract proper values of trajectory reconstruction parameters according to a number of characteristics of data sets, as well as the extension of this technique so as to be able to identify different movement types (pedestrian, bicycle, motorbike, car, truck, etc.) and hence to apply customized trajectory reconstruction.

With respect to *privacy issues*, major research directions include *privacy usability*, that is, how to provide personalizable, conceptually founded, and simple-to-use privacy mechanisms so to enhance user experience; and *context-aware location privacy*, that is, tailoring privacy protection based on the context in which individuals are located. While semantic location privacy is a first attempt to introduce the contextual dimension in privacy, this notion can be extended along several directions; for example, to account for the temporal and social dimension of privacy.

2.7 Bibliographic Notes

In this section, we distinguish and annotate some works from the literature.

With regard to the data-handling approaches, [Yan et al. \(2010\)](#) proposed a Gaussian kernel-based local regression model to smooth out GPS feeds. [Brakatsoulas et al. \(2005\)](#) proposed the methodology for map matching that is

illustrated in Figure 2.2. [Quddus et al. \(2007\)](#) proposed a technique for replacing each position of the original trajectory by the point on the network that is the most likely position of the moving object. [Greenfeld \(2002\)](#) proposed a method based on topological analysis using the observed position of the individual without assuming any knowledge of the expected traveling route and the speed or heading information supplied by the GPS. Furthermore, [Newson and Krumm \(2009\)](#) used hidden Markov model approaches to find the most likely road route corresponding to a sequence of positions.

[Meratnia and de By \(2004\)](#) proposed the Top-Down Time Ratio (TD-TR) and Open Window Time Ratio (OPW-TR) algorithms for the compression of spatio-temporal data. [Potamias et al. \(2006\)](#) proposed the two algorithms, called Thresholds and STTrace, respectively, for online trajectory data compression. [Kellaris et al. \(2009\)](#) present a different approach by replacing certain episodes of a trajectory by selected shortest paths between the beginning and ending position of these episodes. As for the trajectory reconstruction topic, [Marketos et al. \(2008\)](#) presented a method for determining different trajectories as part of a trajectory reconstruction manager. On the other hand, [Yan et al. \(2011\)](#) presented a technique for reconstructing semantic trajectories from the raw GPS mobility records.

With regard to privacy issues, [Gruteser and Grunwald \(2003\)](#) introduced the concept of location k -anonymity in the context of LBS; [Jensen et al. \(2009\)](#) introduced the dichotomy of identity privacy versus location privacy; Casper ([Chow et al., 2009](#)) is a major privacy preserving framework supporting location k -anonymity; the velocity-based attack is described in more detail in [Ghinita et al. \(2009\)](#); [Damiani et al. \(2010, 2011\)](#) introduce the semantic location cloaking paradigm.

3

Trajectory Databases

Ralf Hartmut Güting, Thomas Behr, and Christian Düntgen

3.1 Introduction

In this chapter, we consider the problem of modeling and representing trajectories in the context of database systems. Since about 1995 there has been research on *moving objects databases* (MODs), also termed *spatio-temporal databases*. The general goal has been to allow one to represent moving entities in databases and to enable a user to ask all kinds of questions about such movements. This requires extensions of the DBMS data model and query language. Further, DBMS implementation needs to be extended at all levels, for example, by providing data structures for representation of moving objects, efficient algorithms for query operations, indexing and join techniques, extensions of the query optimizer, and extensions of the user interface to visualize and animate moving objects.

Moving objects databases come in two types. The first represents a set of currently moving objects. One is interested in maintaining the current locations and asking queries about current and expected near future locations. The second type maintains complete histories of movement. These are sometimes called *trajectory databases* and are the topic of this chapter.

Whereas spatio-temporal databases had been around for a much longer time, they supported only discrete changes of geometries over time. The emphasis in the new field of moving objects databases is to consider *continuously changing geometries*. Neither the position of a car on a road nor the shape and location of a hurricane changes in discrete steps; these are clearly continuous phenomena.

A driving force in the development of database systems has always been to provide to the user a simple conceptual model of data. Relational databases have been so successful because they introduced the simple view of representing

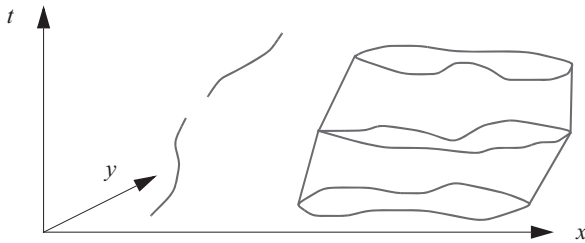


Figure 3.1 A value of type moving point (*mpoint*) and a value of type moving region (*mregion*).

data in tables and allowing one to manipulate and combine tables, rather than thinking of records in files containing fields in certain formats.

In a similar way, moving objects databases let the user view a vehicle moving on a road simply as a time-dependent position (either relative to the Euclidean plane or to the road). Mathematically, this is a function

$$f : \textit{instant} \rightarrow \textit{point}$$

if *instant* represents a continuous domain of time and type *point* represents (x, y) positions in the Euclidean plane. Such a function can be visualized in a 3D (x, y, t) space as shown in Figure 3.1.

Obviously, to arrive at a powerful query language on moving objects we need not only a simple view of data but also operations to manipulate them. What can we do with a continuous curve as shown in Figure 3.1?

For example, we can project it into the (x, y) plane. This forgets the temporal information and returns just the path in the plane the object (e.g., vehicle) has taken.

We can also project it on the time axis and get the time interval(s) when the object existed (more precisely, when its movement information is available).

We can reduce it to the times when the position has certain properties, for example, when it is inside a given region of the plane, or within some distance to another object, perhaps even a moving object.

A model of data together with some operations on it is captured by the concept of an *abstract data type* (ADT). Hence the idea is to model the time-dependent position of a vehicle as an abstract data type. Because only the position as a point is represented (ignoring the shape of the vehicle), the data type is called *moving point* (*mpoint*).¹ Similarly, for some entity for which capturing the extent is relevant (e.g., a forest fire), the time-dependent shape and location is represented in a data type *moving region* (*mregion*).

¹ We denote data types in italics and underlined.

The operations that we introduce need a name as well as definition of argument and result types. More formally, this is called a *signature*. For the three operations mentioned above, the signatures would be:

trajectory: $\underline{mpoint} \rightarrow \underline{line}$
deftime: $\underline{mpoint} \rightarrow \underline{periods}$
at: $\underline{mpoint} \times \underline{region} \rightarrow \underline{mpoint}$

Of course, the data types to represent the arguments and results must be available in the system as well. Types *line* and *region* may be available in DBMSs with spatial support. Type *periods* represents a set of disjoint time intervals and must be added.

What are the advantages of using such a model? Chapter 1 introduced *movement tracks* as the set of captured data over time for a moving object and explained that it typically can be represented as a sequence of pairs (*instant*, *position*), hence as a sequence $\langle (t_1, p_1), \dots, (t_n, p_n) \rangle$ where t_i is of type *instant* and p_i of type *point*. Given a DBMS that has such data types, we can then simply represent a set of captured moving tracks in a table with schema:

```
Observations(Id: int, Time: instant, Position: point)
```

Is it not sufficient to use such a representation in a DBMS? It is sufficient as long as one tries to formulate only very simple queries. Basically, simple queries (for a MOD) become difficult to formulate and advanced queries become practically impossible. Consider two simple queries.

1. Where have the vehicles been at 6:30 P.M.?

The problem is that positions generally have not been recorded at 6:30 P.M. In the SQL query, for each vehicle we have to find the last recorded position before 6:30 P.M. and the first after 6:30 P.M. Then, in the `select` clause we need to perform interpolation between the two time instants and positions with the time argument 6:30 P.M.

In a MOD, one would instead have a table

```
Vehicles(Id: int, Trip: mpoint)
```

and the query is written as

```
select Id, val(Trip atinstant six30) as Pos630 from Vehicles
```

The query operations used are explained in Section 3.2.

2. At what times and positions did vehicles pass the river Rhine?

Here it is not so easy to determine positions before and after crossing the river Rhine and then do the interpolation as above. Perhaps the best strategy is to perform a self-join on the `Observations` table to put together pairs of adjacent observations and then to construct a line segment connecting

them. Using some spatial database capability, these line segments can then be checked for intersection with the river Rhine. If we have kept times and positions for the observations corresponding to the start and the end of the line segment, we may again do interpolation as for the previous query.

Assume we have a table for rivers.

```
Rivers(Name: string, Curve: line)
```

In the MOD, the query is written as follows.

```
select v.Id, inst(initial(v.Trip at r.Curve)) as PassingTime,
       val(initial(v.Trip at r.Curve)) as PassingPos
from Vehicles as v, Rivers as r
where r.Name = "Rhine" and v.Trip passes r.Curve.
```

Again, query operations are explained in the following section.

Besides easier formulation of queries, a MOD system can offer more efficient implementation techniques including indexing and query optimization as the system is “aware” of the moving objects.

The rest of the chapter is structured as follows. Section 3.2 describes the data model and query language for a MOD based on abstract data types. There are two prototypical implementations of this model, `SECONDO` and `Hermes`. In Section 3.3 we describe `SECONDO`. Section 3.4 discusses alternative representations of sets of moving objects in the context of this model, including creating the representations from raw trajectories. Section 3.5 addresses indexing of moving objects. Section 3.6 provides a short introduction to `Hermes`, the other MOD prototype, and explains some differences. The chapter ends with conclusions (Section 3.7) and bibliographic notes (Section 3.8).

3.2 Data Model and Query Language

In this section we address the extensions of a DBMS data model and query language to support representation and querying of moving objects. We have already seen in the introduction that the basic idea is to use abstract data types. These can be embedded in the role of attribute types into a relational or other DBMS model, and the ADT operations can be embedded into the DBMS query language, typically SQL.

The fundamental data type *moving point* (*mpoint*) to represent a trajectory also has been introduced already. To obtain an expressive query language, the model provides several further data types together with a carefully designed set of operations. In the following section we motivate and introduce these types and operations by examples. Later we consider the design principles that have led to this model and we briefly sketch its implementation.

3.2.1 Motivating Examples

Examples are based on a database that is delivered with the `SECONDO` system presented in Section 3.3. `SECONDO` is open source, so the reader can in fact install `SECONDO` and run the example queries.

The example database is called `berlintest`. It contains spatial data about the city of Berlin and some moving object data. Here we will use the following database objects. Note that in `SECONDO` a database may hold not only relations but also “atomic” objects of any available data type.

Relation `Trains` describes underground trains moving according to schedule on a certain day in the city of Berlin.

```
Trains(Id: int, Line: int, Up: bool, Trip: mpoint)
```

Each tuple describes one train trip by its identifier, the number of the train line to which it belongs, in which direction along the route it was going, and the complete movement description in attribute `Trip`.

Further objects are:

```
train7: mpoint, mehringdamm: point, thecenter: region
```

Here `train7` is a DB object with a value of type *mpoint*. `mehringdamm` is an underground train station in Berlin. Finally, `thecenter` is a region roughly describing the city center.

Let us start with some simple expressions on atomic objects. Expressions are composed of database objects, constants, and operations. `SECONDO` provides the `query` command to evaluate expressions, so one can write `query 3 * 4` and get 12 as a result.

```
query train7
```

This is already a very simple expression and it returns a value of type *mpoint*. In `SECONDO`, this value is displayed at the GUI as a point at the position of the start time. The movement can then be animated.

The following operations have already been introduced in the introduction:

trajectory:	<i><u>mpoint</u></i>	→	<i><u>line</u></i>
deftime:	<i><u>mpoint</u></i>	→	<i><u>periods</u></i>
at:	<i><u>mpoint</u></i> × <i><u>region</u></i>	→	<i><u>mpoint</u></i>

The expressions

```
train7 at thecenter, trajectory(train7 at thecenter),
deftime(train7 at thecenter)
```

return `train7` reduced to the times when it was in the city center area, the path taken in the center, and the time when it was at the center.

We can determine the distance between two moving objects or a moving object and a static object.

```
query distance(train7, mehringdamm)
```

Clearly as `train7` is moving, the distance to `mehringdamm` is time dependent. Hence the result is a real number varying with time. There is a data type for this called *moving real* (*mreal*) and the **distance** operation has signature

distance: $\underline{mpoint} \times \underline{point} \rightarrow \underline{mreal}$

It would be nice if we could determine when and where the speed of `train7` has been higher than 50 km/h. We can write it as follows.²

```
query speed(train7) > 50
```

Here a time-dependent speed, obviously an *mreal*, is compared to a *real* constant. The result is a time-dependent Boolean value, represented in a type *mbool*. Hence the two operations used have signatures:

speed: $\underline{mpoint} \rightarrow \underline{mreal}$
<: $\underline{mreal} \times \underline{real} \rightarrow \underline{mbool}$

We can determine the position of a moving object at any instant of time (it may be undefined if the *mpoint* function is not defined at that time). We can also reduce it to a given time interval (or set of time intervals).

```
let six30 = theInstant(2003, 11, 20, 6, 30);
let kmh = 1000 / 3600;
```

```
query val(train7 atinstant six30)
```

```
query trajectory(train7 atperiods
  deftime( (speed(train7) > (50 * kmh)) at TRUE ) )
```

Here we define `six30` as 6:30 A.M. on the day when trains are defined. We also introduce `kmh` as the factor to convert km/h to m/s. The first query then determines the position of `train7` at 6:30. The second reduces `train7` to the periods of time when its speed was higher than 50 km/h. The signatures used are the following.

² To be honest, one has to be a bit careful with the units used. In the `berlintest` database, geometries are given in units of meters, hence the speed of `train7` will be returned in m/s rather than km/h and one needs to apply the appropriate factor to the constant, omitted here for clarity.

atinstant: $\underline{mpoint} \times \underline{instant} \rightarrow \underline{ipoint}$
inst: $\underline{ipoint} \rightarrow \underline{instant}$
val: $\underline{ipoint} \rightarrow \underline{point}$
at: $\underline{mbool} \times \underline{bool} \rightarrow \underline{mbool}$
deftime: $\underline{mbool} \rightarrow \underline{periods}$
atperiods: $\underline{mpoint} \times \underline{periods} \rightarrow \underline{mpoint}$

The operation **atinstant** returns a data type $\underline{intime}(\underline{point})$, \underline{ipoint} for short. The type represents a pair (i, p) consisting of an $\underline{instant}$ and a \underline{point} . From such pairs one can determine the two components using the operations **inst** and **val**. Operation **at** reduces a time-dependent Boolean value to the times when it assumes the second argument. **deftime** works for type \underline{mbool} in the same way as for \underline{mpoint} .

The need to reduce a moving object to the times when it fulfills certain properties occurs frequently. For a moving object x , the expression

\times **atperiods deftime**(predicate(x) **at** TRUE)

can be abbreviated to \times **when**[predicate(x)] using operator **when** with signature

when: $\underline{mpoint} \times \underline{mbool} \rightarrow \underline{mpoint}$

Hence we can write the previous query more simply, as

```
query trajectory(train7 when[speed(train7) > (50 * kmh)])
```

It goes without saying that all the operations presented can be used in set-oriented queries, that is, in the `select` or `where` clause of an SQL query.

We also need some predicates to determine whether a moving object passes through a certain area or is defined at a given time. The following query finds all trains passing through `mehringdamm` and determines the times when they arrive at or leave this station.

```
select Id, Line, Up,
       inst(initial(Trip at mehringdamm)) as ArrivalTime,
       inst(final(Trip at mehringdamm)) as DepartureTime
from Trains
where Trip passes mehringdamm
```

Here operations are used:

passes: $\underline{mpoint} \times \underline{point} \rightarrow \underline{bool}$
initial, final: $\underline{mpoint} \rightarrow \underline{ipoint}$
at: $\underline{mpoint} \times \underline{point} \rightarrow \underline{mpoint}$

As a final example, how can we find pairs of trains that met, that is, have been at the same place at the same time? When and where did they meet?

```
select t1.Id, t1.Line, t2.Id, t2.Line,
       inst(initial(intersection(t1.Trip, t2.Trip))) as MeetingTime,
       val(initial(intersection(t1.Trip, t2.Trip))) as MeetingPlace
from Trains as t1, Trains as t2
where t1.Id < t2.Id and sometimes(t1.Trip = t2.Trip)
```

This query uses new operations

=:	$\underline{mpoint} \times \underline{mpoint}$	$\rightarrow \underline{mbool}$
sometimes:	\underline{mbool}	$\rightarrow \underline{bool}$
intersection:	$\underline{mpoint} \times \underline{mpoint}$	$\rightarrow \underline{mpoint}$

3.2.2 Design Principles

The examples have demonstrated that it is useful to have a collection of related data types and operations to obtain a query language for moving objects. The quality of such a language, that is, its ease of use and expressive power, depends on the principles applied in the design of types and operations. The following principles are observed.

- D1 For all base types of interest, there are corresponding time-dependent types.
- D2 Definitions of static and time-dependent types should be consistent.
- D3 For each time-dependent type, there are types to represent the projection to the domain and range of the respective function.
- D4 The type system has many types – to avoid a proliferation of operations, one should use generic operations as much as possible.
- D5 The space of possible operations should be explored systematically.
- D6 Operations on static and time-dependent types should be consistent.

The type system used is shown in Figure 3.2. It starts from the set of standard types *int*, *real*, *bool*, and *string*, and spatial types *point*, *points*, *line*, and *region*. All these types are made uniformly time dependent by introducing a type constructor, *moving*. It returns for a given static type α the type whose values are partial functions from the time domain into α .

More formally, let A_α denote the domain of type α , that is, the set of possible values of type α . Then the domain for type *moving*(α) is

$$A_{\text{moving}(\alpha)} := \{f \mid f : A_{\text{instant}} \rightarrow A_\alpha \text{ is a partial function}\}$$

One can observe that design rules D1 and D2 are fulfilled.

The *range* type constructor provides for a given type α , which must have a total order, the type whose values are finite sets of disjoint intervals over

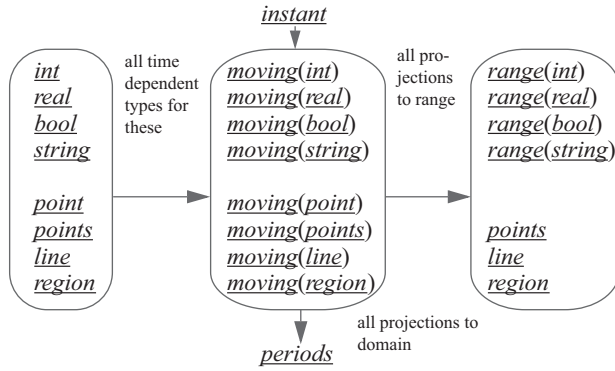


Figure 3.2 Type system.

the domain of α . So, for example, $\text{range}(\text{real})$ is a set of real-valued intervals; periods is in fact another name for $\text{range}(\text{instant})$. The types $\text{range}(\alpha)$ together with the spatial types points , line , and region are sufficient to represent the projections into the ranges for all types $\text{moving}(\alpha)$. Further, the values of all types $\text{moving}(\alpha)$ can be projected on the time axis resulting in a periods value. Hence design rule D3 is fulfilled.

The design of operations proceeds in three steps:

1. Carefully define a set of operations on the static types.
2. By a technique called *lifting*, make these operations time dependent.
3. Add some specific operations for the time-dependent types.

Lifting means to make a static operation time dependent by allowing any (combination) of its arguments to be time dependent. For example, consider the equality and intersection operations on two points. By lifting, the following signatures are available.³

$= :$	$\text{point} \times \text{point} \rightarrow \text{bool}$	intersection :	$\text{point} \times \text{point} \rightarrow \text{point}$
	$\text{mpoint} \times \text{point} \rightarrow \text{mbool}$		$\text{mpoint} \times \text{point} \rightarrow \text{mpoint}$
	$\text{point} \times \text{mpoint} \rightarrow \text{mbool}$		$\text{point} \times \text{mpoint} \rightarrow \text{mpoint}$
	$\text{mpoint} \times \text{mpoint} \rightarrow \text{mbool}$		$\text{mpoint} \times \text{mpoint} \rightarrow \text{mpoint}$

Lifted versions of these two operations are used in the last query of Section 3.2.1.

3.2.3 Implementation

In the model described so far, the semantics of time-dependent types, that is, of types $\text{moving}(\alpha)$, have been simply defined as partial functions, disregarding

³ We generally abbreviate the formally defined notation $\text{moving}(\alpha)$ by $\underline{m\alpha}$.

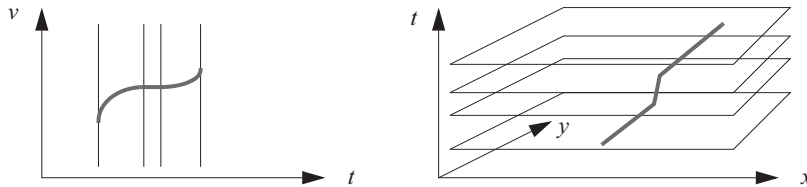


Figure 3.3 Sliced representations for *moving(real)* and *moving(point)*.

completely the issue of how such functions can be represented. A function $f : A_{instant} \rightarrow A_\alpha$ is simply an infinite set of pairs from $A_{instant} \times A_\alpha$.

We call a model where it is allowed to define the semantics of types just in terms of infinite sets an *abstract model*. An abstract model is conceptually simple and elegant, but to implement it, we have to provide a *discrete model*. In a discrete model, all the infinite sets of the abstract model have to be described in terms of finite representations.

The discrete model for the design above introduces for the time-dependent types the so-called *sliced representation*. That means that to represent a function of time, the time domain is cut into disjoint time intervals (*slices*) such that within each slice the development can be represented by some simple function of time. “Simple” actually means finitely representable. In other words, the function for a slice can be described by a few parameters rather than an infinite set of pairs. Figure 3.3 illustrates the sliced representation for a *moving(real)* and a *moving(point)*.

The representation of a single slice, consisting of the time interval and the function description, is called a *unit*. In the discrete model it makes sense to introduce explicit data types for units, for example, *upoint*, *ureal*, *ubool*. Such types are available in the SECONDO system described below.

The representations of functions within a slice (called *unit functions*) are chosen to support as many operations of the abstract model as possible in a consistent way. For a *moving(point)* a linear function of time is used. For *moving(real)*, unit functions are quadratic polynomials of time or square roots thereof. This allows one to represent the time-dependent distances between moving objects, or the development of the perimeters or sizes of moving regions, correctly.

3.3 SECONDO

In this section we describe the SECONDO DBMS prototype. In the context of this chapter, SECONDO is of interest for the following reasons: (1) It implements the model of Section 3.2. (2) It allows one to visualize and animate moving objects and the results of queries. (3) It is extensible at all levels (kernel, optimizer, and

GUI). (4) It provides data manipulation and querying at two levels, that is, not only in SQL, but also in so-called executable language. The last two features are important in highly dynamic fields such as trajectory analysis where new methods often need to be added and used even before they can be fully integrated into query optimization.

3.3.1 Overview

SECONDO is a DBMS prototype developed since about 1995 at University of Hagen. It runs on Windows, Linux, and MacOS X platforms and is freely available, open-source software.

It does not have a fixed data model. Instead, it provides a *system frame* that can be filled with implementations of different data models. The parts that are model dependent are implemented within so-called *algebra modules*. Each algebra module provides a collection of data types (type constructors, to be precise) and operations. Note that algebra modules encompass all parts of a data model implementation. Hence there are algebra modules with types for relations and tuples and query processing operations such as join methods, and there are algebra modules with types for indexes such as a B-tree or R-tree with the respective search operations.

SECONDO consists of three major components, namely, the *kernel*, the *optimizer*, and the *graphical user interface (GUI)*. These are written in different programming languages and can run as cooperating processes.

The kernel implements specific data models and is extensible by algebra modules. It provides query processing over the implemented algebras. It uses an underlying storage manager (BerkeleyDB) to provide stable storage at the level of files and records, including transaction management, locking, and recovery. The kernel is written in C++.

The optimizer is more restricted than the kernel with respect to the data model as it assumes an object-relational model (including complex attribute data types such as *mpoint*). Its core capability is cost-based conjunctive query optimization.⁴ It translates SQL queries to query plans in the executable language. The optimizer is written in Prolog.

The GUI provides an extensible graphical user interface, appropriate for an extensible DBMS. It is extensible by viewers; a viewer can provide its own graphical representation, animation, or interaction mode for a specific data type or collection of types. The GUI contains a powerful viewer for spatial and time-dependent types which itself is extensible by display methods for new data types. The GUI is written in Java.

⁴ *Conjunctive query optimization* is the fundamental problem: Given a set of relations and a set of selection and/or join predicates, determine an optimal plan.

3.3.2 Writing Queries in Executable Language

The SECONDO kernel provides a complete interface for data manipulation and querying that is data model independent. It provides the following generic commands:⁵

```
create <ident>: <type expression>
update <ident> := <value expression>
let <ident> = <value expression>
delete <ident>
query <value expression>
```

A database is essentially a collection of named objects. In the basic commands, a *type expression* is any well-formed expression over the type constructors of the active algebras, and a *value expression* is any expression involving database objects, constants, and operations of the active algebras. With the basic commands, one can `create` an object of a given type (with undefined value), one can `update` the value of an object, one can create a new object whose type and value are given by the value expression (`let`), one can `delete` an object from the database, and, finally, one can evaluate an expression and show the result at the user interface.

In Section 3.2 we have already seen example uses of the `query` and `let` commands. The `query` command has been used to evaluate expressions on atomic data types. In this section we show how expressions can actually represent efficient execution plans for a database system.

Roughly speaking, the basic idea is to write a query like an expression in relational algebra where operations are applied sequentially to obtain a query result. However, there are two important differences:

- Instead of materializing relations, for efficiency reasons individual tuples need to be passed between operations (called pipelining).
- Operations of relational algebra are descriptive in the sense that their meaning is a mathematical function telling which result relation is derived from argument relations. For example, the join operation has many different implementations. In the executable language, operations have associated fixed algorithms such as specific join methods.

Pipelining is implemented in SECONDO by providing a special type constructor called *stream*. Operations defined in an algebra can have arguments or results of

⁵ We only show the basic commands for data manipulation; there are further commands for inquiries about the system or the database, transactions, import and export, etc.

type *stream*(x). Implementation of operators and evaluation through the query processor is then set up to pass arguments via pipelining.

The following example query implements a simple selection on the `Trains` relation:

```
query Trains feed filter[.Trip passes mehringdamm] consume
```

Operations are written in postfix notation. Operation **feed** passes tuples from a relation into a stream. **filter** evaluates a predicate on each tuple of a tuple stream. **consume** collects a tuple stream into a relation. These three operations have signatures:

feed:	$rel(\text{tuple})$	\rightarrow	$stream(\text{tuple})$
filter:	$stream(\text{tuple}) \times (\text{tuple} \rightarrow \text{bool})$	\rightarrow	$stream(\text{tuple})$
consume:	$stream(\text{tuple})$	\rightarrow	$rel(\text{tuple})$

Here `tuple` is a type variable representing some tuple type.

The following query is an example use of a hashjoin operation:

```
query Trains feed {t1} Trains feed {t2} hashjoin[Line_t1, Line_t2]
count
```

The notation `{t1}` denotes a renaming that appends the string `_t1` to every attribute name, to make attribute names of the two arguments for the hashjoin distinct.

Various index types are also provided by some algebra modules, such as B-trees or R-trees. The following command creates a B-tree index on attribute `Id` of the `Trains` relation:

```
let Trains_Id_btree = Trains createbtree[Id]
```

It can then be used to retrieve a train with a given `Id`, say 50:

```
query Trains_Id_btree Trains exactmatch[50] consume
```

To summarize this section, `SECONDO` has a precise textual language to describe query plans. Queries in executable language are completely syntax- and type-checked and errors reported.

3.3.3 Writing Queries in SQL

One can also write queries in SQL and use query optimization. One has to observe some small notational differences as the `SECONDO` optimizer is programmed in

Prolog and the queries written are in fact Prolog terms. Still, they look quite similar to regular SQL.

The last query from Section 3.2 can be entered into SECONDO as follows:

```
select [t1:id, t1:line, t2:id, t2:line,
       inst(initial(intersection(t1:trip, t2:trip))) as meetingtime,
       val(initial(intersection(t1:trip, t2:trip))) as meetingplace]
from [trains as t1, trains as t2]
where [t1:id < t2:id, sometimes(t1:trip = t2:trip)]
```

The main differences in notation are that lists need to be written in square brackets, colon is used instead of period for qualified attributes, and names of relations and attributes need to be written in lower case. Further, the `where` clause is generally a conjunction of predicates, separated by commas rather than a single Boolean expression.

The optimizer provides cost-based query optimization and produces a plan in SECONDO executable language. For the query above, the following plan is constructed.

```
query Trains feedproject[Id, Line, Trip] {t1}
  Trains feedproject[Id, Line, Trip] {t2}
  symmjoin[sometimes(.(Trip_t1 = ..Trip_t2))]
    {0.0238913, 0.350099}
  filter[.(Id_t1 < .Id_t2)] {0.517808, 0.00916338}
  extend[
    Meetingtime: inst(initial(intersection(.Trip_t1, .Trip_t2))),
    Meetingplace: val(initial(intersection(.Trip_t1, .Trip_t2)))]
  project[Id_t1, Line_t1, Id_t2, Line_t2, Meetingtime,
    Meetingplace]
  consume
```

Due to the space limitation we will not explain this plan in detail. In addition to query operations, the optimizer also inserts annotations into the plan such as the selectivity of predicates and the cost of evaluating them. These are used for query progress estimation during execution.

The user may enter this query plan directly and have it executed without involving the optimizer. After evaluation, the result of the query is presented at the user interface.

3.3.4 Visualization and Animation of Data Sets and Results

The graphical user interface is extensible by viewers. The Hoese-Viewer is specialized in displaying spatial data and animating moving objects. It can

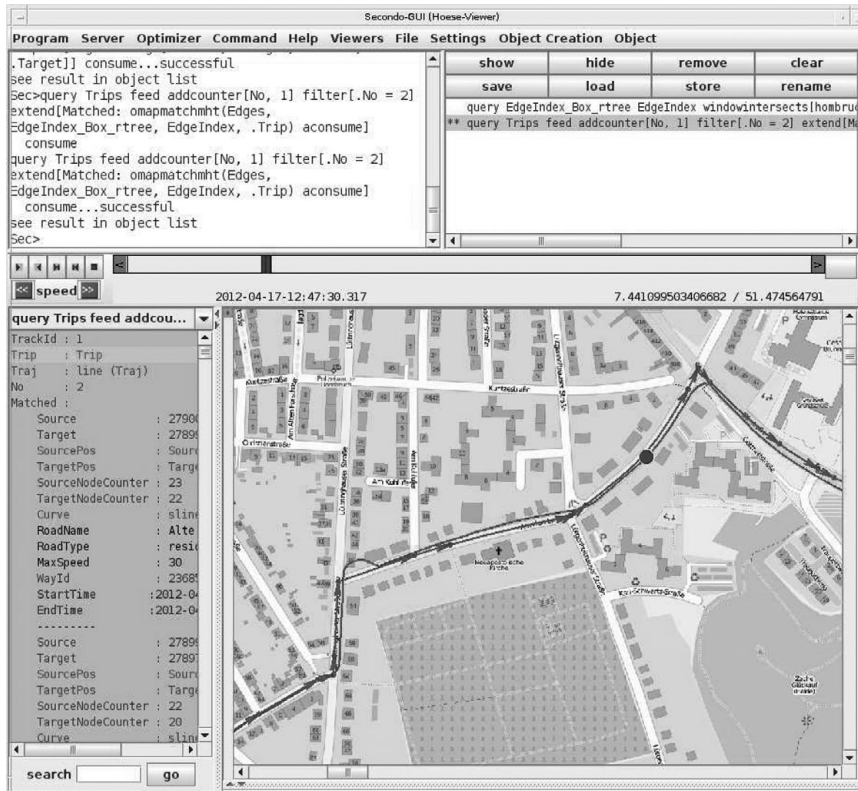


Figure 3.4 Map matching based on network built from OSM data.

also show map backgrounds from tiled map servers such as OpenStreetMap or GoogleMaps. Figure 3.4 shows a map-matched trajectory. Map matching (see Chapter 2) was done based on a directed graph representation of the road network constructed within *SECONDO* from OpenStreetMap source data. The original trajectory of the *mpoint* is displayed together with the sequence of edges obtained from map-matching. The black circle indicates the current position of the moving object during the animation. The current time and coordinates are shown at the top of the viewer.

3.4 Representations for Sets of Trajectories

Storing and analyzing trajectories relies on methods to represent trajectory data within a database. In this section, we show how trajectory data can be loaded and represented in *SECONDO*. The DB commands are presented in the *SECONDO* executable language.

3.4.1 Loading Data

First, we show how the raw trajectory data from a CSV text file `Traj.csv` can be imported to a `SECONDO` database.⁶ We assume the text file has the schema

```
(Id: int, Line: int, Up: bool, Time: instant, PosX: real,
  PosY: real)
```

which would correspond to raw data observed for the `Trains` relation introduced in Section 3.2.

```
let TrainsRaw = [const rel(tuple([Id: int, Line: int, Up: bool,
  Time: instant, PosX: real, PosY: real])) value ()]
  csvimport['Traj.csv', 0, "", "", ""]
  projectextend[Id, Line, Up, Time; Pos: makepoint(.PosX, .PosY)]
  consume;
```

This creates a relation

```
TrainsRaw(Id: int, Line: int, Up: bool, Time: instant, Pos: point)
```

where attribute `Pos` contains the position data as a *point* (easting, northing). In the following, we briefly investigate two different ways to represent trajectories more effectively according to the data model of Section 3.2 within `SECONDO`: the *compact representation* and the *unit representation*.

3.4.2 Compact Representation

In `TrainsRaw` the information on a vehicle is distributed among many tuples. Using the model of spatio-temporal data types (Section 3.2), we now express the same data in a relation with only a single tuple per vehicle. The data type *mpoint* is used to capture the temporal development of attribute `Pos`. We achieve this by grouping `TrainsRaw` by `Id` and applying the **approximate** operator to each group. Using `Time` as the least significant sorting criterion prior to grouping guarantees that the positions for each train enter the **approximate** operator in increasing temporal order:

```
let Trains = TrainsRaw feed
  sortby[Id, Line, Up, Time]
  groupby[Id, Line, Up; Trip: group feed approximate[Time, Pos] ]
  consume;
```

The result is the relation `Trains` with the schema shown earlier. This is what we call the *compact representation* of moving object data. It is easy to apply

⁶ It is also possible to import NMEA recordings using an operator **mneaimport**.

many different kinds of temporal and spatio-temporal operations as introduced in Section 3.2 to the temporal attribute.

3.4.3 Unit Representation

A second way to represent moving object data is to employ the respective unit types. Several operations allow one to transform a value of a type *moving*(α) to a stream of values of the corresponding unit type and vice versa, so it is easy to translate between, say, an *mpoint* and a set of *upoints* and to use both kinds of data types together. A *upoint* represents a single time interval and a linear movement of a single object during this time. Let us create the unit representation for relation `Trains`:

```
let UnitTrains = Trains feed
  projectextendstream[Id, Line, Up; UTrip: units(.Trip)]
  addcounter[No, 0] consume;
```

The result is a relation

```
UnitTrains(Id: int, Line: int, Up: bool, UTrip: upoint, No: int)
```

For each vehicle identifier, `UnitTrains` contains a set of tuples, each of which contains one of the temporally disjoint units whose union forms the train's complete trajectory.

The **units** operator converts each *mpoint* to a stream of *upoints*, and **projectextendstream** creates one copy of the input tuple, projected on the attributes listed, for each *upoint* value. The **addcounter** operator extends the tuples with a counter attribute called `No`, starting from 0. Because this *unit representation*, as we call it, replicates attributes `Id`, `Line`, and `Up`, it is less space efficient. However, it has a higher degree of organization than `TrainsRaw`, and is quite useful when creating indexes supporting certain query types.

3.5 Indexing

Indexing, of course, has been a major research topic in the field of spatio-temporal databases (also termed moving objects databases) and it is beyond the scope of this chapter to treat the issue at any depth. Surveys describing and classifying an impressive number of proposed structures are mentioned in Section 3.8.

A major distinction concerns indexing current and expected near future movement versus indexing histories of movement, or trajectories. In the context of trajectory databases only the latter case is of interest. Further, one can distinguish whether movement is described relative to the Euclidean plane (that is, by

(x, y) coordinates) or relative to a network, called *free* and *network-constrained* movement, respectively.

Structures proposed to index free movement include the STR-tree and the TB-tree. Both are R-tree variants with the goal of keeping 3D line segments of the same trajectory (units in the terminology of this chapter) clustered together on pages. Whereas the STR-tree modifies the insertion and split strategy of the R-tree toward this goal, the TB-tree does this in a more radical way and ensures that a leaf page contains only segments of the same trajectory.

Besides such specialized structures, regular R-trees can also be used to index the spatial, temporal, or spatio-temporal dimensions.

In network-constrained movement, the position of a moving object is described relative to an edge of the network graph or a path in the network. Two index structures for this case are the FNR-tree and the MON-tree.

Secondo includes implementations of the R-tree as well as of the TB-tree and the MON-tree. In most applications, for example, the BerlinMOD benchmark (see Section 3.8), just R-trees are used. Generally, the index serves to retrieve sets of candidates based on bounding box comparisons, which need to be further checked for exact fulfillment of a query predicate. This is the so-called filter-and-refine strategy.

When indexing moving points by R-trees, different granularities can be chosen. The roughest one is to index the *mpoint* as a whole. If an *mpoint* was observed over a long period, its bounding box may be very large, leading to a lot of dead space within the index. The index will contain only a few entries, but its selectivity is bad; this means the resulting candidate set will contain a lot of false hits. The other extreme is to index single units of the *mpoint*. Here, compared with indexing of the whole *mpoint*, less dead space is produced. But the complete *mpoint* is distributed over many index entries. A third way is indexing groups of connected units. All three possibilities are available in SECONDO.

3.6 Hermes

Another system dealing with moving objects is *Hermes*. It is implemented on top of the Oracle 10g database system using PL/SQL as a programming language. Beside the core system of Hermes, there is an implementation of a web-based query builder and viewer. Hermes does not implement own data structures for spatial objects, rather it uses the spatial objects of the underlying system.

Because Hermes implements the same data model as SECONDO does, the data types and operations on them are quite similar. Additionally to the types provided by SECONDO, Hermes has implementations for moving circles, moving rectangles, and moving collections (sets of moving objects of different types).

Like SECONDO, Hermes uses the sliced representation for representing moving objects. Units belonging to a moving object are stored within a nested table.

Besides the moving data types, Hermes contains a TB-tree implementation. This structure supports the standard operations for this index (point query and range query), but also k-NN and similarity queries.

Hermes' query language is SQL extended by spatio-temporal operations. Although SQL is familiar to most database systems' users, formulating complex temporal queries in SQL is a hard task and queries tend to degenerate to deeply nested function calls.

3.7 Conclusions

In this chapter, we have motivated a high-level conceptual model of trajectories as continuous functions, represented by abstract data types. These serve as a foundation to extend the data model and query language of a DBMS to support representation and querying of movement data. We have shown how queries can be formulated in this framework. The implementation within a DBMS prototype was sketched.

3.8 Bibliographic Notes

The field of moving objects databases is covered in depth in the textbook by [Güting and Schneider \(2005\)](#). The data model of Section 3.2 was developed in a series of papers. In [Güting et al. \(2000\)](#), the type system and operations are carefully designed. Further papers define the discrete model and develop algorithms for the operations (see [Güting and Schneider, 2005](#), for references). The model was extended to a network-based representation of moving objects (or trajectories) in [Güting et al. \(2006\)](#). Recently, it was generalized to model objects moving in different environments (for example, road networks, public transport, indoor spaces) and according to different transportation modes ([Xu and Güting, 2013](#)).

The *SECONDO* system is freely available for download from its Web site,⁷ where a lot of further documentation can be found.

Survey articles on spatio-temporal indexing are [Mokbel et al. \(2003\)](#) and [Nguyen-Dinh et al. \(2010\)](#). The TB-Tree is described in [Pfoser et al. \(2000\)](#), the MON-tree in [Almeida and Güting \(2005\)](#). The Hermes system, which also partially implements the model of Section 3.2, is described in [Pelekis and Theodoridis \(2005\)](#) and [Pelekis et al. \(2008a\)](#).

SECONDO supports further query types such as continuous nearest neighbor queries ([Güting et al., 2010](#)) and spatio-temporal pattern queries ([Sakr and Güting, 2011](#)). The latter are discussed in Chapter 12.

⁷ <http://dna.fernuni-hagen.de/Secondo.html/>

BerlinMOD is a benchmark for evaluating MOD systems, implemented within *SECONDO*. It allows one to create scalable trajectory data sets. It is based on a simulation approach: For 2000 (fictitious) people living in Berlin, their car trips are “observed” over a period of one month. The mentioned parameters define the standard benchmark at scale factor 1.0. However, one can set parameters to select any number of people and length of observation period. The benchmark further defines a set of representative queries to evaluate the performance of a MOD system. The BerlinMOD benchmark is presented in [Düntgen et al. \(2009\)](#). Its Web site⁸ provides scripts and further documentation.

⁸ <http://dna.fernuni-hagen.de/Secondo.html/BerlinMOD/BerlinMOD.html>

4

Trajectory Data Warehouses

Alejandro A. Vaisman and Esteban Zimányi

4.1 Introduction

In previous chapters we have seen that the usage of location-aware devices enables the collection of large volumes of trajectory data. Effective analysis of such data imposes new challenges for their management, while raising opportunities for discovering behavioral patterns that can be exploited in applications such as location-based services or traffic control management.

Data warehouses (DW) and online analytical processing (OLAP) have been successfully used for transforming detailed data into valuable knowledge for decision-making purposes. Extending DWs for coping with trajectory data, leading to trajectory data warehouses (TDW), allows us to extract essential knowledge from raw or semantic trajectories. For example, a TDW can be used for analyzing the average speed of cars in different urban areas.

Trajectory data in a warehouse must be typically analyzed in conjunction with other data, for example, to find out the correlation between the speed of cars and temperature, precipitation, or elevation. In light of these needs, in this chapter we provide an overall view that integrates trajectory data in a more general data warehousing framework, which we call *spatio-temporal data warehousing*.

We start this chapter by introducing in Section 4.2 the notion of data warehousing and describing the main elements in a DW architecture. After giving in Section 4.3 the running example used throughout this chapter, we address in Section 4.4 spatio-temporal data warehousing, and show that trajectory data warehouses can be regarded as a particular case of spatio-temporal DW. We introduce in Section 4.5 continuous fields and show that they enhance the possibilities of decision making. In Section 4.6 we discuss a representative TDW, the one proposed by the GeoPKDD project. We conclude in Section 4.7.

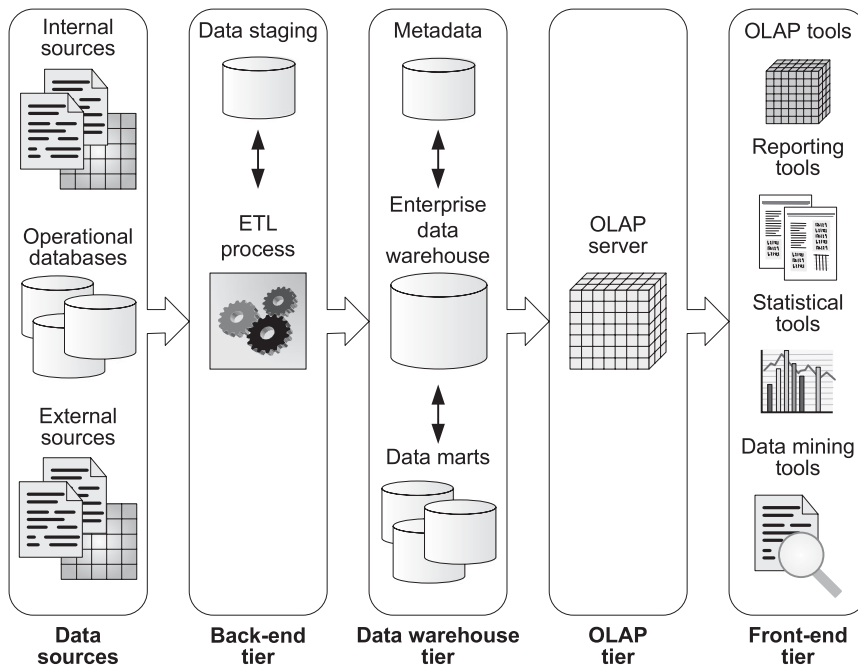


Figure 4.1 A typical data warehouse architecture.

4.2 Data Warehousing

Data warehouses are large data repositories that support the decision-making process. Figure 4.1 shows a typical multi-tier data warehousing architecture. We can see that data coming from heterogeneous data sources, after a staging process that acts as a kind of buffer, pass through a process known as ETL, standing for *extraction, transformation, and loading*. The *extraction* phase gathers data from the data sources. These may be operational databases, but also files in various formats, which may be internal or external to the organization. The *transformation* phase modifies the data from the format of the data sources to that of the warehouse. This includes several aspects: cleaning, which removes errors in the data and converts them into a standardized format; integration, which reconciles data from different data sources, both at the schema and at the data level; and aggregation, which summarizes the data obtained from data sources according to the level of detail (granularity) of the data warehouse. Finally, the *loading* phase feeds the data warehouse with the transformed data. This also includes refreshing the data warehouse, that is, propagating updates from the data sources to the data warehouse at a specified frequency in order to provide up-to-date data for the decision-making process. We will see later that

the trajectory reconstruction process explained in Chapter 2 is part of the ETL process in a TDW architecture.

Continuing with Figure 4.1, a DW makes use of metadata, which include information about the DW schema, the data source schemas, the mappings between source and DW attributes, as well as the frequency of data refreshment. From the organizational DW smaller DWs can be built to satisfy departmental needs. These DWs are called *data marts*.

On the next tier, an OLAP server provides a multidimensional view of the data stored in the DW. This enables analysts, managers, and executives to gain insight into data through interactive access to a wide variety of possible views of information. Thus, at a *conceptual level*, data are perceived by the user as a hypercube where each cell contains values, called *measures*, which quantify *facts*. The axes of the hypercubes are called *dimensions*. Dimensions are typically organized into *hierarchies*, which allow to aggregate measures at different levels of detail. Queries addressed to the OLAP server are expressed using OLAP operators such as slice, dice, roll-up, and drill-down. The *slice* operator removes a dimension in a cube, that is, obtains a cube of $n - 1$ dimensions from a cube of n dimensions. This is analogous to a relational algebra projection. *Dice* applies a Boolean condition to a cube, and returns another cube containing only the cells that satisfy such condition. This is analogous to a relational algebra selection. *Roll-up* aggregates measures according to a dimension hierarchy, using an aggregate function, to obtain measures at a coarser granularity. *Drill-down* disaggregates previously summarized measures, and can be considered the inverse of a roll-up.

Finally, the user interacts with the OLAP server through several tools, such as OLAP, reporting, statistical, and data-mining tools. In the case of an OLAP client, the user can then perform OLAP analysis interactively.

If a DW stores trajectory data, we are in the presence of a TDW. Typical analysis over a TDW includes finding out the distribution of trajectories by road type (which requires a roll-up operation to aggregate trajectories by road type, and a slice operation to keep the dimensions of interest), or the total number of cars in a certain location at a given moment. We will give examples of TDW queries in the following sections.

At the *logical level*, a typical implementation, referred to as relational OLAP (ROLAP), stores the data in relational databases. This leads to two kinds of tables. *Fact tables* store the data elements under analysis (e.g., trajectories in a TDW), while *dimension tables* describe the axes of analysis (e.g., roads, vehicle type) of the data contained in the fact tables. If dimension tables are denormalized, that is, there is a single table for the whole dimension, we have a *star schema*. Otherwise, that is, if there is one table for each level in a dimension hierarchy, we have a *snowflake schema*. Fact tables are usually normalized.

analyzing the facts. In our case we would like to analyze the trajectories by days, districts, roads, and the cars that performed the trajectory. Therefore, we need to segment the trajectories into *episodes* such that each episode is related to a single district, road, and day. Nevertheless, since we need to keep track of all episodes belonging to a single trajectory, we define an additional dimension that groups the data pertaining to each trajectory as a whole.

As shown in the figure, there is a fact relationship, `Episode`, that is related to five dimensions: `Time`, `District`, `Road`, `Trajectory`, and `Car`. Dimensions are composed of levels and hierarchies. For example, while the `Road` dimension has only one level, the `District` dimension is composed of two levels, `District` and `Zone`, with a one-to-many parent-child relationship defined between them. Levels have attributes that describe their instances, referred to as members. For example, level `District` has attributes such as `name`, `population`, and `area`. A level or an attribute can be *spatial*, that is, it has an associated geometry (e.g., point, line, or region) that is indicated by a pictogram. In our example, dimension levels `District` and `Zone` are spatial, and their geometry is a region; dimension `Road` is also spatial, and its geometry is a line. On the other hand, `startLocation` and `endLocation` are spatial attributes of the `Trajectory` dimension, and their geometry is of type point.

There are four measures: `route`, `distance`, `duration`, and `avgSpeed`. The first one, `route`, keeps the movement track of the episode. It is a *spatio-temporal measure* of type time-dependent (or moving) point, as indicated by the symbol $m(\bullet)$. The other measures are numerical ones, derived from `route`.

Finally, topological relationships may be represented using pictograms in fact relationships and in parent-child relationships. For example, the topological relationship in `Episode` indicates that whenever a district and a road are related in an instance of the relationship, they must overlap. Similarly, the topological relationship in the hierarchy of dimension `District` indicates that a district is covered by its parent `Zone`.

As stated before, the movement tracks of episodes are kept in measure `route`, while data describing the whole trajectories are kept in dimension `Trajectory`. Alternatively, we could have represented episodes or even whole trajectories in a dimension. Our model is flexible enough to represent a wide spectrum of situations, where trajectories can be aggregated along spatial and alphanumeric dimensions, or facts can be aggregated over a trajectory dimension. The choice among these representations depends on the queries to be addressed. Indeed, the *complexity* of the queries and their *execution time* will depend on how much the information requested is precomputed in measures, as data warehouses are optimized for aggregating measures along dimensions. In other words, although it is possible to aggregate data from dimensions, queries will be more elaborate to write, and less efficient to execute.

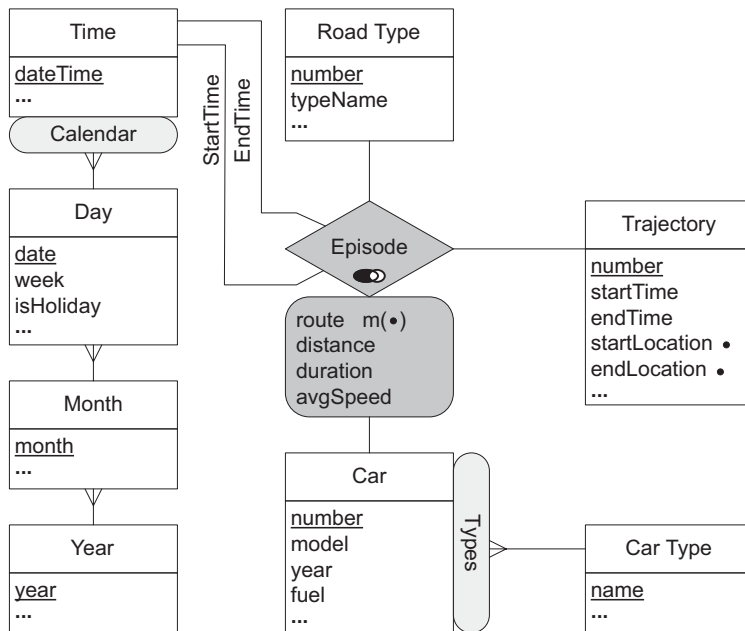


Figure 4.3 An alternative partition of episodes with respect to road type.

The DW depicted in Figure 4.2 partitions a trajectory into *episodes* with respect to days, roads, and districts. An alternative schema shown in Figure 4.3 partitions trajectories with respect to the road type in which they occur. For example, a trajectory can be segmented into episodes occurring in highways, national roads and regional roads. This partitioning is close to the notion of episodes discussed in Chapter 1. Notice also that the time granularity in Figures 4.2 and 4.3 differs. In the former case, the granularity is day, although we keep the movement track in the `route` measure with a timestamp granularity. In the latter case, we relate each episode with its initial and final timestamps. The choice among the two alternative data warehouse schemas depends on application requirements and the typical OLAP queries to be addressed.

When trajectories are used as measures, the problem of aggregation arises. In the examples of Figures 4.2 and 4.3, we segmented the trajectories into episodes and kept their movement track in a geometry of type time-dependent point. Thus, we can aggregate such episodes (or the whole trajectories) along the different dimensions. An alternative approach for trajectory aggregation aims at identifying “similar” trajectories and merging them in a class. This aggregation may come together with an aggregate function, which may be the `count` function in the simplest case, although more complex ones may be used. The main problem consists in adopting an appropriate notion of *trajectory similarity*, through the definition of a similarity measure, for example, a *distance*

function. The simplest approach to define similarity between two trajectories is viewing them as vectors and using the Euclidean distance as similarity measure. The problem with this technique is that it cannot be easily applied to trajectories having different length or sampling rate (see Chapter 2), and it is not effective in the presence of noise in the data. A typical way of aggregating trajectories is clustering them together, considering different distance functions or other characteristics (e.g., same starting point, same ending point, etc.). Discovering trajectories with the same pattern is another way of aggregating trajectories. This is extensively covered in Chapters 6, 7, and 8 of this book.

Finally, in Section 4.6 we will study an alternative design of a trajectory data warehouse, where space and time are partitioned into spatio-temporal cells, and where each cell contains aggregated measures that are precomputed from the trajectories that cross the cell. Examples of such aggregated measures would be the number of trajectories or their average speed. In this way, the movement tracks of individual trajectories are no longer stored in the data warehouse, only aggregated data about the trajectories are kept.

4.4 Querying Trajectory Data Warehouses

In order to address queries to our TDW we translate the conceptual schema in Figure 4.2 into a snowflake schema. `Episode` becomes a fact table, dimension levels become dimension tables (with identifier `id`), and foreign keys are used for linking the fact table to dimension tables, and to link dimension tables that represent two consecutive levels in a dimension hierarchy. For example, the hierarchical relationship between `District` and `Zone` is represented by the attributes `zone` in the former and `id` in the latter, where `id` is a foreign key referencing `zone`. The resulting schema is given next.

```
Episode(time, district, road, trajectory, car, route, distance,
        duration, avgSpeed)
Time(id, date, week, isHoliday, ..., month)
Month(id, month, ..., quarter)
Quarter(id, quarter, ..., year)
Year(id, year, ...)
District(id, name, population, area, ..., zone)
Zone(id, number, name, ...)
Road(id, number, name, ...)
Trajectory(id, number, distance, duration, startTime, endTime,
           startLocation, endLocation, ...)
Car(id, number, model, year, fuel, ..., carType)
CarType(id, name, ...)
```

4.4.1 OLAP Queries

We use a functional SQL-like query language for expressing OLAP queries. This language, denoted by \mathcal{Q}_{agg} , is based on the well-known relational calculus with aggregate functions proposed by Klug. We show next what a \mathcal{Q}_{agg} query looks like, using our running example.

Query 4.1. “Give by zone the total number of episodes performed by diesel cars in February 2011.”

```
SELECT z.number, nbrEpisodes
FROM Zone z
WHERE nbrEpisodes = COUNT( SELECT e.id
    FROM Episode e, Car c, Time t, District d
    WHERE e.car = c.id AND e.time = t.id
    AND e.district = d.id AND d.zone = z.id
    AND c.fuel = 'diesel' AND t.date >= 1/2/2011
    AND t.date < 1/3/2011 )
```

For each zone, the inner query counts the number of trajectories in the zone satisfying the conditions in the query, and the result is stored in the variable `nbrEpisodes`. Notice that the inner query performs in the `WHERE` clause a *dice* operator by selecting facts with diesel cars in February 2011. The only attribute in the `SELECT` clause of the inner query is the identifier of the episodes. This corresponds to a series of *slice* operators removing all dimensions associated with the facts. Finally, the correlation between the inner and the outer queries through districts performs a *roll-up* operator.

The query just presented involved the fact table `Episode`. We give next an example of an OLAP query involving the `Trajectory` dimension.

Query 4.2. “Give the average duration of trajectories that traversed the Lambrate district in the last quarter of 2010.”

```
AVG( SELECT j.duration
    FROM Trajectory j
    WHERE EXISTS ( SELECT *
        FROM Episode e, District d, Time t
        WHERE e.trajectory=j.id AND e.district=d.id
        AND e.time=t.id AND d.name='Lambrate'
        AND t.date >= 1/10/2010 AND t.date <= 31/12/2010 ) )
```

Here, for each instance of the `Trajectory` dimension, the inner query verifies that at least one episode of the trajectory is related to the Lambrate district and occurred on the last quarter of 2010. Notice that the durations of the trajectories are precomputed in the `Trajectory` dimension and therefore it is possible to

apply the average function to them. If the durations of the whole trajectories must be calculated, then the query would be as follows.

```
AVG( SELECT totDuration
      FROM Trajectory j
      WHERE EXISTS ( SELECT *
                    FROM Episode e, District d, Time t
                    WHERE e.trajectory=j.id AND e.district=d.id
                    AND e.time=t.id AND d.name='Lambrate'
                    AND t.date >= 1/10/2010 AND t.date <= 31/12/2010 )
      AND totDuration = SUM( SELECT e.duration
                            FROM Episode e WHERE e.trajectory=j.id ) )
```

As can be seen in the examples above, an OLAP query is just a relational calculus query with aggregation.

To characterize OLAP queries we consider a set of *base types*, namely *int*, *real*, *bool*, and *string*, with the usual interpretation, except that their value may be undefined. In addition, we define an *identifier type* *id* (introduced in the examples above), which is used to identify dimension level members. There are also *time types*, which are *instant* and *periods*, the latter being a set of time intervals. Finally, there is a type constructor $\text{range}(\alpha)$, where $\alpha \in \{\text{int}, \text{string}, \text{bool}, \text{real}, \text{instant}\}$, which yields sets of intervals over α . Thus, the type *periods* is just a shorthand notation for $\text{range}(\text{instant})$. Base and time types have an associated set of operations, defined in Chapter 3.

It can be proved that the language \mathcal{Q}_{agg} , defined over the sets of base and time types, has the same expressive power of the relational calculus extended with aggregate functions. Based on this, it follows that the class of OLAP queries includes all the queries that are expressible by \mathcal{Q}_{agg} . Therefore, a *data warehouse* is a data repository that supports OLAP queries.

4.4.2 Spatial OLAP

We consider now the *spatial data types* *point*, *points*, *line*, and *region*, with their associated operations. For example, the predicate *inside* can be used to test whether a point is inside a region. To express the following query, we need to extend \mathcal{Q}_{agg} with spatial data types.

Query 4.3. “For roads intersecting the Lambrate district, give the number of trajectories in the last quarter of 2010.”

```
SELECT r.name, nbTrajs
FROM Road r, District d
WHERE d.name='Lambrate'
AND intersects(r.geometry,d.geometry)
```

```

AND nbTrajs = COUNT( SELECT e.trajectory
FROM Episode e, Time t
WHERE e.road=r.id AND e.time=t.id
AND t.date >= 1/10/2010 AND t.date <= 31/12/2010 )

```

The outer query selects the roads that intersect the Lambrate district using the `intersects` predicate, which determines if a pair of geometries intersect. Then, the inner query (an OLAP query as defined above) joins the fact table `Episode` with the dimension level `Time`, selects the episodes that occurred in the last quarter of 2010 and in the road of the outer query, counts the number of trajectories, and stores this number in the variable `nbTrajs`.

\mathcal{Q}_{agg} augmented with spatial types yields the class of *Spatial OLAP (SOLAP) queries*. As a consequence, we denote *spatial data warehouse* as a data warehouse that supports SOLAP queries.

4.4.3 Spatio-Temporal OLAP

As explained in Chapter 3, *time-dependent types* are obtained by applying the type constructor `moving(·)` to a *base* or *spatial* type. For example, a value of type `moving(point)` is a continuous function with signature $f : \text{instant} \rightarrow \text{point}$. Time-dependent types are partial functions, that is, they may be undefined for certain periods of time. Time-dependent types are equipped with a set of operations, also defined in Chapter 3. For example, the projection of a time-dependent point into the plane consists of the points and lines returned by the operations `locations` and `trajectory`, respectively. Further, all operations over a nontemporal type are *lifted* to allow any of the argument types to be a time-dependent type and returns a time-dependent type. As an example, the `distance` function, with signature `point × point → real`, has lifted versions where one or both of its arguments can be time-dependent points and the result is a time-dependent real. Intuitively, the semantics of such lifted operations is that the result is computed at each time instant using the nonlifted operation.

Analogously, *aggregation* operators can also be lifted. For example, a lifted `avg` operator combines a set of time-dependent reals describing velocity for several cars, and results in a new time-dependent real where the average is computed at each instant. In addition, *time-dependent aggregation* operators compute a scalar value from all the values taken by a time-dependent type. For example, operator `mavg` can be used to obtain the average value from a time-dependent real describing velocity.

Spatio-temporal OLAP (ST-OLAP) accounts for the case when the spatial objects evolve over time. Thus, to express the following query, we need to extend \mathcal{Q}_{agg} with both spatial types and the time-dependent types introduced above.

Query 4.4. “For each road, give the geometry of the segments of the road on which at least one trajectory passed on May 1, 2012.”

```
SELECT r.name, travGeom FROM Road r
WHERE travGeom = UNION( SELECT trajectory(e.route)
    FROM Episode e, Time t WHERE e.road=r.id
    AND e.time=t.id AND t.date=1/5/2012 )
```

In this query we apply the `trajectory` operation to the `route` measure (of type time-dependent point) in order to obtain a line containing all the points traversed by the time-dependent point. Then, we perform a spatial union on all the geometries thus obtained, and store the result in the variable `travGeom`.

We next present another example of a spatio-temporal OLAP query.

Query 4.5. “Give the number of trajectories that started in the Lambrate district on May 1, 2012.”

```
COUNT( SELECT j.id
    FROM Trajectory j, District d
    WHERE d.name='Lambrate' AND date(j.startTime)=1/5/2012
    AND intersects(j.startLocation,d.geometry) )
```

Notice that because `j.startTime` returns a timestamp, the `date` function is applied for obtaining the corresponding day. The query takes advantage of the fact that the start time and the start location of trajectories are precomputed in the `Trajectory` dimension. If this were not the case, the query would read:

```
COUNT( SELECT e.id
    FROM Episode e, District d WHERE d.name='Lambrate'
    AND inst(initial(e.route)) =
        MIN( SELECT inst(initial(e1.route)) FROM Episode e1
            WHERE e1.trajectory=e.trajectory )
    AND date(inst(initial(e.route)))=1/5/2012
    AND intersects(val(initial(e.route)),d.geometry) )
```

In this case, the first episode of a trajectory is selected by verifying that the start time of the episode given by `inst(initial(e.route))` is the smallest among all those of the episodes composing the trajectory. Then, it remains to be tested that the start instant of the episode is on May 1, 2012, and that the start location of the episode given by `val(initial(e.route))` intersects the geometry of the Lambrate district. Because `inst(initial(e.route))` returns a timestamp, `date` is applied for obtaining the corresponding day.

Based on the above, we define the class of *spatio-temporal OLAP (ST-OLAP) queries* as the one composed of all the queries that can be expressed by \mathcal{Q}_{agg} ,

augmented with spatial and time-dependent types. Therefore, a *spatio-temporal data warehouse* is a warehouse that supports ST-OLAP queries.

As we have stated in the introduction, a *trajectory data warehouse* is a particular case of spatio-temporal data warehouse, where the facts are trajectories, part of trajectories, or some aggregation of trajectories or parts of trajectories.

4.5 Continuous Fields

Continuous fields are phenomena that change continuously in space and/or time. Examples include altitude and temperature, where the former varies only on space and the later varies on both space and time. Continuous fields have been extensively studied in GIS, although multidimensional analysis of continuous fields is a novel area of research. We will show in this section that combining trajectory data with continuous field data provides additional analysis capabilities for decision making.

At a *conceptual level* continuous fields can be represented as a function that assigns to each point of space (and possibly in time) a value of a particular domain (e.g., integer for altitude). However, at a *logical level*, continuous fields must be represented in a discrete way. For this, we need first to discretize the space, that is, to partition the spatial domain into a finite number of elements (what is called a *tessellation*), and then assign a value of the field to a representative point in each partition element. Furthermore, because values of the field are known only at a finite number of points (called *sampled points*), the values at other points must be inferred using an *interpolation function*. In practice, different tessellations and different interpolation functions may be used. The most popular representation is the *raster* tessellation, which partitions the space in regular elements (squares, cubes, etc.) and assigns the same value to each point belonging to an element.

We extend next our conceptual model with continuous fields, independently of their underlying implementation. Fields can be seen as two- or three-dimensional cubes with a single measure. For example, a time-dependent field representing temperature can be seen as a spatio-temporal cube that associates a real value to any given point in space and time. This view of fields as cubes allows us to seamlessly combine fields with regular cubes composed of fact relationships and dimensions. As we will see in the queries below, relating fields to fact relationships or to dimensions is performed through spatial or spatio-temporal operators. Fields can also be included as measures in fact relationships, although this is beyond the scope of this chapter.

Figure 4.4 extends our example with continuous fields. *Nontemporal* fields are identified by the $f(\text{📍})$ pictogram, while *time-dependent* ones are identified by the $f(\text{📍,🕒})$ pictogram. There are two nontemporal fields, `Elevation` and `LandUse`. The former could be used, for example, for analyzing the correlation between

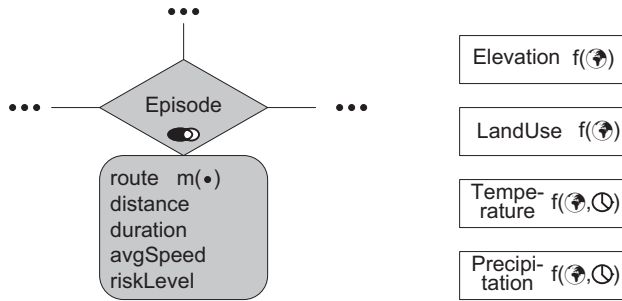


Figure 4.4 Extending our running example with continuous fields.

speed of trajectories and elevation (or slope), and the latter to select trajectories starting in a residential area and finishing on an industrial area. Further, there are two time-dependent fields, *Temperature* and *Precipitation*. In addition, numerical measures can be calculated from field data. An example is given by measure *riskLevel*, which represents knowledge from domain experts about the relative risk of the episodes. Such a measure (e.g., a real value) can be computed from the measure *route* and the four fields. For example, an episode with high speed in descending slopes, in residential areas, with frozen temperatures, or with high precipitation will have high a risk level.

To be able to express OLAP queries involving fields, we define *field types*, which capture the variation in space of base types. They are obtained by applying a constructor *field*(\cdot). Hence, a value of type *field*(*real*) (e.g., representing altitude) is a continuous function $f : \text{point} \rightarrow \text{real}$. Field types have associated operations, which are analogous to those defined for time-dependent types in Chapter 3. In particular, field types have *lifted* operations that generalize those of the base types. Their semantics is such that the result is computed at each point in space using the nonlifted operation. *Aggregation* operators are also lifted. For instance, a lifted *avg* operator combines several fields, yielding a new field where the average is computed at each point in space. In addition, *field aggregation* operators compute a scalar value from all the values taken by a field. For example, operator *favg* can be used to obtain the average value from a field describing altitude.

Time-dependent fields are obtained by composing the *moving* and *field* type constructors. For example, a value of type *moving*(*field*(*real*)), which defines a function $f : \text{instant} \rightarrow (\text{point} \rightarrow \text{real})$, can be used to represent temperature, which varies on time and space. In our model the types *moving*(*field*(*real*)) and *field*(*moving*(*real*)) are equivalent, that is, they define a spatio-temporal cube that associates a real value to each point in the cube. All operations defined for time-dependent types in Chapter 3 apply for

time-dependent fields. However, lifted operators must be renamed to differentiate those that operate on space or on time. For example, `sum_s` and `sum_t` correspond to the `sum` operator lifted in space and in time, respectively. Thus, given a set of time-dependent fields t_i representing the number of cars of type i that are present at a location in space at a particular instant, `sum_s({ti})` will result in a time-dependent field t obtained by applying the operator `sum_t` to each point in space, because each point in space defines a time-dependent real. Similarly, `sum_t({ti})` will result in a time-dependent field t obtained by applying the operator `sum_s` to each instant, since each instant defines a field of reals.

In addition, new spatio-temporal operators have to be defined. For example, operators `atMPoint`, `atMLine`, and `atMRegion` restrict the field to a given subset of the spatio-temporal cube defined by a time-dependent spatial value. In particular, projecting a time-dependent field to a time-dependent point with function `atMPoint` will keep only the points in the field that belong to the moving track of the point (i.e., a 3D line in the cube).

Consider the following query, which involves the field `LandUse`.

Query 4.6. “Give the average duration of the trajectories that started in a residential area and that ended in an industrial area on February 1, 2012.”

```
AVG(SELECT j.duration
      FROM Trajectory j, LandUse l
      WHERE date(j.startTime)=1/2/2012 AND date(j.endTime)=1/2/2012
            AND intersects(j.startLocation,defspace(at(l,'Residential'))),
            AND intersects(j.endLocation,defspace(at(l,'Industrial'))))
```

Here, function `at` projects the land use field to the values of type residential or industrial, function `defspace` obtains the geometry of the restricted field, and function `intersects` ensures that the start or end location is included in the obtained geometry. Because it is supposed that the attribute `startTime` is of type timestamp, function `date` is used for obtaining the corresponding date.

The next query involves the time-dependent field `Temperature`.

Query 4.7. “For episodes that occurred on February 1, 2010, give the average speed and the maximum temperature during the episode.”

```
SELECT e.number, e.avgSpeed, mmax(atMLine(l,e.route))
FROM Episode e, Time t, Temperature l
WHERE e.time=t.id AND t.date=1/2/2010
```

In the above query, function `atMLine` projects the time-dependent field to the movement track of the episode, resulting in a time-dependent real. Then, function `mmax` obtains the maximum temperature value during the episode.

The class of *spatio-temporal OLAP and continuous field (STOLAP-CF) queries* is the class that contains the queries expressed by Q_{agg} augmented with spatial types, time-dependent types, and field types. It follows that a *continuous field data warehouse* is a data warehouse that supports STOLAP-CF queries.

4.6 An Example Trajectory DW: GeoPKDD

We showed in previous sections that individual trajectories can be represented in facts and/or dimensions and that they can be aggregated and analyzed. An alternative way of analyzing trajectory data, as we commented in Section 4.3, consists in partitioning the space into regions (or road segments) and precomputing aggregated trajectory data relative to each partition. For example, we can partition the space into regular squares and for each square compute the number of trajectories at a given instant. This precomputation allows us to get rid of the trajectories, and analyze them using traditional DWs. One relevant example of this approach is the TDW developed in the GeoPKDD project.¹

The GeoPKDD TDW allows analyzing trajectory data without actually storing the trajectories themselves, but instead storing preaggregated measures resulting from a complex ETL process that feeds the TDW. During this ETL process, the sampled positions received by GPS-enabled devices are converted into trajectory data and stored in a moving object database, using the trajectory reconstruction techniques explained in Chapter 2. The moving object database also contains user profiles, spatial partitions, and temporal intervals.

After the reconstruction step, the TDW is fed with aggregate trajectory data using either a cell-oriented or a trajectory-oriented ETL approach. The *cell-oriented approach* searches for the trajectory portions that lie within the spatio-temporal cells. Then, those portions are decomposed with respect to the user profiles they belong to. On the other hand, the *trajectory-oriented approach* looks for the spatio-temporal cells where each trajectory resides. Then, portions of the trajectory that fit into each of those cells are computed, taking into account the user profiles.

In such a TDW, the dimensions are typically organized as follows. The *temporal* dimension is designed to range over equally sized time intervals, which can be aggregated according to larger intervals as we move up in the dimension hierarchy. The *spatial* dimension represents a partition of the space that defines the cells (or the road segments) where measures are recorded. Further, the fact

¹ <http://www.geopkdd.eu>

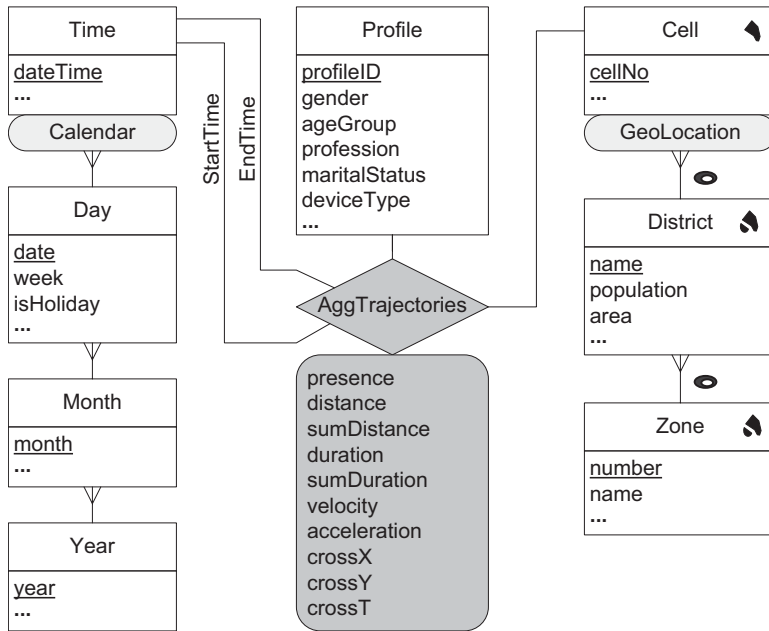


Figure 4.5 The GeoPKDD TDW in the MultiDim model.

table references the dimensions, and includes measures that provide indicators about the trajectories in each element of the partition (e.g., number of trajectories, total time spent in the cell or road segment, etc.). Finally, these aggregate data are exploited using a Visual OLAP interface that allows multidimensional and interactive analysis (covered in Chapters 7 and 8).

Figure 4.5 shows the conceptual schema of such a TDW using the MultiDim model. Dimension `Profile` collects demographic information (such as gender and age group) of the car drivers. In the spatial dimension, `Cell` represents the smallest unit we consider (i.e., a rectangle belonging to a grid that partitions the spatial domain). Further, a cell belongs to one district (this is obviously a simplifying approximation) and a district belongs to one zone. The `Time` dimension is analogous to the one in Figure 4.3, and the fact relationship `AggTrajectories` is related twice to this dimension, as illustrated by the roles `startTime` and `endTime`. Finally, each instance of the fact relationship contains aggregated measures about the trajectories of a given profile that cross a spatio-temporal cell. These measures are as follows:

- `presence`: the number of *distinct* trajectories.
- `distance`: the average distance of the trajectories.
- `sumDistance`: total distance covered by the trajectories.
- `duration`: the average duration of the trajectories.

- `sumDuration`: sum of the durations of the trajectories.
- `velocity`: average speed of the trajectories.
- `acceleration`: average change of speed of the trajectories.
- `crossX`, `crossY`, `crossT`: total number of *distinct* trajectories crossing the border between the cell and its adjacent cells, along the spatial (X and Y) and temporal (T) axes. These measures will be explained in Section 4.6.1.

We remark that these measures represent *aggregated* numeric information about trajectories. Thus, no spatio-temporal information about trajectories is recorded in the TDW whatsoever. This information lies only in the moving object database, and can be used for answering queries, along with the data in the TDW, when detailed (nonaggregated) information is required. Formally speaking, according to the definitions given in Section 4.4, the data warehouse in Figure 4.5 is a spatial data warehouse. Although useful in many practical situations, this approach does not suffice for a comprehensive analysis of movement data (see Section 4.8).

4.6.1 The Double-Counting Problem

As we have seen, the individual trajectories are not stored in the GeoPKDD TDW; only aggregate information is kept. As result, the *double-counting problem* may appear during aggregation over the partitioned space. We use the measure `presence`, explained above, to show the problem. Consider the three trajectories over the space divided into six regions R1 through R6 in Figure 4.6. If we perform a roll-up to aggregate the number of trajectories in regions R4, R5, and R6 (suppose they constitute a district), we would obtain a total of six trajectories (resulting from adding three trajectories in R4, two in R5, and one in R6), while the correct number to obtain would have been three trajectories. Solving this problem requires accessing the moving object database to compute super-aggregates in all dimension levels. This problem may occur while answering the following query.

Query 4.8. “Give the number of trajectories per district on January 1, 2010.”

In the above query, the measure `presence` must be aggregated over all the cells that belong to a district. A first solution would be to simply sum up the measure values of these cells. In the literature, this is a common, although very imprecise, approach to aggregating spatio-temporal data.

Another approach uses linear interpolation to prevent omitting in the result the cells crossed by a trajectory but in such a way that no sample point of the trajectory occurred within them. This approach borrows from statistical methods to deal with the double-counting problem. The basic idea is the following. Let us denote $\text{pres}_{C_{x,y,t}}$ the presence measure in a given cell $C_{x,y,t}$. Given

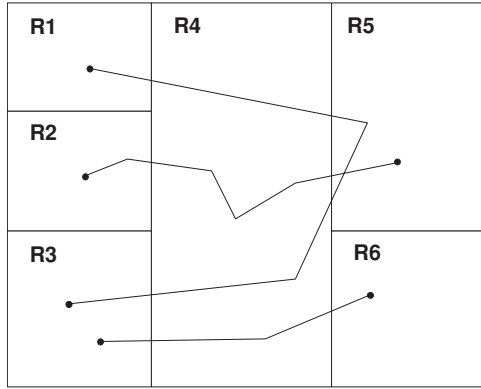


Figure 4.6 The double-counting problem.

a cell $C_{x,y,t}$, the measures `crossX` and `crossY` give the number of distinct trajectories crossing the spatial borders between $C_{x,y,t}$ and $C_{x+1,y,t}$ and $C_{x,y+1,t}$, respectively. Analogously, `crossT` gives the number of distinct trajectories crossing the temporal border between $C_{x,y,t}$ and $C_{x,y,t+1}$. Knowing the values of presence for two adjacent cells, $C_{x,y,t}$ and $C_{x+1,y,t}$, the aggregate value of `pres` over a new cell $C_{x',y',t} = C_{x,y,t} \cup C_{x+1,y,t}$ can be computed as follows:

$$\text{pres}_{C_{x',y',t}} = \text{pres}_{C_{x,y,t}} + \text{pres}_{C_{x+1,y,t}} - C_{x,y,t}.\text{crossX}$$

Similarly, the values $C_{x,y,t}.\text{crossY}$ and $C_{x,y,t}.\text{crossT}$ can be used to compute the presence in cells $C_{x,y,t} \cup C_{x,y+1,t}$ and $C_{x,y,t} \cup C_{x,y,t+1}$, respectively.

4.6.2 Querying the GeoPKDD TDW

We now use our query language Q_{agg} for querying the GeoPKDD TDW. As in Section 4.4, we assume a straightforward translation of the MultiDim schema in Figure 4.5 into a snowflake schema. Notice that, because the TDW does not contain moving object data, but only spatial data representing the partition of the space, only SOLAP queries can be addressed to the TDW.

For example, Query 4.8 above reads in Q_{agg} :

```
SELECT d.name, sumPres
FROM District d
WHERE sumPres= SUM( SELECT a.presence
  FROM AggTrajectories a, Cell c, Time t1, Time t2
  WHERE a.cell=c.id AND contains(d.geometry,c.geometry)
  AND a.startTime=t1.id AND a.endTime=t2.id
  AND intersects(range(t1.dateTime,t2.dateTime),1/1/2010) )
```

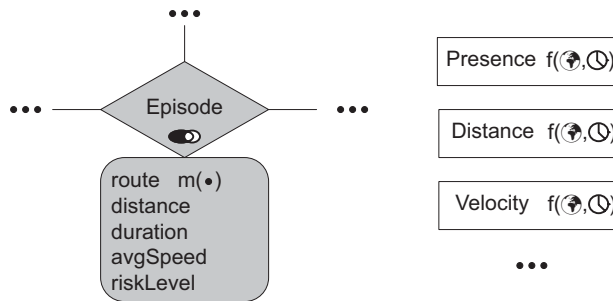


Figure 4.7 Combining the TDW in Figure 4.2 with the GeoPKDD TDW in Figure 4.5, the latter seen as set of time-dependent fields, one for each measure.

For each district we sum the `presence` measure for all cells contained in the district and such that the interval defined by start and end time of a trajectory intersects January 1, 2010.

4.6.3 Representing the GeoPKDD TDW as Continuous Fields

The reader may have noticed that *each measure* of the TDW in Figure 4.5 defines a collection of time-dependent fields, one for each user profile. These time-dependent fields are defined over spatio-temporal cells that have a fixed granularity of, say, one square kilometer and one hour. We can produce a time-dependent field for each measure by projecting out the `Profile` dimension and aggregating the measure with the functions studied in Section 4.5. For example, from the `presence` measure in Figure 4.5 we can produce a time-dependent field `Presence` by adding up the total presence by hour and square kilometer for all profiles using one of the functions `sum_s` or `sum_t`. We can proceed analogously with every measure in the cube, ending up with a collection of time-dependent fields with the same granularity as that of the original TDW. Notice that the functions defining these fields are stepwise ones, that is, the value of the measure is constant in each spatio-temporal cell.

As shown in Figure 4.7, using the approach above we can combine the TDW of our running example in Figure 4.2 with the GeoPKDD TDW in Figure 4.5. As an example, consider the following query, which combines the field `Presence` from the GeoPKDD TDW with our running example for discovering dense traffic areas in residential zones.

Query 4.9. “For districts with more than 70% of residential use, give the average presence of cars in January 21, 2012, at each point of the district.”

```
SELECT d.name, projPres
FROM District d, LandUse l, Presence p
```



```
WHERE projPres = favg(atperiod(atregion(p,d.geometry),21/1/2012))
AND (area(defspace(atregion(at(l,'Residential'),
d.geometry)))/area(d.geometry)) >= 0.7
```

In this query, the time-dependent field `Presence` is projected to the geometry of the district and to the date January 21, 2012, and the `favg` operator is applied to compute the average of presence across hours of the day. The resulting nontemporal field is kept in the variable `projPres`. On the other hand, the nontemporal field `LandUse` is projected to residential zones and to the geometry of the district, and the corresponding region is divided by the area of the district to verify the 70% condition specified in the query.

4.7 Conclusions

We have discussed data warehousing techniques that, in the presence of trajectory data, help to improve the decision-making process. For this, we defined the notion of trajectory data warehouses (TDW) as a particular case of spatio-temporal data warehouses, where trajectories can be represented both as measures and dimensions. By means of a running example we showed how a TDW can be modeled, designed, and queried, in order to deliver an aggregated view of trajectory data. In addition, as a particular case study, we discussed the GeoPKDD TDW, where facts contain aggregated trajectory measures instead of the trajectories themselves. Finally, we showed that representing the GeoPKDD TDW as a collection of continuous fields, one for each measure, provides additional possibilities for analysis.

4.8 Bibliographic Notes

Basic data warehousing concepts can be found in the classic book by Kimball (1996). This chapter is based on previous research work on spatio-temporal data warehousing and continuous fields performed by the authors (Vaisman and Zimányi, 2009a,b). Hierarchies in OLAP are studied, among other works, in Cabibbo and Torlone (1997). MultiDim, the conceptual model we use in this chapter, was introduced in Malinowski and Zimányi (2008). The query language we use throughout the chapter is based in the classic relational calculus with aggregate functions introduced by Klug (1982). The data type system follows the approach of Güting and Schneider (2005). The view of continuous fields as cubes was introduced in Gómez et al. (2012). The GeoPKDD TDW, its associated ETL process, and the double-counting problem during aggregation are studied in Orlando et al. (2007). A good discussion on TDW is presented

in [Pelekis et al. \(2008b\)](#) and in [Marketos et al. \(2008\)](#). Analysis tools for the TDW can be found in [Raffaetà et al. \(2011\)](#). [Andrienko and Andrienko \(2010\)](#) provide a state-of-the-art analysis on trajectory aggregation. They show that approaches like the one of the GeoPKDD TDW sometimes are not enough for a comprehensive trajectory analysis.

5

Mobility and Uncertainty

Claudio Silvestri and Alejandro A. Vaisman

5.1 Introduction

Mobility data are inherently uncertain due to several contributing factors related to different phases of their life cycle, from acquisition to interpretation. When data are processed, uncertainty propagates to intermediate and final results. Thus, it is important to be aware of uncertainty in trajectory data and explicitly account for it in their modeling and managing. For example, consider a simple scenario where people move around a city and disclose their positions twice an hour; to avoid stalking, the disclosed position is randomly selected from inside a circle with a radius of one kilometer, which contains the position of the user. Not being aware of uncertainty could lead to inconsistent conclusions. For instance, we could erroneously assume that a group of people have met or that someone has visited a privacy-sensitive place. On the contrary, taking uncertainty into account, we can avoid such erroneous conclusions; for example, if someone was farther than one kilometer from the place of an accident, we can certainly assume that this person was not involved in that accident.

We next introduce a well-known taxonomy of uncertainty (see Bibliographic Notes section), aimed at clearly defining terms that are often given multiple meanings in the literature.

A Taxonomy of Uncertainty

The taxonomy we present here considers, at the highest abstraction level, that uncertainty in mobility and geographic information is caused by the complexity of the system conformed by three kinds of entities: human being, earth (i.e., geographic/moving), and computing machinery. In simple terms, uncertainty reflects the variety of the geographic and movement reality, the computational capability of machines, and the limits of human cognition.

A first distinction classifies uncertainty as: (1) uncertainty of the *entities within* each one of the three domains above, and (2) uncertainty of the

relationships between entities in the three domains. For example, both the uncertainty due to the finite representation of coordinates and the one due to unknown positions fall into class (1), since they are caused respectively by the uncertainty in the computer representation and in the human cognition (lack of knowledge/memory) of entities.

The second branch (i.e., the uncertainty due to human, machine, and geography/movement relationships) can be refined according to the *kind* of difference existing between the corresponding entities in the different domains. In particular we can distinguish: *inaccuracy/error*, a deviation of a measurement from the reality; *incompleteness*, caused by a partial description of the reality; *inconsistency*, indicating the existence of different computational and cognitive statements referring to the same entity (e.g., because of semantic mismatch or contradiction, or simply due to different representations); and *imprecision*, which refers to a lack of exactness of computational or cognitive values. We can further classify imprecision depending on its degree in: *nonspecificity*, meaning that only a set containing the true value is known; *ambiguity*, when it is not possible to define univocally a set containing the exact value; and *vagueness*, when it is not possible to define a set containing the exact value, because true or false are just two of the possible truth values. We refer to *fuzziness* when the truth of a value is replaced by a continuously changing degree of truth. In both cases, no *sharp/crisp* boundary separates true and false values.

Uncertainty in Mobility Data

Using different position collection techniques entails different kinds of uncertainty affecting recorded data. Some of the tracking methods described in Chapter 2 have irrelevant errors on position and time measurement for most application scenarios, whereas other ones are intrinsically less precise. In other cases, the position is not measured, for example when it is manually inserted during a data entry process. In this case the position could be inaccurate, because of digitization errors, or vague, due to the nature of entities involved. For example, a valley is a vague concept and it is hard to devise crisp borders that have separate interior and exterior points. As a consequence, it is not possible to select the trajectories that stopped inside a valley in an exact way. Similarly, due to the lack of crisp borders of a zone frequently subject to avalanches, it is difficult to determine the number of skiers at risk even if we know exactly all of their trajectories.

Mobility data are characterized by several dimensions. In particular, in addition to space and time, data related to movement semantics and user actions could also be present. Each of these dimensions is potentially affected by one of the above kinds of uncertainty. For example, the semantic annotation and segmentation of trajectories could be affected by uncertainty in the spatial dimension. Thus, in case the geometry of a place of interest (POI) is fuzzy, or the positions of the objects are inaccurate, it could be difficult to assert that an object stopped at a POI.

In this chapter, after discussing the principal causes of uncertainty in mobility data, we address trajectory uncertainty and discuss two models for its representation: the cylinder and the space-time prisms model. We also address trajectory uncertainty for movement constrained to road networks. In this context, we show how the space-time prisms model can be used to address the map-matching problem introduced in Chapter 2. Finally, we also discuss how uncertainty can be accounted for in trajectory clustering.

5.2 Causes of Uncertainty in Mobility Data

Appropriate accounting for uncertainty requires being aware of its sources, both in data collection and data processing. This identification is crucial to decide if uncertainty should be accounted for in a given situation and how to manage it. Therefore, before moving on to the representation of uncertainty, we briefly discuss its main causes, distinguishing the uncertainty in the movement data per se from that introduced by postprocessing or deliberate accuracy/specificity reduction. Further, we analyze the observational error introduced by the main trajectory-tracking techniques suitable for mobility data.

Uncertainty in Localization

The uncertainty introduced when measuring moving object positions depends both on the technique adopted and on the context in which it is applied, as we detail later in this section. Regardless of the specific method used to track object positions, we can identify two kinds of sources of uncertainty: (1) those related to the nonspecificity of the acquired position, and (2) those related to the inaccuracy in the position measurement process. A presence sensor, for example, reveals the identity of objects that are within its range. Thus, by design, the spatial extent containing an object is known but the actual position of such object is unknown; therefore, the resulting position is affected by nonspecificity. On the other hand, the results of GPS position and time measures are precise, but affected by context-dependant stochastic errors, making them inaccurate. Note that for some position-tracking technology, both aspects may coexist. Consider a wireless communication equipment (GSM, WiFi, RFID, Bluetooth, etc.) used to detect when objects enter its range. In this case the position of the spotted object is a vague region, due to the possibly mutating environment. For example, some kind of obstacle may be on the line of sight of the receiving antenna, hindering the communications and thus potentially causing the object to be out of the range of the equipment.

Uncertainty Due to Intentional Accuracy Degradation

A measured position, by itself imprecise and inaccurate to some extent, can be further degraded either at collection time or later, before subsequent processing or disclosure. This apparently surprising choice is usually determined by privacy

or efficiency concerns. For example, Chapter 2 describes how the position of a mobile user may be obfuscated to protect his or her privacy either at the time the position data are acquired or before performing an operation that could disclose potentially harmful information to third parties. The same chapter presents methods for the compression of trajectories, which discard nonrepresentative positions in order to reduce the size of the digital representation of the trajectory. Also in this case the transformation yields a result that is less similar to the actual trajectory than the measured one. Finally, when representing a collection of trajectories, a further level of compression makes sense: grouping similar segments of trajectories and storing just a representative portion of each cluster instead of all of the original segments. Once again, this is a trade-off between accuracy and compact representation.

Uncertainty Due to Incomplete Data

Another source of uncertainty is the incompleteness of data. A typical example is the sampling of a trajectory: we know the position of an object at given time instants (both affected by observational errors). The positions occupied by the object between two samples can be obtained by means of interpolation techniques, making assumptions about the object's movement; for instance, using linear interpolation we are assuming that the object moves from one sample point to the next one at constant speed. Location inference is another possibility, which is based on the use of information about the object or about the context to restrict the possible object positions. For example, we may know that some action performed by the moving object was only possible at given positions, or that an object can only perform certain movements.

5.2.1 Localization Techniques and Uncertainty

The trajectory tracking methods presented in Chapter 2, as any other method of measurement, are affected by observational errors. These errors can directly affect the position and time measurement (when the measure is direct), or propagate to the computed position and time values (in the case of indirect measurement). We next discuss uncertainty in the localization techniques described in Chapter 2, and some other ones that the interested reader may find in the bibliographic notes section.

GPS

The computation of the GPS position is based on the computation of the distance of the receiver from a set of GPS satellites. This distance is measured indirectly, based on the different travel time of signals from the satellites to the receiver. Thus, an error in time measurement is propagated through the computation, and affects the accuracy of the resulting position. In practice, to obtain a

position, the distance from four known satellites is needed. The larger the number of satellites in sight, the higher the accuracy of the computed GPS position. The nominal accuracy of GPS position is 20 meters; however, by using more advanced techniques, it is possible to obtain higher accuracy: under one meter with ordinary differential GPS devices, and down to a few millimetres using specially equipped receivers to detect the phase differences between distinct satellite signals.

GSM

There are many ways of tracking GSM phones. The most basic one is to use the *call record data* containing the IDs of the starting and ending cells associated with the calls. In this case the uncertainty of the position is essentially due to the nonspecificity of the spatial information: depending on the density of the cellular network, the size of a cell could range from a hundred meters to some kilometers. More advanced alternatives are common to other wireless networks and are discussed later.

5.2.2 Generic Methods for Wireless Communication

RFID, Bluetooth, and WiFi, as well as GSM, are discussed together, based on the assumption that the mobile device *signal* can be identified by some fixed reference points, called *anchors*.

Range-Based Methods

This is the case of Bluetooth fixed receivers continuously querying nearby objects, RFID readers, and WiFi access points, but also of GSM cells, in case devices entering and exiting cells are logged. Once the *coverage* of an antenna is known, it is possible to restrict the position of the object to an area. If there is more than one anchor, it is possible to intersect the ranges to obtain a more accurate position. In this case the uncertainty is determined by both the size and the overlap of the antenna ranges. The denser the cells (as in metropolitan GSM areas), the more accurate is the determined position. Obstacles, at worst, may rule out some anchor even if it is in the proximity, thus losing an opportunity for position refinement.

Range-Free Methods

These methods rely on information about the *radio signal strength* (RSSI) received by the different anchors/antennas. The absolute value is not relevant, since they are conceived to work with the ratio between RSSIs. One of these methods, for example, is based on the computation of the centroid of the anchor points weighted with the respective RSSIs, which is invariant if the proportion of weights is preserved. The uncertainty is due to RSSI error propagation, and

to factors that affect signal in a nonlinear way, for example the presence of objects along the signal path, whereas omnidirectional signal attenuations are not relevant.

Distance and Direction-Based Methods

Unlike the methods described in the previous paragraph, distance-based methods use absolute RSSI to directly compute distance estimations. This method, however, requires calibration of the specific radio used, and it is particularly sensitive to any kind of perturbation. For this reason it is best suited for short-range distance measurement with known devices. A different approach, based on the use of a particular array of antennas that are able to compute RSSI for different directions, uses both the angle of arrival (AOA) and the RSSI to detect the position of the device. In both cases, assuming no obstacle, the uncertainty derives from the original observational errors involved in the computation of the indirect position measure, namely AOA and RSSI errors.

5.3 Uncertainty Models for Spatio-Temporal Data

We now turn to the problem of studying uncertainty of the trajectory of a moving object. A moving object's trajectory is obtained from raw trajectories, which are finite sequences of time-space points. The most-used technique for reconstructing trajectories from trajectory samples (see Chapter 2 in this book) is linear interpolation. However, it relies on the assumption that in between sample points, an object moves at constant minimal speed. It would be more realistic to assume that moving objects have some physically determined speed bounds. Given such upper bounds, *uncertainty models* have been proposed to estimate the possible positions between every two consecutive points in a trajectory sample. Note that uncertainty in trajectory databases may also arise from other sources, discussed earlier in this chapter, and also in Chapter 2. In this way, uncertainty not only refers to the possible locations of a moving object between two points in a sample, but also to these points themselves, which are not (in general) exactly recorded. Although the trajectory of a moving object has been traditionally modeled as a polyline in a 3D space (two dimensions for geography and one for time), modern approaches model such trajectory as a volume in 3D, either cylindrical or of a more complex kind. More precisely, in some of these approaches, the uncertainty of the moving object's position in between sample points is studied using *space-time prisms*. Informally, the space-time prism between two consecutive sample points is defined as the collection of space-time points where the moving objects may have passed, given a speed limitation. Geometrically, it is the intersection of two cones in the space-time space such that all possible trajectories of the moving object between the

two consecutive space-time points, given the speed bound, are located within them.

A rigorous analysis of a moving object’s trajectory requires that both the data model and the query language account for uncertainty. That means the language constructs must be aware that the data being queried are uncertain. Typical queries on moving object data ask for the objects inside a region sometime during a time interval, or for the ones always inside a region during a time interval. For example, taking into account the uncertainty of the objects’ position, one may query the objects that were possibly inside the region or the ones that were definitely there. For example, we may be interested in a query like: “Give me the current location of a bus that will possibly be at the corner of Avenue A and Avenue B at some time between 4:00 P.M. and 4:30 P.M.”

In the remainder of the section we study two models for considering uncertainty in trajectories. We also discuss uncertainty in road networks, and conclude the section studying how uncertainty is accounted for in trajectory clustering.

5.3.1 A Simple Model for Trajectory Uncertainty

Let \mathbb{R} denote the set of the real numbers, and \mathbb{R}^2 the 2D real plane. We consider objects moving in a subset of the 2D (x, y) space \mathbb{R}^2 and describe this movement in the (t, x, y) space $\mathbb{R} \times \mathbb{R}^2$, where t represents time. Moving objects (which hereon we assume to be points) produce, as we have already seen in this book, the kind of curves that we denote as *trajectories*. In practice, trajectories are only known at discrete moments in time, and given as sequences of the form $S = \{(x_0, y_0, t_0), (x_1, y_1, t_1), \dots, (x_N, y_N, t_N)\}$. Given a trajectory T between times t_1 and t_N , the expected location of the object at a point in time t between t_i and t_{i+1} ($1 \leq i < N$) could be obtained through linear interpolation between (x_i, y_i) and (x_{i+1}, y_{i+1}) .

Note that in its general form, a trajectory can represent both the past and future motion of objects. For future movement one can think of the trajectory as a set of points describing the motion plan of the object. The most common assumption is that we have a set of points that the object is going to visit, and that between the points the object is moving along the shortest path.

This simple model allows defining the notion of *uncertain trajectory*, obtained by associating an uncertainty threshold r with each line segment of the trajectory. For a given motion plan, the moving object associated with, for instance, a GPS device will update a server if and only if it deviates from its expected location (according to the trajectory) by r or more. In practice, a GPS update is sent at certain predefined intervals; therefore, the location of the object is known, and by linear interpolation, the object’s expected location can be computed at

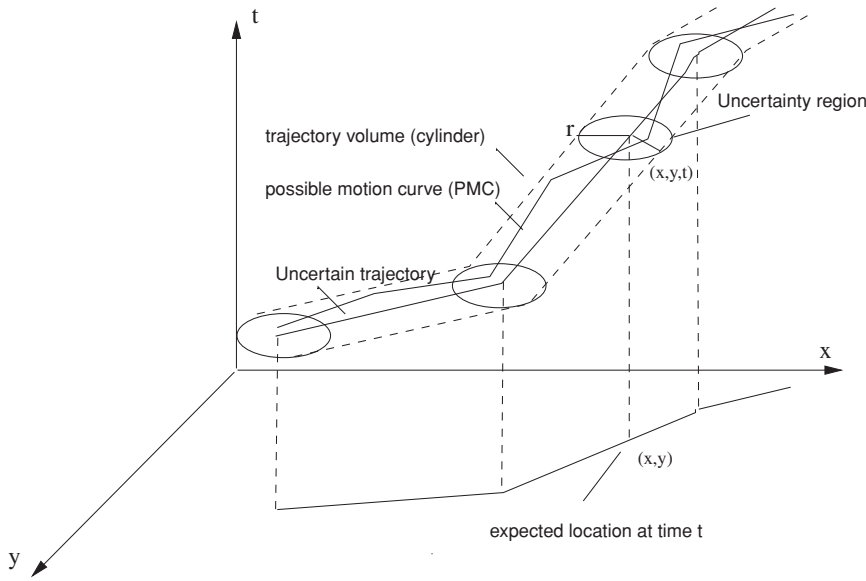


Figure 5.1 Uncertain trajectories.

any point in time. The deviation is just the distance between the actual and the expected location. Formally:

Definition 5.1. Let r denote a positive real number and T denote a trajectory between times t_1 and t_n . An *uncertain trajectory* UTr is the pair (T, r) , where r is called the uncertainty threshold. For each point (x, y, t) in T , its uncertainty area is a horizontal disk (i.e., the circle and its interior) with radius r centered at (x, y, t) , where (x, y) is the expected location at time $t \in [t_1, t_n]$. □

Figure 5.1 graphically depicts this definition.

Definition 5.2. Let $UTr = (T, r)$ be an uncertain trajectory between instants t_1 and t_n . A *possible motion curve* $PMC(T)$ of T is any continuous function f_{pt} with signature $Time \rightarrow R^2$ defined in the interval $[t_1, t_n]$ such that for any $t \in [t_1, t_n]$, the 3D point $(f_{pt}(t), t)$ is inside the uncertainty area of the expected location at time t . □

Intuitively, a PMC describes a possible route (and its associated times) that a moving object may take without generating an update. In other words, in a practical situation, a moving object does not need to update the database as long as it is on some possible motion curve of its uncertain trajectory. The projection over the plane of a possible motion curve is called a *possible route*.

Querying the Model

According to the model, we can classify operators for querying moving objects with uncertainty in two classes: (a) operators for *point queries*; (b) operators for querying the relative position of a moving object with respect to a region, within a given time interval. Each one of these operators corresponds a *spatio-temporal range query*.

Operators for Point Queries

Two operators for point queries are defined in the literature:

- `Where At(T, t)`: returns the *expected* location on the route of trajectory T at time t .
- `When At(T, l)`: returns the times at which the moving object whose trajectory is T is *expected* to be at location l . (Note that in this case the answer may be a set of time instants, if the moving object passes through a point more than once).

If the location $l = (x_l, y_l)$ is not on the route of T , the `WhenAt(T, l)` operator finds the set of all the points C on this route that are closest to l , and returns the set of time instants at which the object is expected to reach each point in C .

Operators for Spatio-Temporal Range Queries

These operators comprise a set of Boolean predicates such that each predicate is satisfied if the moving object is inside a given region R during a given time interval $[t_s, t_e]$. Queries may ask if the condition is satisfied sometime or always within $[t_s, t_e]$ (due to the motion of the object), and/or if, due to the uncertainty, the object *possibly* or *definitely* satisfies the condition at some time within the interval. The main operators corresponding to spatio-temporal range queries are:

- `Possibly Sometime Inside(T, R, t_s, t_e)`. The predicate is true iff there exists a $PMC(T)$ for the trajectory T and a time $t \in [t_s, t_e]$ such that $PMC(T)$ at time t is inside the region R .
- `Possibly Always Inside(T, R, t_s, t_e)`. The predicate is true iff there exists a $PMC(T)$ that is inside the region R for every $t \in [t_s, t_e]$.
- `Always Possibly Inside(T, R, t_s, t_e)`. True iff for every time value $t \in [t_s, t_e]$ there exists some (not necessarily unique) $PMC(T)$ that is inside (or on the boundary of) R at t .
- `Always Definitely Inside(T, R, t_s, t_e)`. This is true iff at every time $t \in [t_s, t_e]$, every possible motion curve $PMC(T)$ is in region R .
- `Definitely Sometime Inside(T, R, t_b, t_e)`. This is true iff for every possible motion curve $PMC(T)$ of the trajectory T , there exists some time $t \in [t_b, t_e]$ in which the particular motion curve is inside R .

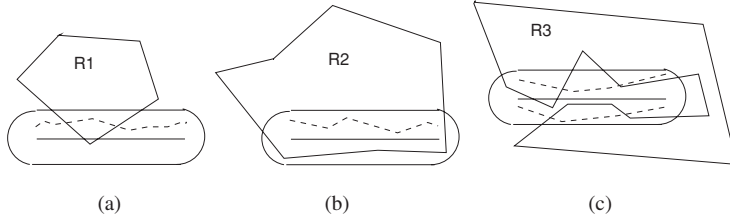


Figure 5.2 Uncertain query operators: (a) Possibly Sometime Inside R1; (b) Possibly Always Inside R2; (c) Always Possibly Sometime Inside R3 (c). Dashed lines indicate PMCs that satisfy the predicates. Solid lines represent the routes, and solid ellipses the uncertainty zones.

Figure 5.2 illustrates the semantics of the first three operators.

5.3.2 The Space-Time Prism Model

We now discuss the more general *space-time prism* model for uncertainty management, and describe its possible application to different problems. This model assumes that besides the time-stamped locations of the object, also some background knowledge, in particular a (e.g., physically or law-imposed) speed limitation v_i at location (x_i, y_i) is known. The speed limits that hold between two consecutive sample points can be used to model the uncertainty of a moving object's location between sample points. The approach of Section 5.3.1 (sometimes called the cylinder approach) depends on an uncertainty threshold value $r > 0$ which produces a sort of buffer along the trajectory. Instead, in the space-time prism approach, for each consecutive pair of points $(t_i, x_i, y_i), (t_{i+1}, x_{i+1}, y_{i+1})$ in a trajectory T , their related space-time prism does not depend on an uncertainty threshold value, but rather on a maximal velocity value v_{\max} of the moving object.

Intuitively, the space-time prism between two consecutive points is defined as the set of time-space points where the moving objects may have passed, respecting the speed limitation. The chain of space-time prisms connecting consecutive trajectory points is denoted the *lifeline necklace* (see Figure 5.3).

We now formalize the concepts above. We know that at a time $t, t_i \leq t \leq t_{i+1}$, the object's distance to a point (x_i, y_i) is at most $v_i(t - t_i)$ and its distance to (x_{i+1}, y_{i+1}) is at most $v_i(t_{i+1} - t)$. The spatial location of the object is therefore somewhere in the intersection of the disc with center (x_i, y_i) and radius $v_i(t - t_i)$ and the disc with center (x_{i+1}, y_{i+1}) and radius $v_i(t_{i+1} - t)$. The geometric location of these points is referred to as a *space-time prism*, and defined as follows, for general points $p = (t_p, x_p, y_p)$ and $q = (t_q, x_q, y_q)$, and speed limit v_{\max} .

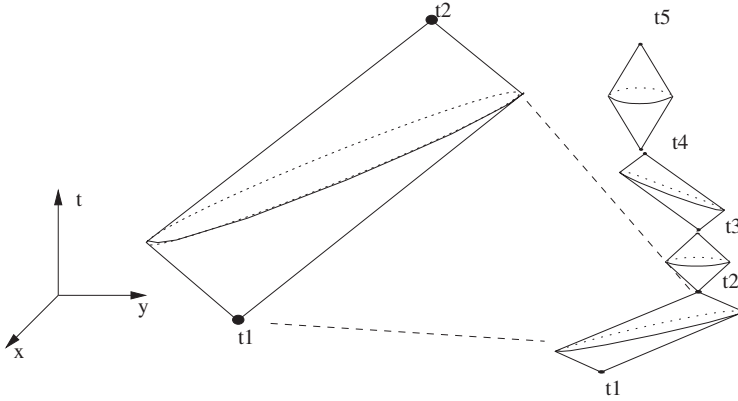


Figure 5.3 Space-time prisms and lifeline necklaces.

Definition 5.3. The *space-time prism* with origin $p = (t_p, x_p, y_p)$, destination $q = (t_q, x_q, y_q)$, with $t_p \leq t_q$, and maximal speed $v_{\max} \geq 0$ is the set of all points $(t, x, y) \in \mathbb{R} \times \mathbb{R}^2$ that satisfy the following constraint formula.

$$\Psi_B(t, x, y, t_p, x_p, y_p, t_q, x_q, y_q, v_{\max}) := (x - x_p)^2 + (y - y_p)^2 \leq (t - t_p)^2 v_{\max}^2 \wedge (x - x_q)^2 + (y - y_q)^2 \leq (t_q - t)^2 v_{\max}^2 \wedge t_p \leq t \leq t_q. \quad \square$$

In the formula $\Psi_B(t, x, y, t_p, x_p, y_p, t_q, x_q, y_q, v_{\max})$, t, x, y are variables defining the subset of $\mathbb{R} \times \mathbb{R}^2$, while all the other terms are parameters.

5.3.3 Uncertainty in Road Networks

So far we have not made any assumption about where the trajectories under study develop. These trajectories are usually called *unconstrained*. However, in general, trajectories develop within a road network in \mathbb{R}^2 . In this case, they are denoted *constrained* trajectories. This constrained movement has its own peculiarities. Before studying them, we first need to formalize the notion of a *road network*.

Definition 5.4. A *road network* RN is a graph embedding in \mathbb{R}^2 a labeled graph given by a finite set of vertices $V = \{(x_i, y_i) \in \mathbb{R}^2 \mid i = 1, \dots, N\}$, and a set of edges $E \subseteq V \times V$ that are labeled by a *speed limit* and an associated *time span*. This graph embedding satisfies the following conditions. Edges are embedded as straight line segments between vertices, and may intersect in nonvertex points, to support modeling bridges and tunnels. If an edge is labeled by the speed limit, then its time span is the time needed to get from one side of an edge to another when traveling at the speed limit. \square

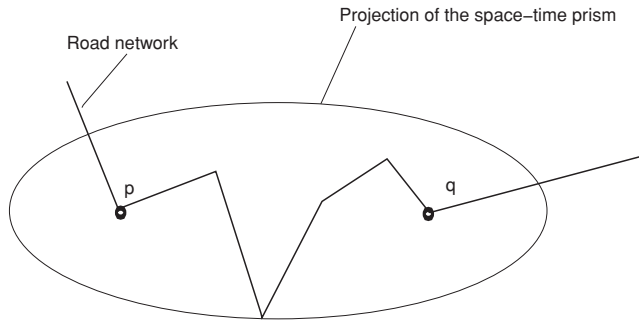


Figure 5.4 A projection of a prism and a road network.

A trajectory on a road network RN is then a trajectory whose spatial projection is in RN . In the remainder we consider a uniform speed limit v_i on the network to construct the space-time prism between two sample times t_i and t_{i+1} .

Space-Time Prisms in Road Networks

Using space-time prisms on a road network is usually more involved than simply taking the intersection of a space-time prism representing unconstrained movement and the road network. Consider, for instance, the projection of the unconstrained space-time prism along the time axis onto the xy -plane. This projection is an ellipse such that its foci are the points of departure and arrival, that is, p and q . At a time t between two instants t_p and t_q , the object's distance to p is at most $v_{\max}(t - t_p)$ and its distance to q is at most $v_{\max}(t_q - t)$. Adding those distances gives $v_{\max}(t - t_p) + v_{\max}(t_q - t) = v_{\max}(t_q - t_p)$, which is constant. Therefore, all possible points a moving object with speed limit v_{\max} could have visited must lie within this ellipse with foci p and q , and the sum of their distances to p and q is less than or equal to $v_{\max}(t_q - t_p)$. Any trajectory that touches the border of the ellipse and has more than two straight line segments is longer than $v_{\max}(t_q - t_p)$ (see Figure 5.4). This particular trajectory lies in the ellipse and hence in the intersection of the unconstrained space-time prism and the road network, *but it does not lie in the road network space-time prism entirely*, because there are points on it that can be reached in time but from which the destination cannot be reached in time, and vice versa. Just suppose a case where there is no path on the road network from a vertex p that reaches another one q in a given time interval. The intersection of the space-time prism with the road network would not be empty. However, the road network space-time prism clearly is, because there is no way to reach q from p using the network.

To define space-time prisms on a road network, we need an appropriate distance function on the network. This distance measure is derived from the *shortest-path distance* used in graph theory.

Definition 5.5. Consider a road network RN , given by the tuple (V, E) and to points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ on RN , not necessarily vertices; the point p lies on the embedding of the edge $((x_{p,0}, y_{p,0}), (x_{p,1}, y_{p,1}))$ and q lies on the embedding of the edge $((x_{q,0}, y_{q,0}), (x_{q,1}, y_{q,1}))$. We construct a new road network RN_{pq} from RN , such that $V_{pq} = V \cup \{p, q\}$, and $E_{pq} = E \cup \{((x_{p,0}, y_{p,0}), (x_p, y_p)), ((x_p, y_p), (x_{p,1}, y_{p,1})), ((x_{q,0}, y_{q,0}), (x_q, y_q)), ((x_q, y_q), (x_{q,1}, y_{q,1}))\}$. \square

Definition 5.5 builds a new network by splitting the edges on which p and q are located. The speed limits are the ones of the original edges, and the time spans of the new edges are computed according to Definition 5.4. Based on this construction, we define the distance along the road network RN and the space-time prism between p and q on RN .

Definition 5.6. Let RN be a road network and let $p, q \in \text{RN}$. The road network time between p and q , denoted by $d_{\text{RN}}(p, q)$, is the shortest-path distance (i.e., as usual in graph theory) between p and q in the graph (V_{pq}, E_{pq}) , with respect to the time span labeling of the edges. \square

Note that the road network time between p and q in Definition 5.6 returns the earliest possible time from q to p and vice versa. The metric takes two points from a road network and returns the shortest time needed to get from one to the other when traveling at the allowed maximal speed at each segment. If there are different speed limits per edge, then the metric of Definition 5.6 is the shortest time span metric on the temporal projection of the spatio-temporal data. In this case the shortest paths are not always the fastest paths. Conversely, if all edges in road network have the same speed limit, then the metric results in the shortest path on the graph embedding.

We are ready to define a space-time prism on a road network. We provide a simplified definition, and omit the most technical details.

Definition 5.7. A space-time prism on a road network between two spatio-temporal points (x_p, y_p, t_p) and (x_q, y_q, t_q) , is the geometric location in $\mathbb{R} \times \text{RN} \subset \mathbb{R} \times \mathbb{R}^2$ of all points a moving object could have visited when traveling, restricted to RN , from an origin p to a destination q within a time frame ranging from t_p to t_q , respecting the speed limits on the edges of RN . That means that given a point $u = (x, y) \in \text{RN}$, $d_{\text{RN}}(p, u) + d_{\text{RN}}(u, q) \leq (t_q - t_p)$. \square

Figure 5.5 shows an example of a space-time prism.

5.3.4 An Application: Using Space-Time Prisms for Map Matching

Chapter 2 studied a typical problem that presents in network-constrained trajectories: map matching. Informally, this problem consists in mapping a trajectory

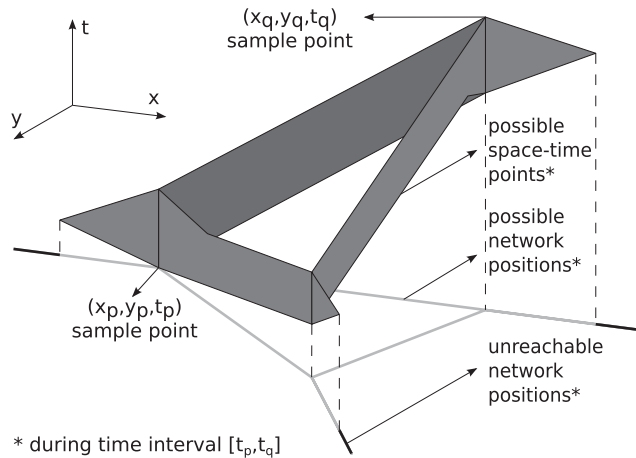


Figure 5.5 A space-time prism on a road network. Note that the possible positions of moving objects (represented by the cones in Figure 5.3 for unconstrained movement) occur over the network, and that edges have potentially different speed limits.

to the edges and nodes of the network. In that chapter, map matching algorithms were classified as geometric, topological, hybrid, and probabilistic. In this section we show how the space-time prisms model can be applied to solve this problem. This method was applied to a real-world case study involving an emergency service in a European city (for privacy reasons we cannot disclose further information). This service wanted to optimize the time to arrive at the place of intervention. Even though the company could solve this problem purchasing a standard route planner, the shortest/fastest route computed by these commercial route planners would not be the best solution, because, for instance: (1) they do not take into account the time of the observations (e.g., at five o'clock there is always a traffic jam at the city station, so cars must avoid this area around that time, if possible); (2) they do not take into account certain locations, such as schools; (3) they do not take into account additional information (such as school routes or tram lines). Thus, they decided to design a tool to solve the problem described above. As a first step of this work, there was the need to perform data analysis over a set of routes followed by cars during their interventions. The officers were asked to record their positions using a GPS device, from the moment they got a call from the headquarters to the moment when they arrived at the intervention site. Measures were recorded every ten meters, and drivers were requested to fill out a survey with questions, for example, about the reason for taking a particular route. In this scenario, a typical problem that arises is that about ninety-five percent of the points fall outside the road actually taken. Thus, there is a need to map points to the road network, that is, a map matching problem. This problem is formalized in Definition 5.8.

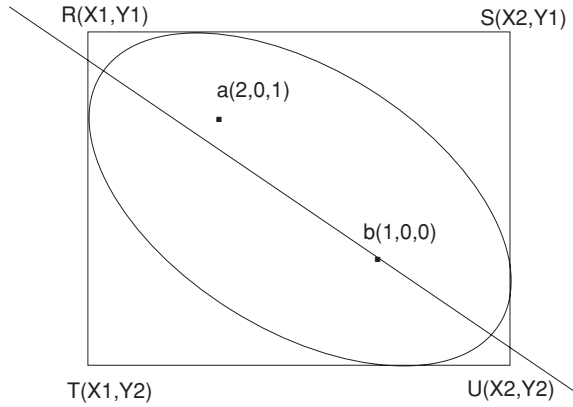


Figure 5.6 Projection of a space-time prism for two points, a and b .

Definition 5.8. An object is moving along a finite system (or set) of streets, \bar{N} . A location-aware device provides an estimate for the vehicle's location at a finite number of points in time, denoted by $\{0, 1, \dots, t\}$. The vehicle's actual location at time t is denoted by \bar{P}^t and the estimate is denoted P^t . *Map matching* is the process of determining the street in \bar{N} that contains P^t . That is, to determine the street that the vehicle is on at time t . \square

For applying the space-time prisms method to the map-matching problem, we must first make the following considerations. Given the time between two consecutive recorded points, a and b , and a maximal speed, a car could have been in many possible locations, determined by the projection of the space-time prisms over the plane. Even though this projection would be an ellipse, we can simplify this computation defining a bounding box given by two points, $R(X_1, Y_1)$ and $U(X_2, Y_2)$, as Figure 5.6 shows (the line within the ellipse represents the actual road). R is computed as follows: X_1 is the farthest point on the x -axis that can be reached moving away from b driving at maximum speed v_{max} . Analogously, Y_1 is the farthest point on the y -axis that can be reached moving away from b driving at maximum speed v_{max} . U is computed as follows: X_2 is the farthest point on the x -axis that can be reached moving away from a driving at maximum speed v_{max} . Analogously, Y_2 is the farthest point on the y -axis that can be reached moving away from a driving at maximum speed v_{max} .

We next sketch an algorithm for map matching based on space-time prisms (in the following, ST-MM).

1. First, bound the network by calculating, for each pair of consecutive points, the roads that connect them (as described above).
2. For each GPS point, compute the n closest road segments, assigning weights to each segment in a way such that the one closest to the point gets weight n ,

the second closest receives weight $n - 1$. The closest n road segment receives weight 1. Notice then that *the road segments to be included are computed using space-time prisms*. A score for a segment s is computed adding up the weights of all the segments that match s .

3. Finally, compute, within this limited network, the k -shortest paths, taking the shortest path with the highest score computed in Step 2.

Chapter 2 studied geometric map-matching algorithms. These algorithms are efficient due to their simplicity. On the other hand, geometric algorithms have some drawbacks that sometimes prevent trajectory reconstruction. This is the case, for instance, when observations are taken at irregular intervals, or there are large gaps in the data (this may occur, for example, when an object enters a large tunnel, preventing signal reception). In these cases, we need more sophisticated algorithms, such as the one described above. Summarizing experiments performed on real-world data showed the following:

- ST-MM is sensitive to the maximum speed. For relatively high speeds (70–120 km/h) it is stable and delivers good performance, with an average of around eighty percent trajectory reconstruction rate. When speeds are lower, performance decreases as well as the reconstruction rate.
- Geometric algorithms perform well when data are recorded at regular intervals and there are not large gaps between observations (note that these algorithms are independent of the maximum speed).
- When data are irregular and contain large gaps, ST-MM delivers better reconstruction rates, except when maximum speeds are low. In the latter case, geometric algorithms are more efficient, although reconstruction rates remain low for both algorithms.
- The scenario where ST-MM is clearly better than simple geometric algorithms is the one in which speeds are relatively high and measures are taken at irregular intervals. On the contrary, where speeds are low, geometric algorithms perform better because the prisms in ST-MM include a high number of roads, decreasing performance.

5.3.5 Trajectory Clustering and Uncertainty

Clustering is a data mining technique that partitions a data set into collections of data objects, such that within each partition the objects are “similar” to each other and “different” from the objects contained in other partitions. In the context of moving object data, the clustering technique aims at identifying groups of objects that follow similar trajectories. Clustering is tackled in detail in Chapter 6 of this book. In this section we show how the presence of uncertainty impacts on clustering results.

Many clustering techniques (discussed in Chapter 6), such as the popular k -means, can be applied to the trajectory setting using a so-called *distance function* between trajectories, which measures the similarity between trajectories. This leads to the notion of *distance-based clustering*. Clustering trajectory data usually produces groups containing geographically close trajectories. Many different distance functions can be defined, ranging from the most simple ones (for instance, clustering trajectories with the same origin and/or destination), to very complex mathematical functions.

The space-time prism approach allows defining a distance function for trajectories that accounts for uncertainty. Let us consider two trajectory samples $T1$ and $T2$, such that their uncertainty is represented by two lifeline necklaces, N_1 and N_2 , respectively, that connect consecutive sample points of each trajectory. Intuitively, the larger the intersection of the necklaces with respect to their union, the smaller the distance between both trajectories. In other words, the more uncertainty shared by $T1$ and $T2$, the closer they are. On the other hand, if N_1 and N_2 do not intersect, this indicates that these trajectories could not have met, given the speed limit. Then, a clustering algorithm should not group together these two trajectories. We can conjecture that the temporal projection of the intersection of the space-time prisms of two trajectories represents the instants when the two trajectories *could have met*. Therefore, the longer this period, the more similar the trajectories are. This notion is captured by Definition 5.9.

Definition 5.9. Let us denote A and B two necklaces corresponding to two trajectory samples τ_1 and τ_2 , respectively; also, we denote V_C the volume of a 3-dimensional figure C . Then, the expression

$$d_u(A, B) = 1 - \frac{V_{A \cap B}}{V_{A \cup B}}$$

is named the *uncertainty-based distance* between τ_1 and τ_2 . □

It can be proved that $d_u(A, B)$ is a distance metric, that is, it verifies identity ($\forall i : d(i, j) = 0$ iff $i = j$), positive definiteness ($\forall i, j, i \neq j : d(i, j) > 0$), symmetry ($d(i, j) = d(j, i)$), and triangle inequality ($\forall i, j, k : d(i, j) + d(j, k) \geq d(i, k)$).

The most difficult part of applying this uncertainty-aware distance function consists in the computation of the intersection between two chains of space-time prisms for any given two trajectories whose distance we need to calculate. To make this computation more efficient, information related to the road network can be preprocessed. The reader is encouraged to check the bibliographic notes of this chapter, where references to works describing this computation are given.

Approaches based on fuzzy regions have also been proposed in the recent literature. In these approaches, trajectory databases are considered as fuzzy sets that represent the regions that a trajectory possibly crosses, and fuzzy values represent the probabilities of presence and nonpresence of the moving objects in the area. Based on this model, an uncertainty-aware distance metric is defined and used in a clustering algorithm. In the bibliographic notes we provide the reference to this work.

5.4 Conclusions

Several kinds of mobility-related data are to some extent uncertain; explicitly representing and managing uncertainty ensures that data are handled in a sensible way. In this chapter we first analyzed several causes of uncertainty in data collection and management, and the accuracy of several location-tracking methods. We then described two well-known models for trajectory uncertainty. If movement is constrained to road networks (as it is in most real-world scenarios), uncertainty modeling becomes more involved. Thus, we studied uncertainty in road networks and also presented an approach based on the space-time prism model to address the typical problem of map matching. Finally, we showed how uncertainty can be accounted for in trajectory clustering.

5.5 Bibliographic Notes

A complete study of positioning techniques for wireless sensor networks and their uncertainty, described in Section 5.2.1, is presented in [Dricot et al. \(2009\)](#). The taxonomy of uncertainty adopted in this chapter was introduced in [Shu et al. \(2003\)](#). In [Shu et al. \(2003\)](#) and [Pauly and Schneider \(2010\)](#), the interested reader can find useful additional material on modeling uncertainty in temporal and spatial data. The definitions of trajectory and trajectory sample that we present in this chapter are based on [Kuijpers and Othman \(2009\)](#). The notion of geospatial lifeline was introduced in [Egenhofer \(2003\)](#). The notions of uncertain trajectory and possible motion curve, and how to query this uncertainty model, were taken from [Trajcevski et al. \(2004\)](#). Figures 5.1 and 5.2 are also based on that article. A detailed mathematical analysis on uncertainty in trajectories can also be found in [Trajcevski \(2011\)](#). [Pfoser and Jensen \(1999\)](#) also studied space-time prisms. However, space-time prisms were already known in the time-geography of [Hägerstrand \(1970\)](#). Early adaptations of the space-time prism model to road networks were done by [Miller \(1991\)](#). Also, [Kuijpers and Othman \(2009\)](#) studied the problem of space-time prisms on road networks, and introduced an algorithm for computing and visualizing space-time prisms. The use of space-time prisms for defining a distance function for clustering was introduced in [Kuijpers et al. \(2009\)](#). The computation of the distance function

requires calculating the intersection between two space-time prisms. [Kuijpers and Othman \(2009\)](#) present an algorithm that computes this intersection as an intersection between polygons. Also, Figure 5.5 is taken from [Othman \(2009\)](#). Finally, trajectory clustering based on fuzzy sets is studied in [Pelekis et al. \(2011\)](#).

PART II

**MOBILITY DATA
UNDERSTANDING**

6

Mobility Data Mining

Mirco Nanni

6.1 Introduction

6.1.1 What Is Mobility Data Mining?

The trajectories of a moving object are a powerful summary of its activity related to mobility. As seen in Chapters 3 and 4, such information can be queried in order to retrieve those trajectories (and the objects that own them) that respond to some given search criteria, for instance following a predefined interesting behavior. However, when massive amounts of information are available, we might be able to move a step further and ask that such “interesting behaviors” automatically emerge from the data. That is precisely the domain explored by mobility data mining.

Moving from queries to data mining essentially consists of adding degrees of freedom to the search process that the algorithms perform. For instance, a query might consist of searching those trajectories that at some point perform the following sequence of maneuvers: abrupt slow down, U-turn, and, finally, accelerate. One possible corresponding data mining task, instead, might require one to discover which sequences of maneuvers are performed frequently in the database of trajectories. Then, the output sequences obtained might also contain the *slow down* → *U-turn* → *accelerate* example just mentioned. To perform this data mining process the user needs to specify the general structure of the behaviors he or she searches (sequences), what kind of elements they can contain (the set of maneuvers to consider, as well as a precise way to locate a given maneuver within a trajectory), and a criterion to select “interesting” behaviors – in our example, the user wants only behaviors that appear frequently in the data.

6.1.2 Note on Terminology

In this chapter we will make frequent use of the term “trajectory pattern.” As mentioned in Chapter 1, the notion of trajectory pattern is substantially equivalent to that of “trajectory behavior,” which also appeared in previous chapters of this book. The two notions originate from different communities and simply reflect different perspectives of the same subject: the data management view (where “trajectory behavior” originates) focuses more on determining *which trajectory* is associated to each behavior; the data mining view, on the contrary, is more focused on *what* are the interesting behaviors in the input trajectories.

The several forms and variants of existing analysis tasks that belong to mobility data mining cannot be easily categorized into a set of fixed classes. However, it is possible to recognize a few simple dimensions along which to locate the different analysis methods. In the following we mention one of them, which will also be used later as guideline during the presentation of analysis examples.

6.1.3 Local Patterns versus Global Models

The example of behavior illustrated at the beginning of this section is representative of a class of mining methods, called *local patterns* or, in most contexts, simply *patterns*. The key point of local patterns is the aim of identifying behaviors and regularities that involve only a (potentially small) subset of trajectories, and that describe only a (potentially small) part of each trajectory involved.

The complementary class of mining methods is called *global models*, or simply *models*. Their objective is to provide a general characterization of the whole data set of trajectories, thus going toward the definition of general laws that regulate the data, rather than spotting interesting yet isolated phenomena. For instance, we will see later mining tasks aimed to define a global subdivision of all trajectories into homogeneous groups, as well as tasks aimed to discover rules able to predict the future evolution of a trajectory (i.e., the next locations it will visit).

In the rest of the chapter we will provide an overview of the problems and methods available in the mobility data mining field. For obvious reasons of space, the discussion will not cover exhaustively the available literature on the subject, and instead will propose some representative examples of the various topics. The presentation will mainly follow the distinction between local patterns and global models already introduced. In this chapter we will assume that raw location information, such as GPS traces, has already been preprocessed to obtain trajectories according to the discussions provided in Chapter 2, and will not consider the additional issues related to uncertainty already tackled in Chapter 5. Besides the examples provided here, the reader can find some

applications of the trajectory data mining methods we describe here in the next chapters, especially Chapters 7, 9, and 10.

6.2 Local Trajectory Patterns/Behaviors

The mobility data mining literature offers several examples of trajectory patterns that can be discovered from trajectory data. Among this wide variety, a very large number of proposals actually adopt two basic assumptions: first, a pattern is interesting (and therefore extracted) only if it is frequent, and therefore it involves (or appears in) several trajectories¹; second, a pattern must describe (also) the movement in space of the objects involved, and not only aspatial or highly abstracted spatial features. In this chapter we will adopt such assumptions, in order to better focus the discussion.

While the spatial component of trajectory data is typically part of the patterns extracted, the temporal one (also intrinsic in trajectory data) can be treated in several different ways, and we will use this differentiation to better organize the presentation. Then, while a trajectory pattern always describes a behavior that is followed by several moving objects, we can choose whether they should do so together (i.e., during the period), at different moments yet with the same timing (i.e., there can be a time shift between the moving objects), or in any way, with no constraints on time.

6.2.1 Using Absolute Time or Groups That Move Together

One of the basic questions that arise when analyzing moving objects trajectories is the following:

Are there groups of objects that move together for some time?

For instance, in the realm of animal monitoring such kind of patterns would help to identify possible aggregations, such as herds or simple families, as well as predator–prey relations. In human mobility, similar patterns might indicate groups of people moving together on purpose or forced by external factors, for example, a traffic jam, where cars are forced to stay close to each other for a long time period.

Obviously, the larger the groups and/or the longer the period they stay together, the higher the likelihood that the observed phenomenon is not a pure coincidence. For instance, if two members of a population of zebras under monitoring happen to move close to each other for a short time, that can be seen as a random encounter. However, if dozens of zebras are observed together for

¹ Of course, significant exceptions exist, including the extreme case of outlier detection, consisting of anomalous (and thus infrequent) patterns. For ease of presentation, outlier detection will be described later in this chapter, in the context of *global models*.



Figure 6.1 Visual representation of (a) a trajectory flock, (b) a sample result on a real data set with all trajectories involved, and (c) a zoom on the segments that form the flock. (See color plate.)

several hours, we can safely assume that they form a herd or that something is happening that forces them to keep together.

The simplest form of trajectory pattern in literature that exactly answers the question posed above is the *trajectory flock*. In one of its most common variants, a flock is defined as a group of moving objects that satisfy three constraints as follows:

- A spatial proximity constraint: Within the whole duration of the flock, all its members must be located within a disk of radius r – possibly a different one at each time instant, that is, the disk moves to follow the flock;
- A minimum duration constraint: The flock duration must be at least k time units;
- A frequency constraint: The flock must contain at least m members.

Figure 6.1a shows an abstract example of flock, where three trajectories meet at some point (at the fifth time unit), keep close to each other for some time (four consecutive time units) and then separate (ninth time unit). If, for instance, the constraints chosen by the user are the radius r used in the figure to draw the circles, a minimum duration of four time units (or less), and a minimum size of three members, then the common movement shown in the figure will be recognized as a flock.

Figures 6.1b–c show an example extracted from a real data set that contains GPS tracks of tourists in a recreational park (Dwingelderveld National Park, in the Netherlands). Figure 6.1b depicts the three trajectories that were involved in the flock, while Figure 6.1c shows (a zoom with) only the segments of trajectories that create the flock. As we can see, in this example a flock is a local pattern, both in the sense of involving only a small subset of trajectories (three, in our case), and in the sense of describing an interesting yet relatively small segment of the whole life of the trajectories involved.

The general concepts of *moving together* or *forming a group* are implemented by the flocks framework in the simplest way possible: the objects are required

to be very close to each other during the duration of the flock. However, a group might appear also under different conditions. One of these alternatives is to require that at each timestamp the objects form a *cluster* – thus borrowing ideas and methods from the clustering literature. Notable examples are *moving clusters* and *convoys*, two forms of pattern that at each time stamp group objects by means of *density-based clustering*. Such an approach can be summarized in the following points (see also Figure 6.5c for an example):

- First, all objects that have a large number of neighbors are labeled as *core objects*; among the remaining objects, those that are neighbors of core objects are labeled as *border objects*; the remaining objects are labeled as *noise*;
- Second, core objects are grouped into clusters in such a way that each pair of neighboring core objects falls in the same cluster. Essentially, clusters are computed as transitive closure of the *neighbor* relation;
- Finally, border objects are assigned to the same cluster of their neighboring core objects;² while noise is discarded.

The neighbors of an object are all the objects at a distance not larger than a threshold r , and the minimum number of neighbors required to make an object a core object is also a parameter m . Therefore, we can see that a core object and its neighbors approximately satisfy the *closeness* requirements of a flock – more exactly, these are *density* requirements. The step forward here is that multiple compact groups can be merged together if they are adjacent (see the second step), in order to form larger ones. Besides their sheer size, the groups formed through this process can also have a relatively large extension (therefore not all pairs of objects in the cluster will be close to each other, because they actually are neighbors of neighbors of neighbors) and an arbitrary shape. In several contexts this can be useful, for instance in analyzing vehicle trajectories, since the road network simply forces large groups of cars to distribute along the roads (therefore creating a cluster with a snake-like shape) instead of freely agglomerate around a center (which would instead yield a compact, spherical-shaped cluster).

The key difference between moving clusters and convoys is the fact that convoys require that the population of objects involved in the pattern is always the same, while in moving clusters it can gradually change along the time: the only strict requirements are that at each timestamp a (spatially dense) cluster exists, and that when moving from a timestamp to the consecutive one the population shared by the corresponding spatial clusters is larger than a given fraction (a parameter of the method). A simple example of moving cluster that illustrates this point is shown in Figure 6.2: at each *time slice* a dense cluster is

² Notice that a border object might have two or more neighboring core objects belonging to different clusters. In this case one of them is chosen through any arbitrary criterion.

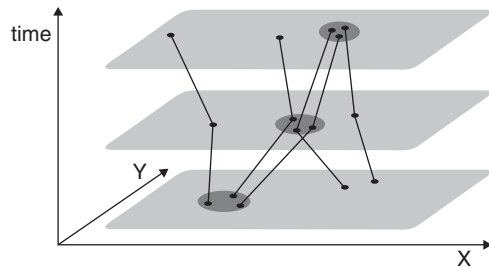


Figure 6.2 Visual example of a moving cluster over three time units.

found, formed by three objects, and any pair of consecutive clusters shares two of the three objects. This way, moving clusters that last a long time might even start from a set of objects and end in a completely different (possibly disjoint) set. In our example, only one object permanently belongs to the moving cluster. In some sense, the pattern is not strictly related to a population that generates it. The purpose of the pattern becomes to describe phenomena that happen in the population, not to find a group of individuals that do something peculiar consistently together.

One element of rigidity that affects both the patterns illustrated so far is the fact that they describe continuous portions of time. For instance, if a herd that usually moves compactly gets dispersed for a short time (for instance, due to an attack by predators) and later becomes compact again, both flocks and moving clusters will generally result into two different and disconnected patterns – the *before* and the *after* the temporary dispersion. One possible way to avoid this loss of information consists of allowing *gaps* in the patterns, that is, a pattern involves a set of timestamps that are not necessarily consecutive. In the literature we can find a solution of this kind, known as *swarm patterns*. Swarms are a general form of patterns that generalize flocks and moving clusters, as any spatial clustering method can be applied at the level of a single timestamp, and then spatial clusters belonging to different timestamps are linked (in case they share an appropriate fraction of population) regardless of their temporal distance.

6.2.2 Using Relative Time

In some contexts, the moving objects we are examining might act in a similar way, even if they are not spatially located together. For instance, similar daily routines might lead several individuals to drive their cars along the same routes, even if they leave home at very different hours of the day. Or, tourists who visit a city on different days of the year might actually visit it in the same way – for instance, by visiting the same places in the same order and spending there

approximately the same amount of time – because they simply share interests and attitude. This leads to a new category of questions, which can be well represented by the following:

Are there groups of objects that perform a sequence of movements, with similar timings, though possibly during completely different moments?

Patterns such as flocks and moving clusters can provide some answers to the question, but usually in small numbers, since the set of answers is limited to movements that happen synchronously among all objects involved. The question involves a much weaker constraint on the temporal dimension of the problem, and therefore might allow many more answers. In the following we will present one example of a pattern that goes in this direction and extracts spatio-temporal behaviors that are followed by several objects, but allowing any arbitrary time shift between them.

Trajectory patterns (T-Patterns) are defined as sequences of spatial locations with typical transition times, such as the following two:

$$\begin{array}{l} \text{Railway Station} \xrightarrow{15min} \text{Museum} \xrightarrow{2h15min} \text{Castle Square} \\ \text{Railway Station} \xrightarrow{10min} \text{Middle Bridge} \xrightarrow{10min} \text{Campus} \end{array}$$

For instance, the first pattern might represent the typical behavior of tourists who rapidly reach a museum from the railway station and spend there about two hours before getting to the adjacent square. The second pattern, instead, might be related to students who reach the university campus from the station by passing through the mandatory passage on the central bridge over the river. A graphical example is also provided in Figure 6.3a.

The two key points that characterize T-Patterns are the following: first, they do not specify any particular route among two consecutive regions described: instead, a typical travel time is specified, which approximates the (similar) travel time of each individual trajectory represented by the pattern. In the gap between two consecutive regions a trajectory might even have stopped in other regions not described in the pattern. Second, the individual trajectories aggregated in a pattern need not to be simultaneous, since the only requirement to join the pattern is to visit the same sequence of places with similar transition times, even if they start at different absolute times.

T-Patterns are parametric on three main parameters: the set of spatial regions to be used to form patterns, that is, the spatial extension of “Railway Station” and any other place considered relevant for the analysis³; the so-called *minimum*

³ Actually, the algorithmic tool provided in literature to extract T-patterns also contains heuristics to automatically define such regions, but in general the domain expert might want to do it manually in order to better exploit knowledge or to better focus the analysis, or both.

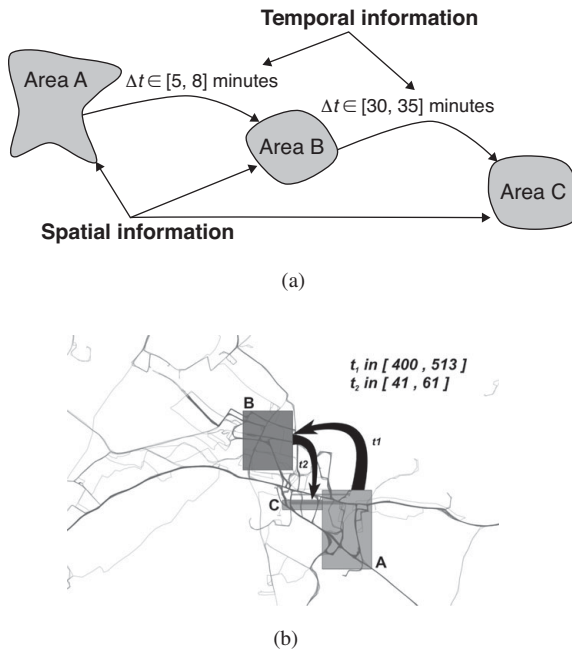


Figure 6.3 (a) Visual representation of a T-pattern and (b) sample result on a real data set.

support threshold, corresponding to the minimum size of the population that contributes to form the pattern (the parameter m for flocks); and a time tolerance threshold τ , which determines the way transition times are aggregated: transition times that differ less than τ will be considered *compatible*, and therefore can be joined to form a common typical transition time.

Figure 6.3a depicts an example of a T-Pattern on vehicle data describing the movements of a fleet of trucks. The pattern shows that there exists a consistent flow of vehicles from region A to region B, and then back to region C, close to the origin. Also, the time taken to move from region A to region B (t_1 in the figure) is around ten times greater than the transition time from B to C. That might suggest, for instance, that the first part of the pattern describes a set of deliveries performed by the trucks, while the second part describes the fast return to the base.

6.2.3 Not Using Time

In many cases it is interesting to understand if there are typical routes followed by significant portions of the population, that is:

Are there groups of objects that perform a common route (or segment of route), regardless of when and how fast they move?

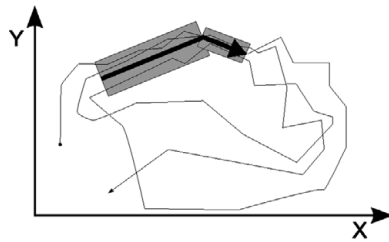


Figure 6.4 Visual representation of a spatio-temporal sequential pattern.

That means, for instance, that we are interested in what path an individual follows, but not the hour of the day he/she does it, nor the transportation means adopted: cars, bicycles, pedestrians, and people on the bus might follow the same path yet at very different speeds, resulting in different relative times. Also notice that we are interested here in routes that might be just a small part of a longer trip of the individual.

The mobility data mining literature provides a few definitions of patterns that can answer the question given above. In particular, we will briefly summarize one of the earliest proposals that appeared, at that time generically named *spatio-temporal sequential pattern* (in contrast, the trend in recent times is to assign elaborate and sonorous names to any new form of pattern or model).

The basic idea, also depicted in Figure 6.4, consists of two steps⁴: first, each trajectory is cut into quasi-linear segments, and then such trajectory segments are grouped based on their distance and direction, in such a way that each group is well described by a single representative segment (see the two thick segments in the figure); second, consecutive segments are joined to form the pattern. Frequent sequences are then outputted as sequences of rectangles such that their width quantifies the average distance between each segment and the points in the trajectory it covers. Figure 6.4 depicts a simple pattern of this kind, formed of two segments and corresponding rectangles. In particular, it is possible to see how the second part of the pattern is tighter than the first one, that is, the trajectory segments it represents are more compact.

6.3 Global Trajectory Models

A common need in data analysis at large is to understand the laws and rules that drive the behavior of the investigated objects. In the context of mobility data mining we refer to such laws and rules as (global) trajectory models, and in this area we can recognize three important representative classes of

⁴ The original proposal of this pattern considers a single, long input trajectory. However, the same concepts can be easily extended to multiple trajectories.

problems: dividing trajectories into homogeneous groups; learning rules to label any arbitrary trajectory with some tag, to be chosen among a set of predefined classes; and predicting where an arbitrary trajectory will move next. In the following we will introduce and discuss each of them.

6.3.1 Trajectory Clustering

In data mining, clustering is defined as the task of creating groups of objects that are similar to each other, while keeping separated those that are much different. In most cases, the final result of clustering is a partitioning of the input objects into groups, called *clusters*, which means that all objects are assigned to one cluster, and clusters are mutually disjoint. However, exceptions to this general definition exist and are relatively common.

While the data mining literature is extremely rich with clustering methods for simple data types, such as numerical vectors or tuples of a relational database, moving to the realm of trajectory makes it difficult to directly apply them. The problem is that trajectories are complex objects, and many traditional clustering methods are tightly bound to the simple and standard data type they were developed for. In most cases, to use them we need to adapt the existing methods or even to reimplement their basic ideas in a completely new, trajectory-oriented way. We will see next some solutions that try to reuse as much as possible existing methods and frameworks; then, we will discuss a few clustering methods that were tailored around trajectory data in the first place.

Generic Methods with Trajectory Distances

Several clustering methods in the data mining literature are actually clustering schemata that can be applied to any data type, provided that a notion of similarity or distance between objects is given. For this reason, they are commonly referred to as *distance-based* methods. The key point is that such methods do not look at the inner structure of data, and simply try to create groups that exhibit small distances between their members. All the knowledge about the structure of the data and their semantics is encapsulated in the distance function provided, which summarizes this knowledge through single numerical values, the distances between pairs of objects; the algorithm itself, then, combines such summaries to form groups by following some specific strategy.

To give an idea of the range of alternative clustering schemata available in literature, we mention three very common ones: *k-means*, *hierarchical clustering*, and *density-based clustering*.

k-means (Figure 6.5a) tries to partition all input objects into k clusters, where k is a parameter given by the user. The method starts from a random partitioning and then performs several iterations to progressively refine it. During an iteration, *k-means* first computes a centroid for each cluster, that is, a representative object

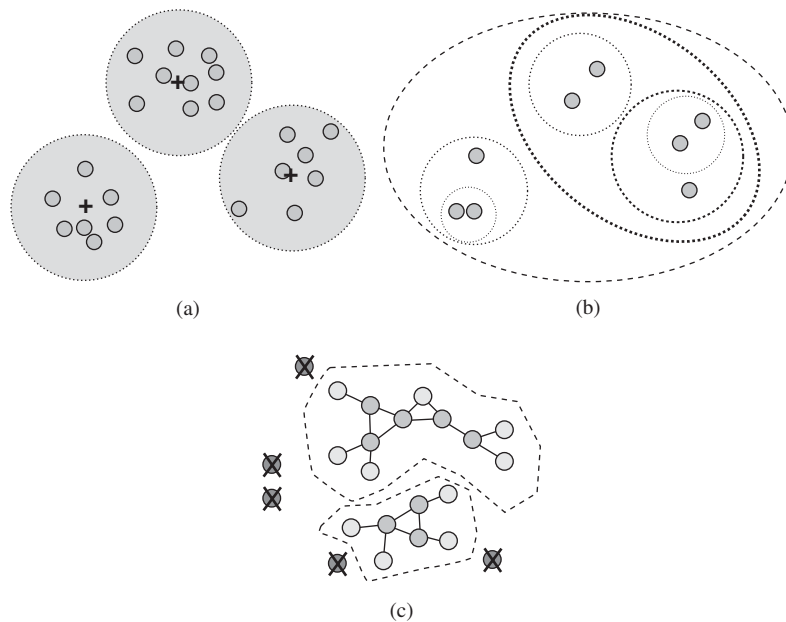


Figure 6.5 Example of clustering with different basic methods. (a) k -means. (b) Hierarchical. (c) Density-based.

that lies in the perfect center of the cluster⁵, then reassigns each object to the centroid that is closest to it. Such iterative process stops when convergence (perfect or approximate) is reached.

Hierarchical clustering methods (Figure 6.5b) try to organize objects in a multilevel structure of clusters and subclusters. The idea is that under tight proximity requirements, several small and *specific* clusters might be obtained, while loosening the requirements some clusters might be merged together into larger and more *general* ones. For instance, *agglomerative* methods start from a set of extremely small clusters – one singleton for each input object – and iteratively select and merge together the pairs of clusters that are most similar. At each iteration, then, the number of clusters decreases by one unit, and the process ends when only one huge cluster is obtained, containing all objects. The final output will be a data structure called *dendogram*, represented as a tree where each singleton cluster is a leaf, and each cluster is a node having as children the two subclusters that originated it through merging.

Finally, *density-based clustering* (Figure 6.5c), as already introduced in Section 6.2.1, is aimed to form maximal, crowded (i.e., dense) groups of objects,

⁵ Notice that such object is a new one, computed from those in the cluster. Therefore, some level of understanding of the data structure is needed here. When that is not possible, usually a variant is applied, called *k-medoid*, that selects the most central object of the cluster as representative.

thus not limiting the cluster extension or its shape and, in some cases, putting together couples of very dissimilar objects. Also, objects that cannot be linked to any cluster are labeled as noise and removed.

How does one choose the appropriate clustering method? While no strict rule can exist, a general hint consists of paying attention to some basic characteristics of the data and the expected characteristics of the output. For instance, if we expect that our data should form compact clusters of spherical shapes (i.e., they should agglomerate around some centers of attraction), then k -means is a good candidate, especially if the data set is large – k -means is known to be very efficient. However, the user should know the number k of clusters to be found in the data, or at least some reasonable guess. That can be avoided with hierarchical, agglomerative algorithms, since the dendograms they produce synthesize the results that can be obtained for all possible values of k , from 1 to N (the number of input objects). The choice of the most appealing k can be postponed after the computation, and be supported by an examination of the dendogram. However, hierarchical clustering is usually expensive (efficient variants exist, yet these introduce other factors to be evaluated), so it is not a good option with large data sets. Finally, density-based methods apparently do not suffer of any of the issues mentioned above, and are also more robust to noisy data, yet the resulting clusters will usually have an arbitrary shape and size – a feature that might be unacceptable in some contexts, and extremely useful in others.

Depending on the analysis task that the user wants to perform, once the clustering schema to be adopted has been selected, he or she needs to choose the most appropriate similarity function, that is, the numerical measure that quantifies how much two trajectories look similar. The range of possible choices is virtually unlimited. The examples that can be found in the literature include the following, approximately sorted in increasing order of complexity⁶:

- Spatial starts, ends, or both: Two trajectories are compared based only on their starting points (the origin of the trip), the ending points (the final destination of the trip), or a combination of them. The distance between the trajectories, then, reduces to the spatial distance between two points. When both starts and ends are considered, the sum or average of their respective distances is computed. The output of a clustering based on these distances will generally put together trajectories that start or end in similar places, regardless of when they do start/end and what happens in the rest of the trajectory.

⁶ Notice that distance computation is at the base of classical database queries such as range queries and k -nearest neighbors (see Chapter 3). Indeed, k -means involves a 1-nearest neighbor query in the cluster assignment step, while density-based methods execute a range query to compute the neighborhood of each point.

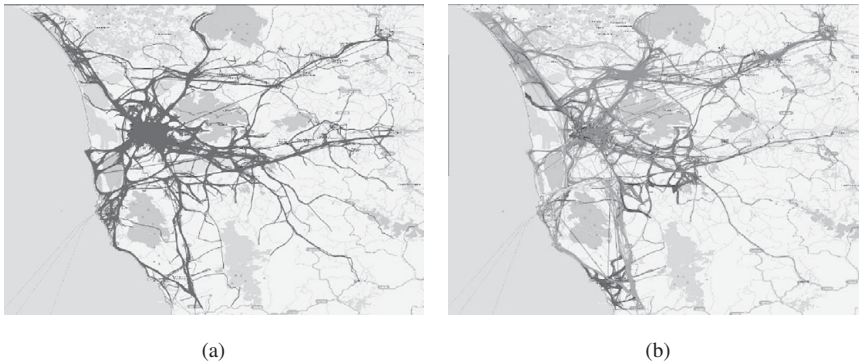


Figure 6.6 Sample trajectory clustering on a real data set of vehicles (GPS data collected by OctoTelematics S.p.A.), obtained using a density-based clustering schema and a spatial route distance function. (See color plate.)

- **Spatial route:** In this case, the spatial shape of the trajectory is considered, and two trajectories that follow a similar path (though possibly at different times and with different speeds) from start to end will result in a low distance.
- **Spatio-temporal route:** In this case, the time is also considered, therefore two trajectories will be similar when they approximately move together throughout their life.

Obviously, the selection of the clustering schema and the selection of the distance function might also be performed in the opposite order. Indeed, in some cases the choice of the distance to adopt is relatively easy or even enforced by the specific application, in which case the selection of the distance is performed first.

Figure 6.6b shows an example of a result obtained by a specific combination of schema and distance, namely a density-based clustering algorithm using the spatial route distance described above. Different clusters are plotted with different colors. The data set used in the example contains trajectories of vehicles in Tuscany, Italy, also plotted on Figure 6.6a.

Trajectory-oriented clustering methods. A complementary approach to clustering, as opposed to the distance-based solutions described so far, consists in algorithms that try to better exploit the nature and inner structure of trajectory data. From a technical point of view, that usually translates to deeply readapting some existing solution in order to accommodate the characteristics of trajectory data.

One important family of solutions makes use of standard probabilistic modeling tools. A very early example was provided by *mixture models-based clustering* of trajectories. The basic idea is not dissimilar from k -means: we assume that the data actually form a set of k groups, and each group can be summarized by

means of a representative object. The difference is that now the representative is a probability distribution of trajectories that fits well with the trajectories in its cluster. Another well-known statistical tool often adopted when dealing with trajectories is hidden Markov models (HMMs). The basic approach, here, consists of modeling a trajectory as a sequence of transitions between spatial areas. Then, a cluster of trajectories is modeled by means of a Markov model (i.e., the set of transition probabilities between all possible pairs of regions) that better fits the trajectories.

Other examples of trajectory-oriented clustering methods can arise by adding novel dimensions to the clustering problem. For instance, in the literature the problem was investigated of finding clusters by means of a distance-based clustering method (a density-based one, more exactly, though a similar process might be easily replicated for other approaches) when it is not known in advance the time interval to consider for clustering. For instance, we might expect that rush hours in urban traffic data exhibit cluster structures that are better defined than what happens in random periods of the day. The problem, then, becomes to find both the optimal time interval (rush hours were just a guess to be confirmed) and the corresponding optimal cluster structure. The solution proposed, named *time-focused trajectory clustering*, adopts a trajectory distance computed as the average spatial distance between the trajectories within a given time interval, which is a parameter of the distance. Then, for each time interval T , the algorithm can be run focusing on the trajectory segments laying within T . The quality of the resulting clusters is evaluated in terms of their density, and a simple procedure is provided to explore only a reasonable subset of the possible values of T . A sample result of the process is given in Figure 6.7, which depicts a set of trajectories forming three clusters (plus some noise) and shows the optimal time interval (that where the clusters are clearest) as dark trajectory segments.

6.3.2 Trajectory Classification

Clustering is also known as unsupervised classification, since the objective is to find a way to put objects into groups without any prior knowledge of which groups might exist, and what their objects look like. In several contexts such knowledge is available, more exactly in the form of a set of predefined *classes* and a set of objects that are already labeled with the class they belong to – the so-called *training set*. The problem, here, becomes finding rules to classify new objects in a way that is coherent with the prior knowledge, that is, they fit well with the training set. For instance, we might have access to a set of vehicle trajectories that were manually labeled with the vehicle type (car, truck, motorbike), and we would like to find a way to automatically label another, much larger set of new trajectories.

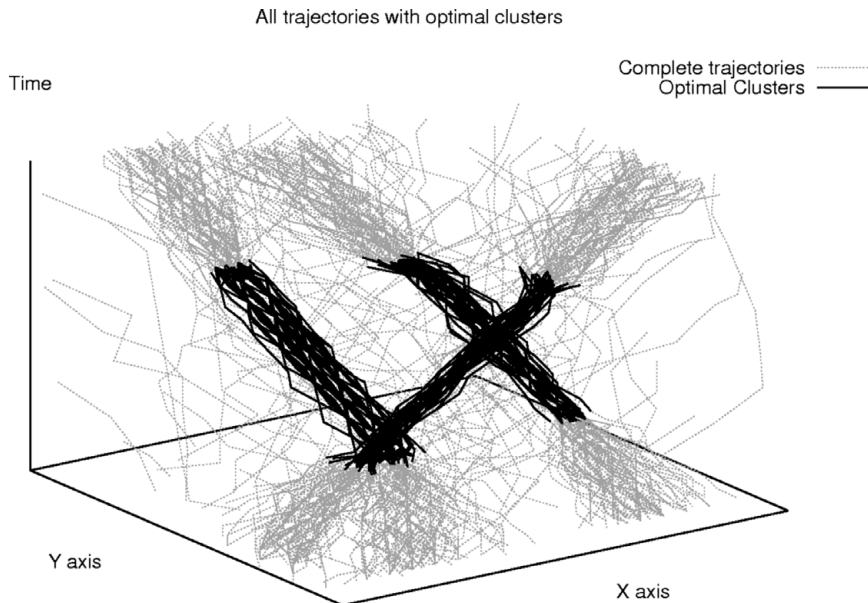


Figure 6.7 Three-dimensional depiction of sample result obtained with time-focused trajectory clustering on a data set of synthetic trajectories.

The simplest solution to the problem is the so-called *k-nearest neighbors* (*kNN*) approach: instead of inferring any classification rule, it directly compares each new trajectory t against the training set and finds the k labeled trajectories that are closest to t . The most popular label among the neighbors is then also assigned to t . The assumption is that the more similar two trajectories are, the more likely they belong to the same class. Obviously, everything revolves around a proper choice for the similarity measure applied, which should be as coherent as possible with the classification problem at hand. As an example, we can expect that a similarity function that takes into consideration the acceleration of objects will recognize well the vehicle type – the lighter the vehicle, the easier it is to reach high accelerations. On the contrary, a measure based only on the places visited might perform more poorly.

The same idea is also applied in several sampling-based solutions to the clustering problem: when the data set is too large to process, one approach consists of randomly sampling a small subset of trajectories and computing clusters on them. Then, all other trajectories are assigned to the cluster (i.e., classified) with a *kNN* approach or by comparing them against the centroid of each cluster.

Approaching the problem from a different viewpoint, each class involved in the classification problem could be modeled through a probabilistic model that is fitted to the available trajectories in the class. Then, each new trajectory can be

assigned to the class whose model most likely generated it. Similarly to what we have seen with clustering, HMMs are a common choice to do it. As compared to clustering, the problem is now simplified, since the association trajectories \leftrightarrow classes is known a priori. Behind the probabilistic framework they operate in, HMMs essentially aggregate trajectories based on their *overall* shape, again assuming that similar trajectories have better chances of belonging to the same class.

The final way to classify trajectories we will see is based on a traditional two-step approach: first extract a set of discriminative features by a preliminary analysis of the trajectories, then use such features – that can be expressed as a database tuple or a vector – to train any existent standard classification model for vector/relational data.

The first step requires one to understand which characteristics of the trajectories appear to better predict which class each trajectory belongs to. One straightforward approach might consist in calculating a predefined set of measures expected to be informative enough for the task. For instance, aggregates such as average speed of the trajectory, its length, duration, average acceleration, and diameter of the covered region might be used. Other, more sophisticated, solutions might instead try to extract finer aspects of the movement, tuned to calculate only the most useful ones. A proposal of this kind can be found in literature with the name *TraClass*, which heavily relies on a trajectory-clustering step. *TraClass* is based on a fundamental observation: in many cases, the features that best discriminate trajectory classes are related to a small part of the overall trajectory. All approaches mentioned so far, on the contrary, uniquely consider overall characteristics – that includes HMM-based solutions, since each model must fit whole trajectories. Single, short-duration events hidden in the long life of a trajectory might then be lost in the process. *TraClass* tries to fill in the gap by extracting a set of trajectory behaviors (which, we recall, look for local behaviors rather than overall descriptions of full trajectories). The basic tool adopted is trajectory segmentation and the clustering of such segments to form movement patterns.

TraClass works at two levels: regions and trajectory segments. At the first one, it extracts higher-level features based on the regions of space that the trajectories visited, without using movement patterns; at the second one, lower-level trajectory-based features are computed, using movement patterns. The extraction phase is made more effective by evaluating the discriminative power of the regions and patterns under construction. For instance, a frequent movement that is performed by trajectories of all classes will be not useful for classification (knowing that a trajectory contains such a pattern does not help in guessing the right class to associate to it); on the contrary, a slightly less frequent pattern that is mostly followed by trajectories of a single class is a very promising feature. In the proposed framework, trajectory partitioning makes discriminative parts

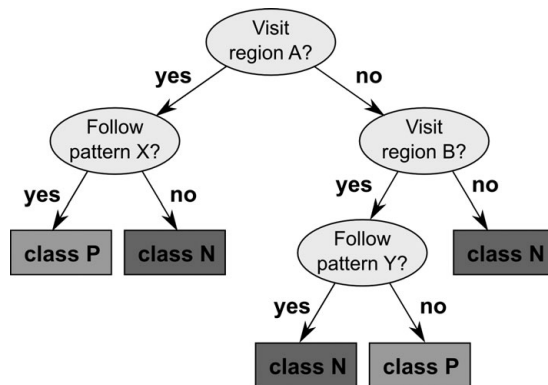


Figure 6.8 Sample decision tree on regions and patterns.

of trajectories identifiable, and the two types of patterns collaborate to better characterize trajectories.

Once a vector of features has been computed for each trajectory, we can choose any generic, vector-based classification algorithm. One representative (and easy to grasp) example is *decision trees*. The resulting classification model has the structure of a tree, whose internal nodes represent tests on the features of the object to classify, and the leaves indicate the class to associate to the objects. Figure 6.8 shows a fictitious example based on TraClass features, with two classes: positive (P) and negative (N). When a new trajectory needs to be classified, the test on the root (the top circle) is performed on it. In the example, if the trajectory actually visits region A, then we move to the left child of the root and continue the evaluation from there, otherwise we move to the right child. In the first case, we have now to test whether the trajectory follows pattern X: in case of a positive answer, the trajectory is labeled with “class P,” otherwise with “class N.” The classification process proceeds in a similar way when different outcomes are obtained, always starting from the root and descending through a path till a leaf is reached, which provides the label prediction. Another way to read a decision tree is as a set of decision rules, one for each path from root to leaf, such as “If (Visit region A) AND (Follow pattern X) THEN Class P.”

6.3.3 Trajectory Location Prediction

Trajectory classification can be seen as the problem of predicting a categorical variable related to a trajectory. However, prediction is most naturally related to the temporal evolution of variables. Since the basic aspect of objects in the context of trajectory is their location, predicting their future position appears to be a problem of primary interest.

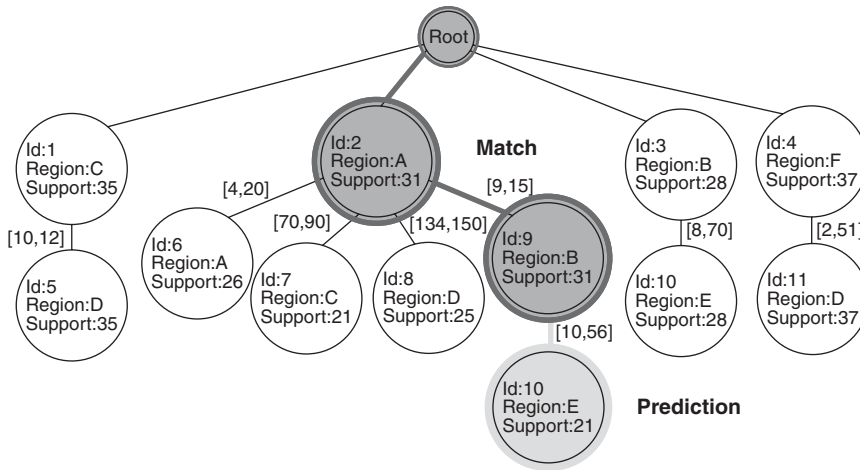


Figure 6.9 Sample prediction tree produced by WhereNext.

The modeling tools that are able to model the sequential evolution of the objects they describe are good candidates for a predictive usage. Indeed, once a trajectory has been associated to the most likely model (for instance, by choosing one of the k HMMs combined in a mixture-model, as described for the clustering problem), such model can be *run* to simulate the most likely next steps. In most cases we can apply the same remarks discussed earlier in this section for classification: if the model is based on an overall summary of the behavior of a set of trajectories, most likely it will not be able to capture local events, even though their appearance is highly correlated with a future behavior – in our case, the next location.

In literature it can be found an approach called *WhereNext*, which works in a way not too dissimilar from the one followed by TraClass for the classification problem. Basically, WhereNext extracts T-patterns (see Section 6.2.2) from a training data set of trajectories and combines them into a tree structure similar to a prefix-tree. In particular, each root-to-node path corresponds to a T-pattern, and root-to-leaf paths correspond to maximal patterns. Figure 6.9 shows a sample prediction tree, condensing 12 patterns, 7 of which are maximal (one per leaf).

When a new trajectory is presented, its most recent segment is compared against the regions represented in the tree, looking for the best match among the root-to-node paths. For instance, Figure 6.9 depicts the case where the last part of the trajectory visits region A followed by region B after a delay between 9 and 15 time units. The match is depicted by the dark shaded sequence. Then, the model finds that the matched sequence is a prefix of a longer pattern, and so it suggests as likely continuation region E (marked in light shaded in the figure), to be reached after a delay between 10 and 56 time units.

6.3.4 Trajectory Outliers

The general objective of clustering is to fit each object in data into some category (and discovering the categories is part of the problem). However, sometimes the analyst is exactly interested in those objects that deviate from the rest of the data set, and therefore cannot really fit any category. Such objects are called *outliers*.

Finding an outlier object means to discover some feature or pattern that holds for the object, and yet is anomalous or at least very rare in the data set. In this sense, the problem can be properly seen as a (infrequent) pattern discovery task. The reason for discussing it now is that most outlier detection methods in literature actually adopt some clustering procedure, and identify outliers as those objects that are (or would be) left out of any cluster. Here we provide two examples.

A basic method for discovering trajectory outliers consists in adopting a density-based clustering perspective, and therefore computing the number of neighbors of each trajectory over a reasonably large neighborhood. Then, the trajectories that have too few neighbors are classified as outliers. As density-based clustering, the method is parametric on the distance measure adopted, and therefore, in principle, any distance between trajectories can be applied. Alternatively, from each trajectory a set of predefined representative features can be extracted, such as average speed and initial position, and then applied any standard distance over vector data.

In Section 6.3.2 the *TraClass* trajectory classification method was presented, which has the characteristic of working over trajectory segments (obtained by properly cutting original trajectories) rather than whole trajectories. By clustering such segments, relevant subtrajectory patterns were extracted and later used for classification purposes. Following the same idea, outliers can be found within trajectory segments, therefore focusing on single parts of trajectory that behave in an anomalous way. In particular, each trajectory segment is compared against the representative segment of each cluster, and if no representative segment fits well enough, the input trajectory segment is classified as an outlier.

6.4 Conclusions

We conclude this chapter with a few notes on the topics presented and some of the open questions in mobility data mining research.

Mobility data mining, as many other instantiations of the general data mining paradigm into specific contexts, brings with itself the general categorization of problems and methods it inherited from standard data mining. In particular, the three main categories – frequent patterns, clustering, and classification – appear again. However, some specificities of trajectory data emerged and stimulated the development of new approaches. In particular, the complexity of the data,

joining temporal and spatial information, greatly increases the search space of most interesting problems, such as finding patterns or discovering discriminative spatio-temporal features for classification or prediction problems.

One aspect of mobility data mining that the reader might have guessed by reading this chapter is the fact that this research field still lacks an overall, comprehensive, and clear theoretical framework. Such a framework should be able to accommodate existing problems and solutions proposed in literature, as well as clarify the relations between them. Some examples of efforts in this direction exist in literature, and we also reported a few of them – for instance, the relation between local trajectory patterns and global trajectory classification models, and their abilities to grasp different, complementary kinds of discriminatory features of trajectory data; or the relations between some of the various forms of trajectory pattern. However, such cases are rather isolated, and at the present, providing an integrated view of methods and issues is still a largely unexplored part of the research field.

Another important point in mobility data mining is the fact that several data sources might provide information about the same mobility phenomena coming from different viewpoints. Each data source usually has distinctive characteristics, strong points, and limitations, and their integration might help in overcoming the limits of each of them. For instance, vehicle GPS data are usually very detailed in space (i.e., spatial uncertainty is small) and time (frequency of data acquisition is relatively high), yet it is inherently limited to the vehicles that are involved in the data collection process; instead, mobile phone service providers are able to collect information about mobility of all their customers, and through the collaboration of a few providers it is possible to cover the activity of very large portions of the real population. One example is *call detail records (CDRs)*, which describe the cell towers that served each call performed by each phone, together with the call's timestamp. CDRs allow us to build mobility trajectories for each customer served. However, such trajectories are very sparse (one point corresponds to a call, which are usually not so frequent) and spatially rough (a point actually represents the whole area served by the cell tower). Activities that try to combine these two data sources have begun to appear recently, with the aim of improving the representativity of GPS data through the extremely high penetration of the (spatially and temporally poor) CDR data.

Finally, so far, our discussion has always implicitly assumed that the trajectory data were analyzed offline and in a centralized setting, that is, by first collecting all data in a single database and then analyzing them. However, mobility data are usually massive and arrive as a continuous stream from the data source(s). Massiveness and the streaming nature of data leads to make it impossible to collect them, at a large scale, in a centralized database, and therefore analysis methods

need to be developed that exploit appropriate technologies, such as distributed databases (a paradigm where data are distributed along several data centers, to be queried to obtain the data needed for each specific analysis or computation step), distributed computation (several nodes with computation powers collaborate to analyze data), and streaming-oriented computation (essentially aimed to perform computation by looking at the input data only once).

6.5 Bibliographic Notes

As mentioned at the beginning of the chapter, the literature on mobility data mining is rather extensive – especially for such a young field – and heterogeneous. Attempting an exhaustive discussion of existing problems and proposals would require much more space and would be beyond our purposes as well. In the following, we will provide a list of essential bibliographic references for the reader, including those describing the methods cited in the chapter and a few pointers for further reading.

The original definition of flock patterns required that the group of objects meet at a single time instant and have the same direction of movement. Successive variants introduced the temporal duration constraint, also adopted in this chapter, starting from [Gudmundsson et al. \(2004\)](#). Moving clusters were defined by [Kalnis et al. \(2005\)](#), provided with a few heuristics for incrementally computing the interesting patterns, while convoys are described in [Jeung et al. \(2008\)](#) and spatio-temporal sequential patterns appear in [Cao et al. \(2005\)](#).

T-patterns were introduced by [Giannotti et al. \(2007\)](#), and later were exploited in building WhereNext – a location prediction method by [Monreale et al. \(2009\)](#) – as well as in several application works.

One rich source for a library of trajectory distances – to be used within generic clustering algorithms – is provided by [Pelekis et al. \(2007\)](#). Several references exist for standard (distance-based) clustering schema that can be applied to trajectory data, including basic introductions to data mining such as [Tan et al. \(2005\)](#).

Model-based approaches to trajectory clustering can be found in several isolated papers, especially on specific application domains (video surveillance, animal tracking, etc.). The mixture-models trajectory clustering described in this chapter was first introduced in [Gaffney and Smyth \(1999\)](#), later extended to include time shifts. Hidden Markov models-based approaches can be found, for instance, in [Mlich and Chmelar \(2008\)](#).

Time-focused clustering, an extension of density-based clustering for trajectories, was presented in [Nanni and Pedreschi \(2006\)](#).

The TraClass framework for trajectory classification was introduced in [Lee et al. \(2008a\)](#), mainly based on previous works of the same authors on trajectory

segmentation and clustering. The same principles were then applied to the outlier detection problem, as described in [Lee et al. \(2008b\)](#).

Finally, a few sources already exist for exploring more deeply the subject of data mining on trajectory data, including the book by [Giannotti and Pedreschi \(2008\)](#), which contains a chapter on spatio-temporal data mining, and the book chapter on spatio-temporal clustering by [Kisilevich et al. \(2010b\)](#).

7

Understanding Human Mobility Using Mobility Data Mining

Chiara Renso and Roberto Trasarti

7.1 The Mobility Knowledge Discovery Process

We often say that “Knowledge is power” and this is particularly true in the mobility field, because mobility knowledge gives great power in terms of appropriate decision making to actors in several application files, ranging from traffic management to urban planning to ethology, just to give some examples. Mobility knowledge can be rephrased as *how, when, where, and why objects move?* For example, a traffic manager could improve traffic sustainability in a city when he or she discovers why a specific traffic congestion happens, or an ethologist could finally gain a deep understanding of why a given animal behaves in a given way.

The mobility data mining research field has seen a growing interest in the last few year – as already stated in Chapter 6 – providing several algorithms and techniques tailored on trajectory data. However, a common problem of these techniques is that the knowledge produced by the mining step is generally not really applicable to the application domain as it is. The lack of semantics of the extracted patterns makes the interpretation task far from obvious. This problem, commonly recognized in the data mining literature, is particularly significant in mobility data mining, where the complexity of the data themselves, together with the extreme multifariousness of the mobility application requirements, makes the knowledge discovery even more challenging. The interestingness evaluation of the discovered movement patterns makes sometimes the KDD process a mere academic application, useless in reality. A way to close the gap between “KDD knowledge” and “application actionable knowledge” is to reinforce the KDD process with tools capable of easily managing the steps integrated with new elements focussed on the semantic aspect for an improved data and patterns understanding. This results in a number of techniques, tailored to mobility data,

aimed at enriching the steps of the knowledge discovery process with the final objective of getting meaning from movement data.

It is worth noticing that most of the methods to compute trajectory patterns, as illustrated in Chapter 6, are focused on the geometric properties of the trajectory. However, just discovering geometric trajectory patterns can be of limited interest since they lack the necessary semantics to be fully understood by the domain expert user. For example, assume that a cluster is found on a trajectory data set representing human movements in a city: this pattern represents *how* people are moving, not *why*, or the reason for the movement. The reason why entities move needs a deeper understanding of the movement based on the *context semantics*, or *context knowledge* – also called *background knowledge*. For example, we can discover that a particular movement happened due to a football match, or that a cluster represents tourists visiting attraction points in a city or commuters during their daily routine. The conceptual lift from finding *how* movement happened (e.g., a cluster) to understanding *why* entities are moving in that way (e.g., due to a commuting flow), needs an improved knowledge discovery process tailored to mobility data characteristics and possibly enriched with *contextual semantic information*.

We believe that mobility understanding and semantic enrichment encompasses the whole discovery process, from the preprocessing step to data mining and the pattern interpretation performed during the postprocessing step. For this reason, this chapter is centered on approaches devoted to improving the understanding of mobility data and patterns with the final objective of giving insights into *why* the movement happens. These approaches range from data preprocessing techniques, to mining, to postprocessing, where semantics have a more pervasive role.

We introduce this process by presenting a system called M-Atlas. M-Atlas provides the basic components for supporting the mobility discovery steps from data preprocessing to mining to postprocessing. We are introducing the system as a specific instance of a mobility knowledge discovery support system highlighting the steps that allow us to infer the new knowledge from trajectory data, possibly combined with semantic information. First of all, we highlight the importance of the preprocessing step to get a better perception of the values and the knowledge embedded in the data, and thus drive the mining task accordingly. During the mining and postprocessing phases the semantics have a major role since they are explicitly considered to interpret of the patterns in terms of mobility knowledge. Therefore, we present the core concepts of the M-Atlas system illustrating tasks such as *data preparation* for a context-driven preparation of data for mining or *data validation* where the mobility data set is evaluated against an application domain knowledge that acts as a “ground truth.” This latter step allows us to establish if, and at which degree, the data set to be analyzed are representative of the real world and therefore the results of

the analysis are still valid in the real world. Other techniques aimed at getting useful results from the mining step include progressive clustering and parameter tuning. The second part of the chapter focuses on trajectory behavior as already introduced in Chapter 1, distinguishing between spatio-temporal and semantic behavior. We illustrate how to extract semantic behavior such as “StuckInTraffic-Jam” or “Commuter” using a semantic-enriched mobility knowledge discovery process.

7.2 The M-Atlas System

M-Atlas¹ is a running system developed to handle all the steps of the mobility knowledge discovery process. M-Atlas is a querying and mining system based on extensions of SQL and centered on the concept of trajectory. Besides the mechanisms for storing and querying trajectory data, M-Atlas has mechanisms for mining trajectory patterns and models that, in turn, can be stored and queried. The basic design choice is *compositionality*, that is, querying and mining of trajectory data; patterns and models may be freely combined in order to provide the expressive power needed to master the complexity of the mobility knowledge discovery process. The conceptual model behind M-Atlas views the knowledge discovery process as the interaction between two conceptual worlds: the data world and the model world. The former is a set of entities to be mined, trajectories in our case; the latter is a set of models and patterns extracted from the data, representing the result of mining tasks. Two kinds of operators connect the two worlds: the *mining* operators, and the *entailment* operators. Mining operators map data into models, or patterns, while entailment operators map models, patterns, and data into the data that satisfy the property expressed in the given model or pattern. This view supports compositionality, as data can be mapped onto models and vice versa, coherently with inductive database vision. Another design choice of the system is that all entities are represented in the object-relational data model, which is more suitable to tackling the structural complexity of spatio-temporal data compared with the standard tabular data.

The M-Atlas system is equipped with a graphical user interface and a set of interactive graphical tools allowing the user to navigate the data and model easily. This has the advantage of making the tool usable by domain expert users to get full advantage of their domain expertise. Each interaction of the analyst with the interface is compiled into a sequence of M-Atlas queries that can be retrieved at any moment to describe or review the entire process. Alternatively, an expert data mining analyst can directly submit queries to the M-Atlas engine, to exploit its full expressiveness.

¹ Available for download at the address <http://www.m-atlas.eu>

The use of a data mining algorithm in a knowledge discovery process is not a straightforward process: usually the choice of the best algorithm and the best parameters setting to extract meaningful and useful patterns is difficult even for an expert analyst user.

In this section we introduce a set of techniques, demonstrated with examples using M-Atlas, to drive a user through the mobility knowledge discovery process by optimizing the data analysis and tuning the parameters setting. The techniques introduced here have been tailored to the case of mobility data, although they can be applied to general data mining.

7.2.1 Data Preprocessing

In this section we present some data preprocessing techniques useful in mobility knowledge discovery, illustrating them through the use of M-Atlas.

Data Validation

Data validation is a necessary step to measure how much the trajectory data set we are going to analyze is consistent and representative of the real world phenomena. Here we consider the data already cleaned and reconstructed as described in Chapter 2. However, the reconstruction step does not eliminate all the possible imperfections in the data and errors at higher level may still exist. This is due to bias in the data (e.g., tracking only a certain category of the users) or technological problems (i.e., an area where the devices don't work) that can produce unusual and unwanted effects on the analysis results. To assess the significance of a data set as a proxy of the real mobility phenomena within a certain area, the trajectory data set (as a set of spatio-temporal points) can be compared against a "ground truth" such as survey data composed by a set of interviews about mobility habits, for example done by phone (or other forms of a priori knowledge). However, an important issue to be considered in this comparison is the population of these two data sets. For example, considering the data set coming from a set of private cars, this covers only vehicular movements, whereas surveys usually include all kinds of movement, including pedestrians and public transportation. Second, the automatic collection procedure and the cleaning step applied for the car data set ensures that all movements are correctly captured, whereas surveys leave space for omissions or distortions. Finally, the data provide no explicit semantic information about the purpose of movements, such as the final destination and profiles of the citizens involved, whereas surveys explicitly collect this information. A significant difference holds also for the size of the sample, which can alter the reality represented in the data set. A method that can help to understand if the data are consistent with the ground truth is to replicate a statistic analysis for each data set and make a comparison. This

phase of the analysis is crucial, as only after assessing the correspondence of the preliminary analysis results with the ground truth can we proceed with the mobility knowledge discovery steps having the guarantee that the results will represent real mobility patterns. In Chapter 10, dedicated to car traffic monitoring, we will see an example of this validation process on the Milano data set.

Trajectory Reconstruction and Preparation

As explained in the previous chapter, the data mining algorithms apply to the concept of trajectory: but which trajectory definition? It is simply the ordered sequence of observations of the user's history? Or a subsequence representing the movements between stops? And how to define and compute a stop? Answering these questions is crucial and affects deeply the results of the knowledge discovery process. For example, if we are interested in frequent paths followed by a certain number of users we need to consider T-pattern applied to the whole user history as single trajectory, so that the support of a single pattern will be the number of users that follow that path. Alternatively, if we are interested in the usage of certain frequent paths then we do not need to distinguish between distinct users. As a consequence, the concept of trajectory to be mined becomes the subsequences of user trajectories delimited by two *stops* as described in Chapter 2. There are several ways of reconstructing trajectories considering different constraints and thresholds thus leading to different sets of trajectories. In M-Atlas we can perform this operation with the *data constructor statement*.

```
CREATE DATA <trajectory_table> BUILDING MOVING_POINTS
  FROM (SELECT userid, longitude, latitude, datetime
        FROM <raw_observation_table>
        ORDER BY userid, datetime)
  SET MOVING_POINT.<constraint_name> = <value> AND ...
```

The syntax of queries in M-Atlas extends the standard SQL. In this query we see a `CREATE DATA` operation building a new kind of data to be stored in the database from a pure relational table. As we can see, a new trajectory table is built from the raw observations using trajectory reconstruction parameters expressed in the *constraints*. Some examples are `MAX_TIME_GAP` or `MAX_SPEED` which realize the two constraints described in Chapter 2. These values depend on the application and their values have to be carefully chosen because they affect all the subsequent analysis. Examples are `MAX_TIME_GAP = 30 min` or `MAX_SPEED = 5 km/h` to cut trajectories where there is a temporal gap of 30 minutes or a max speed of 5 km per hour, respectively.

Data Manipulation

Before the execution of a data mining algorithm the analyst can manipulate (e.g., select the data in a particular area or period) or transform the data (e.g., anonymize for privacy reasons). To these purposes, the system provides a rich set of operations: the *relational statement* represents the creation of a relation between two objects applying a predicate while the *transformation statement* modifies the original data according to a transformation function (or algorithm). To better understand the *relation statements* and *transformation statements* we present two examples. The first one is the relation between a trajectories table and a temporal period table, which computes the temporal distribution of the movements:

```
CREATE RELATION <relation_table> USING INTERSECT
FROM (SELECT t.id, t.object, p.id, p.object
      FROM <trajectories_table> t,
           <time_periods_table> p)
```

The objective of this query is to create a new table where the trajectories are intersected with a temporal period. This is useful in the analysis process when the data to be mined have to be selected based on space and/or time, as in the example above. All the spatio-temporal operators embedded into the system – such as `INTERSECT` – assume a different meaning according to the types of data to which they are applied. For example, the `INTERSECT` operator when applied to two trajectories becomes a spatio-temporal intersection and this operation returns true when the two moving users are in the same place at the same time.

The second example is a transformation operation that builds a new set of trajectories to be mined. A classic example is the anonymization of trajectories, where the initial data set is transformed to guarantee a certain degree of anonymization of the trajectories. The main idea is that, in the anonymized data set, each individual is indistinguishable from other $k - 1$ individuals, as detailed in Chapter 9. However, for explaining the `TRANSFORMATION` operation it is important to point out that the original data set is changed into a new one with some properties that, in this case, guarantee the anonymity of the individuals.

```
CREATE TRANSFORMATION <trans_table> USING K-ANONYMITY
FROM (SELECT * FROM <trajectories_table> t)
SET K-ANONYMITY.K = <k_value>
```

7.2.2 Data Mining

Naturally, the mining step applies the mining algorithm. However, several actions can be taken during the mining step in order to make the knowledge discovery more effective. Moreover, models can be further manipulated and combined.

Data Mining Step

Data mining is the core step of the process and consists in the execution the algorithms, as for example the ones presented in Chapter 6. M-Atlas realizes this step with a *mining statement*:

```
CREATE MODEL <model_table> MINE AS <mining_algorithm_name>
  FROM (SELECT t.id, t.object
        FROM <trajectories_table> t)
  SET <mining_algorithm_name>.<param>= <value> AND ...
```

As we can see, this statement creates a new model as the result of a mining task specifying the mining algorithm to execute on a selection of trajectories where the algorithm has to be applied. This set is identified by the `SELECT` statement on the trajectories table having as attributes the ID (`t.id`) and the trajectory object (`t.object`). The `SET` component defines the algorithm parameters.

Mining a Data Sample

Applying a data mining algorithm to a large trajectory data set may be extremely time- and memory-consuming, making the direct application of the algorithm to the entire data set not possible due the time or memory limitation. This problem can be solved using the data mining algorithms presented in Chapter 6 in combination with data sampling techniques. In general, *sampling the data* is a technique to reduce the size of the data without altering the statistical properties.

The data can be sampled using semantic criteria such as dividing the data using the spatial or temporal characteristics of the trajectories. Whatever sampling technique is chosen by the analyst, the important issue is to maintain the consistency of the data or, at least, understand exactly the bias introduced, as this may strongly affect the extracted patterns.

An example of random sampling realized in M-Atlas is expressed as follows:

```
CREATE MODEL <model_table> MINE AS <mining_algorithm_name>
  FROM (SELECT t.id, t.object
        FROM <trajectories_table> t
        ORDER BY RANDOM()
        LIMIT 20%)
  SET <mining_algorithm_name>.<param>= <value> AND ...
```

We notice here the `RANDOM` keyword that allows us to reorder trajectories in a random way, selecting only the 20% of them. Once the models are extracted on the sampled data, we can apply them to the remaining data set to determine their real support. Chapter 10 presents an example of this technique for the Milano data set.

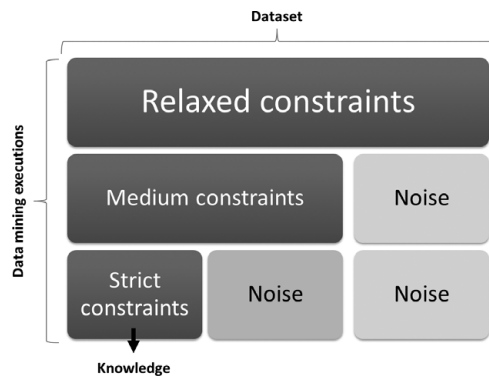


Figure 7.1 The process of extracting knowledge from the data using the progressive mining technique, restricting the constraint at each level.

Model Manipulation

Similarly to the trajectory data, the models resulting from the mining step can be stored and manipulated to produce a useful and meaningful representation of the trajectories behaviors. For this reason, the *relation statements* and the *transformation statements* can be used also on models. In particular, M-Atlas provides a relation that is a bridge between data and models called *entails*, which identifies the data that support a model, realized with the following query:

```
CREATE RELATION <relation_table> USING ENTAILS
  FROM (SELECT t.id, t.object, m.id, m.object
        FROM <trajectories_table> t, <models_table> m)
```

Notice the use of `ENTAILS` keyword in the query. The idea is to apply the entails operation to the join between trajectories and extracted models specified in the `SELECT` statement. This relation is crucial to the knowledge discovery process, as it implements the interaction of the process building complex progressive queries between data and models. This procedure is called *progressive mining* and it is illustrated in the following paragraph.

Progressive Mining

As described in previous sections, the knowledge discovery process is not a straightforward sequence where a single run of data mining algorithm can perform the whole understanding task. The iterative and interactive aspects are crucial to get a real understanding of the data and extracted patterns. The progressive mining technique is the concatenation of a series of mining algorithms, which restrict, at each step, their constraints, removing the *not interesting data* or *noise*. Figure 7.1 shows a graphical representation of the process where at each step the models are extracted and the data supporting them are reused

to apply a stricter version of the mining algorithm. An example is to use the T-clustering algorithm (see Chapter 6), reducing the allowed distance between trajectories or choosing different distance functions that become more precise at each level, such as the *starting points*, *route similarity*, and then *synchronized route similarity*. In M-Atlas, each step is realized as a sequence of two kinds of queries: a mining query to perform the clustering step and a relation query for the entails operation, that is, the selection of trajectories satisfying the cluster definition. This is depicted as follows:

```
CREATE MODEL <model_table> MINE AS T-CLUSTERING
  FROM (SELECT t.id, t.object
        FROM <trajectories_table> t)
  SET T-CLUSTERING.METHOD = <distance function> AND ...

CREATE RELATION <relation_table> USING ENTAILS
  FROM (SELECT t.id, t.object, m.id, m.object
        FROM <trajectories_table> t, <model_table> m
        WHERE m.id<>'noise')
```

The first query performs a clustering task on all trajectories. The following query uses both the resulting model table representing the clustering and the original trajectories data set to find trajectories that belong to some clustering, thus excluding the noise – here specified by the “noise” ID. In every step the classification of the noise can be both unsupervised, for example, the T-clustering, or supervised, where the user individually selects the interesting patterns extracted in the last data mining execution.

Chapter 10 illustrates examples of use of this technique on the Milano data set.

Tuning the Parameters

Tuning the parameters of the data mining algorithm is not easy, because it usually requires several attempts to evaluate the results and adjust the parameters values accordingly. In general, we must consider two aspects when dealing with the parameters settings: the number of patterns and the usefulness of the patterns. Usually the objective of the analyst is to find a *small* set of *useful and meaningful* patterns. Finding a good value for the parameters that guarantees this result is highly arduous. However, some techniques may be used to guess a reasonable value. Essentially, the idea is to progressively adjust the parameter values based on the characteristics of the resulting patterns. As an example, let us consider the T-pattern algorithm presented in Chapter 6, although similar methodology may be used for other algorithms. Recall that the parameters are the support threshold, the time tolerance, and an initial set of spatial regions and the algorithm finds

the most frequent sequences of regions visited by the users with their traveling time. The method we propose adjusts the parameters based on the analysis of the mining results. The objective is to iterate the mining task with different parameter values toward the objective considering the characteristics of the resulting patterns. Therefore, depending on the resulting set of patterns, an action must be taken as summarized here.

The result set is as follows:

- *Small and contains useful patterns*: In this case, the objective of the analyst is reached.
- *Too big or the algorithm is not terminating*: In this case, the support threshold is probably is too low and too many regions become frequent, leading to an explosion of patterns. There are three possible solutions: (1) to increment the support threshold, (2) check the set of regions to reduce them, or (3) increase the time tolerance so more patterns will be merged together.
- *Small, but time intervals are trivial*: The time tolerance is too high and makes the pattern too inclusive, leading to trivial ones. We need to lower the time tolerance.
- *Small, but the sequences of regions are trivial*: In this case, the support threshold is too high and the real patterns are hidden in the data or the set of regions is not meaningful. Some regions could be too large and therefore they can be split into a finer granularity, thus leading to a better differentiation in the resulting patterns.

When a reasonable result is obtained, the analyst can apply a pruning in the postprocessing phase to remove some of the patterns, considering additional properties such as the number of regions in a T-pattern. The parameter setting in any data-mining algorithm is recognized in the literature as an open issue and the optimal solution is far from being trivial. However, having a methodology to drive the parameter setting is a first step in searching for a good solution. Naturally, it could be that in some cases an algorithm is oversensitive to parameter changes, thus making it extremely difficult to find a good parameter setting.

The problem of finding a good initial parameter configuration is also worth a discussion: the analyst can simply start from a reasonable or random set of thresholds and then start tuning the parameters as described earlier. Another, smarter possibility is a parameter estimation performed considering the critical steps of the algorithm. Consider again the basic step of the T-pattern algorithm: the detection of frequent regions in the area under analysis makes the support threshold the most influent parameter for the whole process. We present a heuristics data-driven method to estimate the value for this threshold. This is based on the cumulative frequency distribution of trajectories in the spatial grid cells. An example on the Milano data set is shown in Figure 7.2a. The points

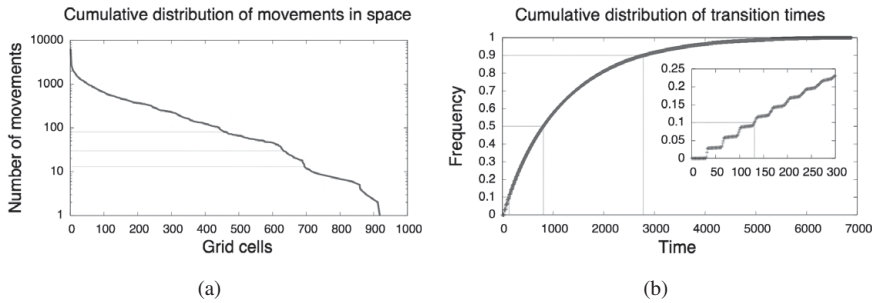


Figure 7.2 Cumulative frequency distribution of trajectories in space. (a) The plot proposes a ranked list of three candidate values for the T-pattern support threshold (13, 24, 82) based on detected points of significant slope variation. (b) Cumulative distribution of transition times between each pair of points in each trajectory.

of significant slope change in this distribution are the best candidates for the support threshold, because these points separate groups of grid cells that have a rather uniform frequency internally, while the frequency between the different groups is very different.

Another crucial parameter for the extraction of T-pattern is the time tolerance τ . In Figure 7.2a we plot all the time distances for every possible pair of points in each trajectory. These represent all the possible transition time candidates in the T-pattern mining algorithm. The sharp steps in the zoomed inset are the artifact of the average sampling rate, ≈ 33 seconds. This is the minimum admissible value for the τ parameter. We note that with a high value of τ the T-pattern computation aggressively merges the transition times. For instance, with 130 seconds 10% of transition times are merged. An adequate candidate for the τ parameter is around the 50th percentile (14 minutes) and, in any case, between the 10th and the 90th percentiles (2 minutes–45 minutes). The frequency distribution of trajectories in M-Atlas is realized computing the *intersection* between the spatial grid and the set of trajectories as specified in the following query:

```
CREATE RELATION intersection_table USING INTERSECT
FROM (SELECT t.id, t.object, s.id, s.object
      FROM <trajectories_table> t, <grid_cells> s)
```

and grouping the result by the cells. We see the use of the `RELATION` query with the `INTERSECT` operation that here applies to trajectories and spatial objects. Once the presence of trajectories in the spatial cells is computed by this query, a frequency distribution has to be computed. Therefore, the cumulative distribution and the identification of the slopes can be done using the mathematical functions provided by standard SQL.

7.2.3 Trajectory Postprocessing

Postprocessing refers to the set of operations that can be done once the mining step has been concluded and usually refers to the evaluation or interestingness of the extracted patterns. The validation of the patterns aims at measuring how much the extracted patterns are valid and not just random results. The patterns interpretation task, instead, is more semantics in the sense that it aims to interpret the patterns in the light of a domain knowledge. The result of this step may trigger a new iteration of the knowledge discovery process.

Pattern Validation

The validation of a set of discovered mobility patterns can be very arduous, as the useful patterns are usually not already known or are trivial behaviors. Comparing the result with domain knowledge, such as a survey (when available), can be useful to validate the knowledge discovery methodology although not appropriate for validating the discovered patterns. In other words, the patterns need an interpretation step that can be done with the participation of the domain expert user or exploiting some form of contextual information, which proves the real usefulness and interestingness of the patterns.

Although the direct assistance of the domain expert in the validation of patterns remains the optimal solution, it is in practice not easy to realize due to the general lack of availability of domain experts. However, there are alternative methods to evaluate some properties of the set of patterns that can be used to help the interpretation. Here, we present two examples that are general and valid for the patterns described in the previous chapter.

The first example is to study the *stability* of a set of extracted patterns over time. When a pattern happens to be stable over time this means it probably reflects a common behavior in the reality. The idea is to compute the patterns with the same parameter values for several time slots (i.e., several weeks or days). If a pattern has the same relative support for all the temporal intervals, it means that the pattern is stable and this confirms the pattern as a regular behavior and not an exception that happens only occasionally.

A similar approach is to study the evolution of the patterns over time. This can be done by extracting the patterns in different time intervals (e.g., days or weeks) and then trying to match them in order to build the *evolution* of the pattern through time. This is useful to understand how the patterns temporally evolve. Although similar to the previous case, this method is more difficult to realize because a distance measure has to be defined over patterns. Considering, for example, the T-flock algorithm: we can discover a set of patterns in the first day $P = \{p_1, p_2\}$ and a second set of patterns in the second day $P' = \{p'_1, p'_2, p'_3\}$, then match and compare them using a distance measure $f(p, p')$ to *link* together the closest patterns. Once we have built the evolution of the patterns over time

we can understand when the patterns remain similar, when and how they change, or when they disappear.

Pattern Interpretation

The intrinsic difficulty of behavior extraction lies in the need of integrating into the discovery process the contextual knowledge. We define *contextual knowledge* any kind of information that is not only related to the geometric parts of a trajectory and that has some relation with the mobility data. Examples of contextual knowledge are: the geographical environment where the objects move (e.g., hotels, roads, parks), any nongeometric moving object feature (e.g., the age of the tracked person), or the application-specific concepts and behavior (e.g., goal of the movement or predefined behavior, such as commuting, shopping, or touring).

Application domain knowledge may be globally represented by formally encoding it into a knowledge representation structure such as an *ontology*, which can be used to represent the main concepts of the application. Formal ontologies are described by languages that are formal and machine readable. They often include reasoning facilities that support the automatic processing of that knowledge. Standards such as description logics (DL) provide a deductive inference system based on a formal, well-founded semantics. The basic components of DL are suitable to represent concepts, properties, and instances. Complex expressions, called *axioms*, can be used to implicitly define new concepts. Combining ontologies with data mining is an intricate, challenging, and growing research field. Besides, in the case of mobility, additional difficulties due to the complexity of the managed data and patterns make this combination even more arduous. The lack of primitive spatio-temporal ontology representation and reasoning mechanisms is the major obstacle for the successful development of this trend.

Some recent proposals are making the first steps in this direction involving contextual knowledge in the form of ontologies. An interesting feature of combining data mining with ontologies in the knowledge discovery process is the possibility of integrating deduction and induction aspects. The inductive power of the data mining, extracting patterns from data (bottom-up), is enriched with the possibility to deductively infer additional information based on some application domain knowledge (top-down). This combination allows us to classify the mobility patterns, as extracted from the mining step, into the application knowledge concepts encoded in the ontology. An example of this induction–deduction combination is the framework Athena, an extension of M-Atlas that is an attempt to exploit ontologies in the mobility knowledge discovery process. Athena represents application domain knowledge in an ontology where axioms define the behavior we want to find in the data. Therefore a classification of the extracted pattern into predefined behavior is performed directly by the ontology

reasoning engine. An example of the use of Athena is reported in the following section.

7.3 Finding Behavior from Trajectory Data

The objective of the mobility KDD process is to give an understanding of movement data, starting from the statistical analysis that gives an indication of the properties of the data set, to the extraction of local patterns and global models that show the hidden correlations of the geometric aspects of the trajectories. However, these steps alone may be not enough for a proper understanding and interpretation of mobility data in terms of movement behavior.

The main outcome of a mobility knowledge discovery process is to extract behavior from raw trajectory data, thus performing a sort of progressive semantic enrichment from the raw data to a semantic behavior. Trajectory behavior can be of different types, from the behavior based only on the geometric properties of the trajectories to the more semantic-oriented behavior involving domain knowledge and other sources of semantic information. In the following, we propose a classification of such behavior types, introducing some examples. Later in the section we show a couple of examples of how these behaviors can be extracted from raw data using the M-Atlas methodology.

7.3.1 Spatio-Temporal Behaviors

These behaviors are characterized by the geometric properties of the trajectories. They can be individual when defined on a single trajectory or collective when multiple trajectories are involved in the behavior definition. Examples of *individual* spatio-temporal behaviors are as follows:

- *Residence*: A trajectory shows the Residence behavior within the area A for the time interval I if during the whole time interval I all its spatio-temporal positions are located inside the area A .
- *SystematicMovement*: A trajectory represents a systematic movement if it entails a frequent movement pattern of the user over a time period.

Examples of *collective* spatio-temporal behavior are:

- *Flock*: A set of trajectories shows the Flock behavior during a given time interval I when all the trajectories of the set stay close to each other during the time interval I . Or more precisely: at each instant t of the time interval I there is a circle such that (1) its radius is smaller than a given threshold, and (2) it contains the positions at t of all the trajectories.
- *Convergence* (also called *Encounter*): A set of trajectories shows the Convergence behavior if every trajectory of the set roughly passes by the same point at the same instant.

- *Leadership*: Let S be a set of trajectories showing the *Flock* behavior. A trajectory T of S shows the Leadership behavior during some given time interval I if, during I , each time the flock S moves, T is ahead of the other trajectories of the flock S .

7.3.2 Semantic Behaviors

These behaviors are identified by semantic properties of the trajectories. Again, we can distinguish between individual or collective behaviors. *Individual* behaviors include the following:

- *Home*: The most frequent place where the user's trajectories are *resident*.
- *Work*: The second most frequent place where the user's trajectories are *resident*.
- *HomeToWork*: A trajectory shows the HomeToWork behavior if the trajectory starts in the Home place and ends in the Work place.
- *CommuterMovement*: A trajectory that is a SystematicMovement and a HomeToWork where the Home is "outside" the city urban area and the Work is "inside" or vice versa.

Examples of collective semantic behaviors are described as follows:

- *StuckInTrafficJam*: A car trajectory shows the StuckInTrafficJam behavior if it is part of a Flock where the speed is always lower than 1/4 of the free speed in that area.
- *Events*: A public place where several trajectories Converge and then Reside for a time interval I .
- *Tourist Guide*: A pedestrian trajectory that is a SystematicMovement starting from a place labeled as "information center" which becomes Leader of a group of trajectories.

The translation of the behavior definitions into a composition of the three main KDD steps can be performed in the M-Atlas system. In Figure 7.3 we present the flow of operations needed to extract the StuckInTrafficJam behavior, used as an example. In the following we present the set of queries that implement the process steps.

7.3.3 Extracting Behavior

Consider a table called `Observations` containing the raw points collected by the GPS devices for a number of users. This table is composed of four columns: `userID`, `longitude`, `latitude`, and `timestamp`. The first M-Atlas query implements the *data construction* step, which builds the trajectories according to the spatio-temporal constraints of 3 hours and 50 meters, respectively. This step

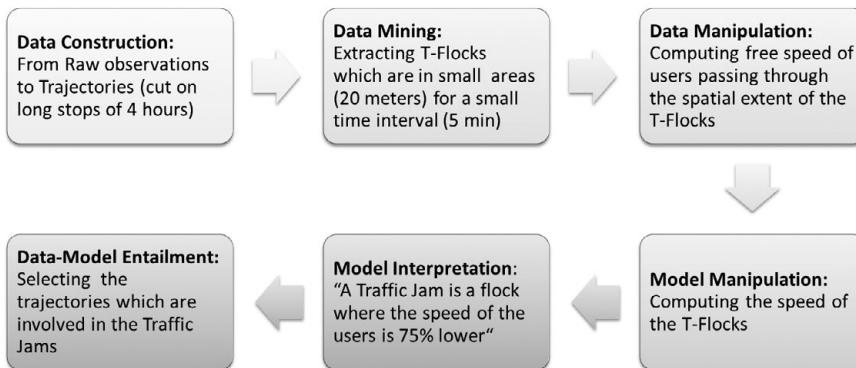


Figure 7.3 The flow of operations needed to extract the StuckInTrafficJam behavior.

of trajectory reconstruction cuts the movement of the users into trips describing real activities and thus avoids long stops, for example, during the night or during the working time.

```

CREATE DATA Trajectories AS MOVING_POINTS
  FROM (SELECT t.userID, t.lon, t.lat, t.timestamp
        FROM Observations)
  SET MOVING_POINTS.MAX_TIME_GAP = 3 hours AND
      MOVING_POINTS.MAX_SPACE_GAP = 50 meters
  
```

The result is a table called `Trajectories` with the three columns `userID`, `trajID`, and `trajectory`. We can notice that the trajectory becomes a data type in the system. For space reasons we skip other possible preprocessing steps and we proceed with the *data mining step* using the T-flock algorithm – described in Chapter 6 – to obtain the groups of at least 10 cars with a maximal distance of 20 meters between them for a time period of at least 5 minutes. These parameters have been chosen as a reasonable approximation of what a candidate traffic jam represents. However, this strongly depends on the application and the characteristics of the analyzed urban area. Typically, bigger cities require larger parameters for the flock to be identified as a traffic jam.

```

CREATE MODEL Flocks USING T-FLOCK
  FROM (SELECT trajID, trajectory FROM Trajectories)
  SET T-FLOCK.MIN_SUPPORT = 10 AND
      T-FLOCK.MAX_SPACE_GAP = 20 meters AND
      T-FLOCK.MIN_DURATION = 5 minutes
  
```

Once again, the result is stored in a table called `Flocks`. At this point we have computed a spatio-temporal behavior, though we need to go a step further toward a semantic behavior, like the `StuckInTrafficJam`. There are still a few tasks to be done to identify traffic jams as the flocks with a low speed when the

semantic aspect is considered. In this example, the free speed² in the analyzed area is a contextual information to be taken into account when defining a traffic congestion. The steps to be performed are: (1) to compute the free speed in the area of each flock using through a *data manipulation* step; (2) to select only the flocks with a speed lower than 1/4 of the computed free speed applying a *model manipulation step*. For the first task, we need to find all the trajectories passing in the area where the flock is found and compute their velocity. This allows us to compute the free speed that is compared later with the flock speed. Therefore, we use the spatial transformation `Intersection` between the T-flock pattern and a trajectory:

```
CREATE TRANSFORMATION SubTrajectories USING INTERSECTION
  FROM (SELECT flockID, flock FROM Flocks),
        (SELECT trajID, trajectory FROM Trajectories)
  SET INTERSECTION.ONLY_SPATIAL = true
```

The resulting table `SubTrajectories` contains the parts of trajectories that intersect only the spatial extent of the Flock (using the `ONLY SPATIAL` parameter): in other words, we are considering the whole set of vehicles that pass in that area in the period of analysis. From this set of subtrajectories we extract the average speed as an estimation of the free speed:

```
CREATE TRANSFORMATION FreeSpeeds USING STATISTICS
  FROM (SELECT flockID, trajID, trajectory
        FROM SubTrajectories)
```

The `STATISTICS` constructor indicates a set of predefined trajectory statistics including the average velocity, all stored in a table. The second task is the computation of the speed of the Flocks to be compared with the free speed.

```
CREATE TRANSFORMATION FlockSpeeds USING STATISTICS
  FROM (SELECT flockID, flock FROM Flocks)
```

To identify the Flocks that are traffic jams we use a *model interpretation* step where we constrain the set of Flocks using the definition of a traffic jam shown in Figure 7.3:

```
CREATE TABLE TrafficJams AS
  SELECT f.flockID, f.flock
  FROM FlockSpeeds s, Flocks f, FreeSpeeds fs
  WHERE s.flockID = f.flockID AND
        s.flockID = fs.FLOCK ID AND
        s.avg_speed <= fs.avg_speed*.25
```

² The term *free speed* indicates the average speed of a vehicle in a road without obstacles such as traffic lights, accidents, or traffic congestion.

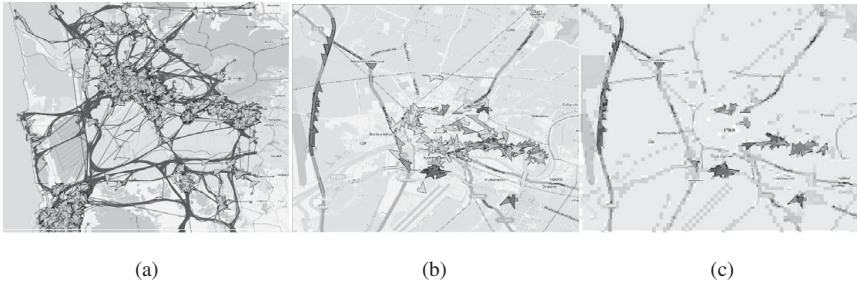


Figure 7.4 A graphical representation of the process of extracting traffic jams from the data. (a) Using the T-flock algorithm all the candidates are extracted. (b) The patterns are colored based on ratio between their speed and the free speed in the same area (Blue >1 , Red <1). (c) The patterns with a speed lower than 1/4 of the free speed. (See color plate.)

Once we have the traffic jams, we can retrieve the trajectories of the users who are stuck there using the a *data-model manipulation* realized through the `Entail` relation predicate:

```
CREATE RELATION StuckInTrafficJam USING ENTAIL
  FROM (SELECT flockID, flock FROM TrafficJams),
  (SELECT userID, trajID, trajectory)
```

The obtained table contains the set of trajectories of the users who are part of a traffic jam (identified by `flockID`). In Figure 7.4 we visualize some of the steps on the map. However, it is important to notice how this process does not complete the understanding of mobility. In fact, the selected trajectories can be further analyzed to determine, for example, the reasons of the traffic jams. An example is to combine the `StuckInTrafficJam` with the `Commuter-Movement` to discover a possible relation of a traffic jam with the commuting behavior.

We have seen in the previous example how the semantic information is embedded into the discovery process when passing from a spatio-temporal behavior to a semantic behavior, for example, passing from the flocks to the `StuckInTraffic` behavior. We have used domain information in the M-Atlas queries to identify the semantic behavior from the extracted flocks. However, the semantic enrichment step is not explicit in the process and it is somehow embedded into the M-Atlas queries by the analyst. A further step in the direction of extrapolating and modularizing the semantic enrichment task from the KDD process is to define the KDD process as a combination of induction (or mining) and deduction (inference of a semantic behavior) reasoning tasks. The framework Athena offers a solution: an extension of M-Atlas exploiting the integration of ontologies in the mobility knowledge discovery process. Essentially, this new process consists of a querying and mining process enhanced with

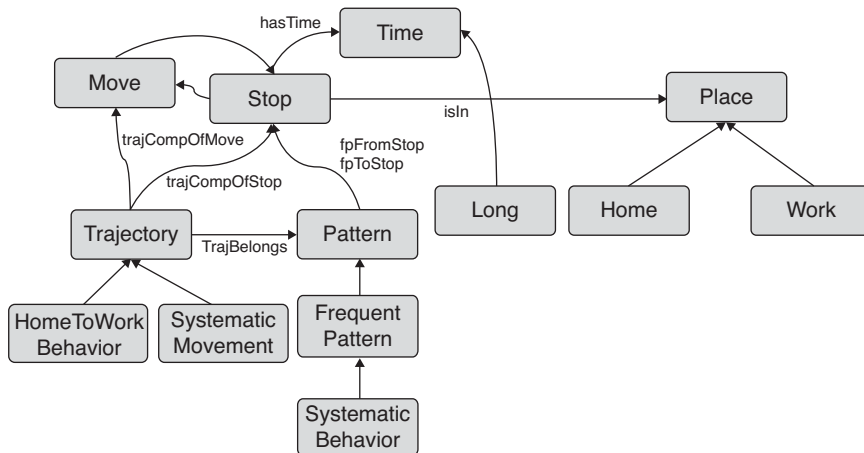


Figure 7.5 A fragment of the ontology used in Athena to discover HomeToWork behavior

reasoning tasks. Athena represents application of domain knowledge into an ontology and a mapping between ontology concepts to data and patterns is defined. Ontology concepts represent the data (e.g., trajectories, roads), patterns (e.g., flocks), and semantic behavior (e.g., StuckInTrafficJam). The ontology embeds the semantic of the domain application and, particularly, the concepts defined by axioms define the semantic behavior we want to infer from patterns and data.

We clarify now using as an example the CommuterMovement behavior. We can represent this behavior in the ontology as an axiom defining *a trajectory moving from outside the city in the morning, stopping a long time in the city center, then moving back from center to the outside in the afternoon*. The mapping between the trajectories, patterns, and the ontology is formalized in a specific mapping file, so that the trajectories, the mobility patterns, and the geographical knowledge become instances in the ontology. The ontology reasoning engine is run to classify patterns and trajectories into the appropriate behavior as defined by the axioms (e.g., the trajectories satisfying the HomeToWork behavior).

In Figure 7.5 we present an example of the ontology definition for the HomeToWork behavior. We can see that HomeToWork and SystematicMovement are subclasses of trajectory because they represent individual behavior, while SystematicBehavior is defined as a special kind of frequent pattern, which in turn is a kind of pattern, thus representing collective behavior.

A special function called `SEMANTIC(object)` is defined in M-Atlas with the objective of returning all the ontology concepts in which the `object` has been classified by the mapping file or by the inference engine. For example, a given trajectory may belong to either the class “Trajectory” as defined by

the mapping file, or can belong to the class “HomeToWork” as inferred by the deductive step by the ontology engine. Therefore, the query:

```
SELECT t.id, t.trajectory
FROM trajectories t
WHERE 'HomeToWork behavior' in SEMANTIC(t.trajectory)
```

returns all the trajectories classified as HomeToWork by the deductive step based on the axioms definition and the ontology inference engine. Now we can combine the results with the concepts of *inside area* and *outside area* of the city contained in a table called Areas to extract the CommuterMovement. This is depicted by the following query:

```
CREATE TABLE CommuterMovement AS
SELECT t.trajId, t.trajectory
FROM trajectories t, areas a, areas a2
WHERE 'HomeToWork behavior' in SEMANTIC(t.trajectory)
AND 'Systematic movement' in SEMANTIC(t.trajectory)
AND ST_contains(ST_PointN(trajectory,first), a.area)
AND ST_contains(ST_PointN(trajectory,last), a2.area)
AND a.label = 'outside' and a2.label = 'inside'
```

where `ST_contains`³ is a spatial predicate which checks if a point is contained in a specific area. In Figure 7.6 we show an example of the resulting trajectories. Similarly to the TrafficJam case presented above, we can combine the results of these two analyses, obtaining the commuters who are stuck in a traffic jam.

```
SELECT sj.userID, sj.flock
FROM CommuterMovement cm, StuckInTrafficJam sj
WHERE cm.trajID = sj.trajID
```

7.4 Conclusions

This chapter introduced a step-by-step KDD process for mobility understanding by using examples from the use of the M-Atlas system. We have shown that the understanding of mobility data is a complex process that involves many different steps, all of which are necessary for the proper understanding of the mobility phenomena. These steps are presented here explaining the techniques that have to be applied to find meaningful behaviors. During this process we observe an increasing involvement of semantic and contextual information embedded progressively into the process. We have defined, as the final result

³ This function and the other used in the query derive from PostGIS and they can be used directly in the M-Atlas system.

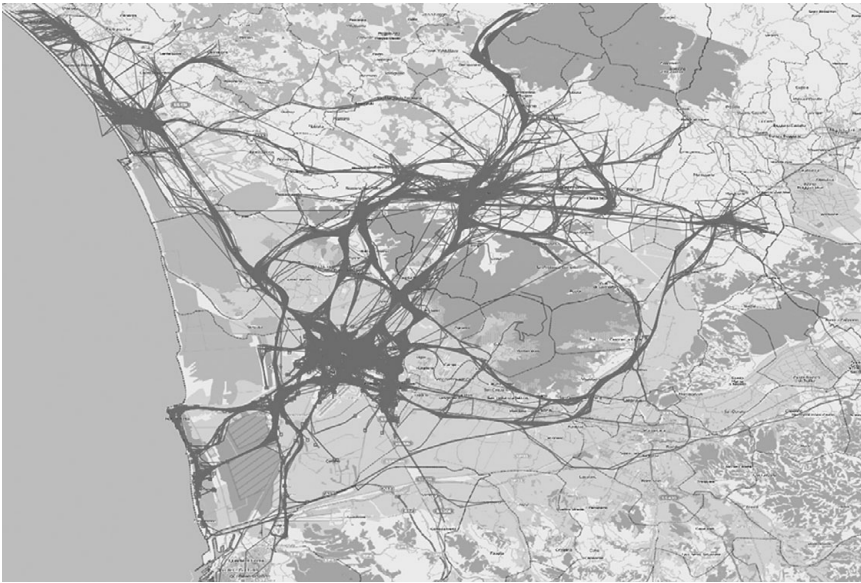


Figure 7.6 The trajectories obtained from the commuter movement analysis.

of this semantic-enriched mobility knowledge discovery process, the concept of semantic trajectory behavior. To reach this objective the semantic information may be integrated into the process mainly in two ways. On the one hand, it is the KDD analyst who, using the M-Atlas primitives, finds the semantic behavior properly exploiting the system functions with appropriate parameter. On the other hand, we also have pointed out the possibility of using ontologies during the postprocessing step to represent explicitly the semantic information and thus automatize the discovery process. In this case we have added an automatic deductive step where application domain knowledge is explicitly represented in the process. In conclusion, the main message derived from the experience of extracting behavior from data is that data mining alone – even when applied to large masses of trajectory data – is not enough to transform data into knowledge; we need a more complex process involving semantic information.

7.5 Bibliographic Notes

The knowledge discovery process was first introduced in [Fayyad et al. \(1996\)](#) for the relational case where the main KDD steps are presented and discussed. The KDD process presented here is the one that is at the basis of most of the data mining and knowledge discovery research. The mobility knowledge discovery process proposed here is basically the Fayyad one, adapted for the trajectory case and eventually enriched with a new deductive step with ontologies. In

fact, this step was not in the original process proposed by Fayyad, but was introduced in the Athena system of [Baglioni et al. \(2012\)](#) where experiments on two trajectories data sets representing cars and pedestrians were reported and discussed.

The M-Atlas system has been introduced here as an example of a system supporting the mobility knowledge discovery process with illustrating examples along the chapter. This tool takes inspiration from the inductive database vision by H. [Mannila \(1997\)](#) and it was originally introduced in [Giannotti et al. \(2011\)](#). There, the experiments were run on two GPS data sets collecting car trajectories from two Italian cities. Parts of these experiments are illustrated in Chapter 10. The implementation of the system is based on PostGIS spatial database system, from which many of its spatial operators have been inherited.

The techniques presented in the preprocessing step derive from literature works. For example, a survey on data set sampling techniques is presented in the book by [Scheaffer et al. \(2005\)](#) while progressive clustering on trajectory data is introduced in the paper by [Rinzivillo et al. \(2008\)](#).

An approach for mobility understanding that has not been presented here is data mining on semantic trajectories, represented as sequences of stops and moves. In this case, standard data mining techniques such as frequent and sequential patterns can be used. For example, in the work by [Alvares et al. \(2007\)](#), trajectories are first preprocessed to transform them into stop and moves, which is essentially a relation representation as stated in [Spaccapietra et al. \(2008\)](#). Then, standard data mining techniques are applied. This simple but clever technique allows the user to extract trajectory patterns that are purely semantic and that cannot be found with classical spatio-temporal data mining, based on the geometry of the trajectories. This has been the first approach to facing the problem of mining semantic trajectories.

8

Visual Analytics of Movement: A Rich Palette of Techniques to Enable Understanding

Natalia Andrienko and Gennady Andrienko

8.1 Introduction

Visual analytics develops knowledge, methods, and technologies that exploit and combine the strengths of human and electronic data processing (Keim et al., 2008). Technically, visual analytics combines interactive visual techniques with algorithms for computational data analysis. The key role of the visual techniques is to enable and promote *human understanding* of the data and *human reasoning* about the data, which are necessary, in particular, for choosing appropriate computational methods and steering their work. Visual analytics approaches are applied to data and problems for which there are (yet) no purely automatic methods. By enabling human understanding, reasoning, and use of prior knowledge and experiences, visual analytics can help the analyst to find suitable methods for data analysis and problem solving, which, possibly, can later be fully or partly automated. In this way, visual analytics can drive the development and adaptation of computational analysis and learning algorithms.

Visualization is particularly essential for analyzing phenomena and processes unfolding in geographical space. Since the heterogeneity of the space and the variety of properties and relationships occurring in it cannot be adequately represented for fully automatic processing, exploration and analysis of geospatial data and the derivation of knowledge from it needs to rely upon the human analyst's sense of the space and place, tacit knowledge of their inherent properties and relationships, and space/place-related experiences. This applies, among others, to movement data.

To support understanding and analysis of movement, visual analytics researchers leverage the legacy of cartography, with its established techniques for representing movements of tribes, armies, explorers, hurricanes, and so on; time geography (a branch of human geography), with its revolutionary idea of

considering space and time as dimensions of a unified continuum (space-time cube) and representation of behaviors of individuals as paths in this continuum; information visualization, with its techniques for user-display interaction supporting exploratory data analysis; and geovisualization, with its interactive maps and associated methods enabling exploration of spatial information.

This chapter gives a glimpse of the variety of the existing visual analytics methods for analyzing movement data. We group the methods into four categories according to the analysis focus:

1. Looking at trajectories: The focus is on trajectories of moving objects considered as wholes. The methods support exploration of the spatial and temporal properties of individual trajectories and comparison of several or multiple trajectories.
2. Looking inside trajectories: The focus is on variation of movement characteristics along trajectories. Trajectories are considered at the level of segments and points. The methods support detecting and locating segments with particular movement characteristics and sequences of segments representing particular local patterns of individual movement.
3. Bird's-eye view on movement: The focus is on the distribution of multiple movements in space and time. Individual movements are not of interest; generalization and aggregation are used to uncover overall spatio-temporal patterns.
4. Investigating movement in context: The focus is on relations and interactions between moving objects and the environment (context) in which they move, including various kinds of spatial, temporal, and spatio-temporal objects and phenomena. Movement data are analyzed together with other data describing the context. Computational techniques are used to detect occurrences of specific kinds of relations or interactions and visual methods support overall and detailed exploration of these occurrences.

We demonstrate the capabilities of visual analytics by examples using a data set consisting of GPS tracks of 17,241 cars collected during one week in Milan, Italy. The data were provided by Comune di Milano (Municipality of Milan).

8.2 Looking at Trajectories

In this section, we consider, first, the techniques for visual representation of trajectories and interaction with the representations; second, the use of clustering methods for comparative studies of multiple trajectories; and, third, the time transformations supporting exploration of temporal properties of trajectories and comparison of dynamic properties of multiple trajectories.

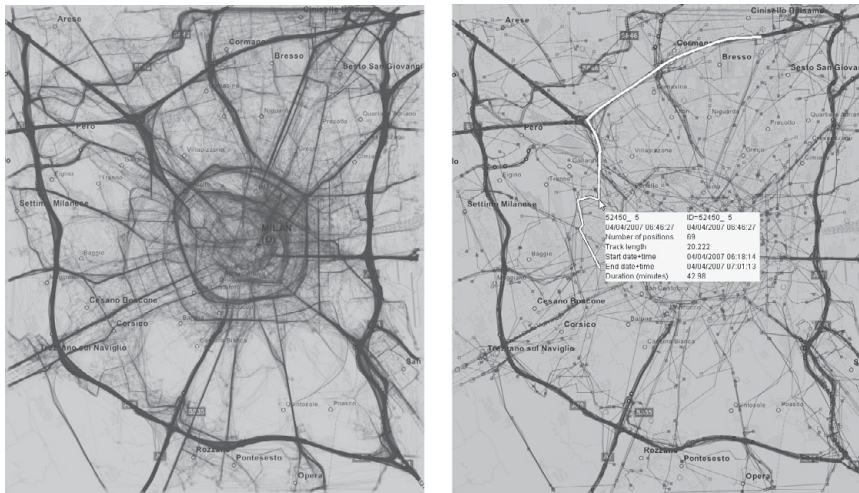
8.2.1 Visualizing Trajectories

The most common types of display for the visualization of movements of discrete entities are static and animated maps and interactive space-time cubes (STC). STC is a unified representation of space and time as a 3D cube in which two dimensions represent space and one dimension represents time. Spatio-temporal positions can be represented as points in an STC and trajectories as three-dimensional lines. When multiple trajectories are shown, the displays may suffer from visual clutter and occlusions. The drawback of STC, besides occlusion, is distortion of both space and time due to projection. It is also quite limited with respect to the length of the time interval that can be effectively explored. To compensate for these limitations, map and STC displays are often complemented with other types of graphs and diagrams.

Common interaction techniques facilitating visual exploration of trajectories and related data include manipulations of the view (zooming, shifting, rotation, changing the visibility and rendering order of different information layers, changing opacity levels, etc.), manipulations of the data representation (selection of attributes to represent and visual encoding of their values, for example, by coloring or line thickness), manipulations of the content (selection or filtering of the objects that will be shown), and interactions with display elements (e.g., access to detailed information by mouse pointing, highlighting, selection of objects to explore in other views, etc.). Multiple coexisting displays are visually linked by using consistent visual encodings (e.g., same colors) and exhibit coordinated behaviors by simultaneous consistent reaction to various user interactions.

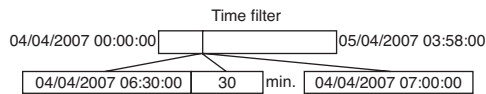
Figure 8.1 gives examples of map and STC displays and demonstrates some basic interaction techniques. The map in Figure 8.1a shows a subset of the Milan data set consisting of 8,206 trajectories that began on Wednesday, April 4, 2007. To make the map legible, the trajectory lines are drawn with only 5% opacity. A temporal filter, as in Figure 8.1c, can be used to limit the map view to showing only the positions and movements within a selected time interval. Thus, the display state in Figure 8.1b corresponds to the 30-minute time interval from 06:30 A.M. till 07:00 A.M. The time filter can also be used for map animation: the limiting time interval is moved (automatically or interactively) forward or backward in time, making the map and other displays dynamically update their content according to the current start and end of the interval.

Figure 8.1b also demonstrates the access to various attributes associated with a trajectory, such as start and end time, number of positions, length, and duration. When the mouse cursor points on a trajectory line, the attributes of this trajectory are shown in a pop-up window as well as the time when the car was in the position at the cursor.

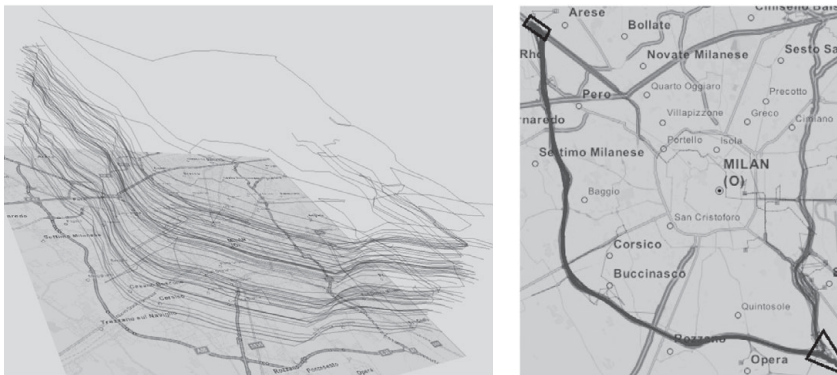


(a)

(b)



(c)



(d)

(e)

Figure 8.1 Visualization of trajectories: map and space-time cube. (a) 8,206 trajectories of cars are shown on a map as lines drawn with 5% opacity. (b) The map shows only positions and movements from a 30-minute time interval selected by means of a temporal filter (c). (d) A space-time cube (STC) shows a subset of trajectories selected by means of a spatial filter (e).

Figure 8.1d demonstrates the space-time cube (STC) display where two dimensions represent the space and the third dimension the time. The time axis is oriented from the bottom of the cube, where the base map is shown, to the top. When all trajectories are included in the STC, the view is illegible due to overplotting. In our example, the STC shows 63 trajectories selected by means of a spatial filter (Figure 8.1e). For the filter, we have outlined on the map two areas to the northwest and southeast of the city and set the filter so that only the trajectories that visited both areas in the given order are visible. There are also many other interactive techniques for data querying and filtering, for example, the ones suggested by [Bouvier and Oates \(2008\)](#) and [Guo et al. \(2011\)](#).

8.2.2 Clustering of Trajectories

Clustering is a popular technique used in visual analytics for handling large amounts of data. Clustering should not be considered as a standalone analysis method whose outcomes can be immediately used for whatever purposes. An essential part of the analysis is interpretation of the clusters by a human analyst; only in this way do they acquire meaning and value. To enable the interpretation, the results of clustering need to be appropriately presented to the analyst. Visual and interactive techniques play a key role here. Visual analytics usually does not invent new clustering methods but wraps existing ones in interactive visual interfaces supporting not only inspection and interpretation but often also interactive refinement of clustering results.

Trajectories of moving objects are quite complex spatio-temporal constructs. Their potentially relevant characteristics include the geometric shape of the path, its position in space, the life span, and the dynamics, that is, the way in which the spatial location, speed, direction and other point-related attributes of the movement change over time. Clustering of trajectories requires appropriate distance (dissimilarity) functions that can properly deal with these nontrivial properties. However, creating a single function accounting for all properties would not be reasonable. On the one hand, not all characteristics of trajectories may be simultaneously relevant in practical analysis tasks. On the other hand, clusters produced by means of such a universal function would be very difficult to interpret.

A more reasonable approach is to give the analyst a set of relatively simple distance functions dealing with different properties of trajectories and provide the possibility to combine them in the process of analysis. The simplest and most intuitive way is to do the analysis in a sequence of steps. In each step, clustering with a single distance function is applied either to the whole set of trajectories or to one or more of the clusters obtained in the preceding steps. If the purpose and work principle of each distance function is clear to the analyst, the clusters obtained in each step are easy to interpret by tracking the history of their derivation. Step by step, the analyst progressively refines his or her

understanding of the data. New analytical questions arise as an outcome of the previous analysis and determine the further steps. The whole process is called “progressive clustering” (Rinzivillo et al., 2008).

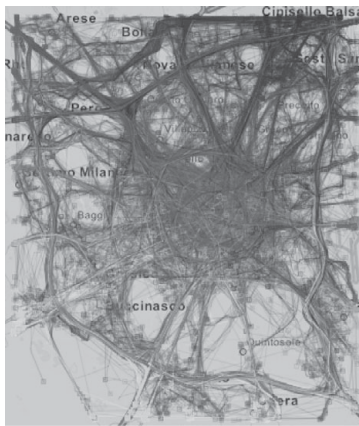
There is an implementation of the density-based clustering algorithm OPTICS in which the process of building clusters is separated from measuring the distances between the objects. This allows clustering with the use of diverse distance functions. Hence, the procedure of progressive clustering is done as follows: The user chooses a suitable distance function and applies the clustering tool first to the whole set of trajectories. Then the user interactively selects one or more clusters and applies the clustering algorithm to this subset using a different distance function or different parameter settings. The last step is iterated. In this way, the user may (1) refine clustering results, (2) combine several distance functions differing in semantics, and (3) gradually build comprehensive understanding of different aspects of the trajectories.

The procedure of progressive clustering is illustrated in Figure 8.2. The first image, Figure 8.2a, shows the result of clustering of the same subset of the car trajectories as in Figure 8.1 using the distance function “common destinations,” which compares the spatial positions of the ends of trajectories. From the 8,206 trajectories, 4,385 have been grouped into 80 density-based clusters and 3,821 treated as noise. Figure 8.2b shows the clusters without the noise. We have selected the biggest cluster, consisting of 590 trajectories that end in the northwest (Figure 8.2c), and applied clustering with the distance function “route similarity” to it. This distance function compares the routes followed by the moving objects. Figure 8.2d presents the 18 clusters we have obtained; the noise consisting of 171 trajectories is hidden. The largest cluster (in red) consists of 116 trajectories going from the city center and the next largest cluster (in orange) consists of 104 trajectories going from the northeast along the northern motorway. The orange cluster and the yellow cluster (68 trajectories) going from the southeast along the motorways to the south and west are, evidently, trajectories of transit cars. The clusters by route similarity are also shown in the STC in Figure 8.2e. This display involves time transformation, which is discussed in the next subsection.

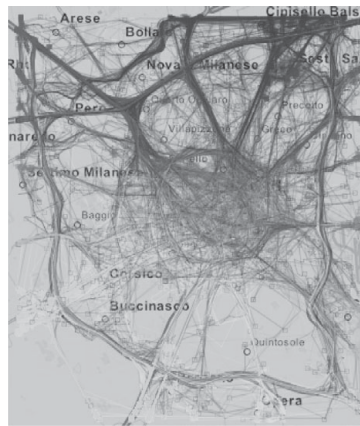
8.2.3 Transforming Times in Trajectories

Comparison of dynamic properties of trajectories using STC, time graph, or other temporal displays is difficult when the trajectories are distant in time, because their representations are located far from each other in a display. This problem can be solved or alleviated by transforming times in trajectories. Two classes of time transformations are possible:

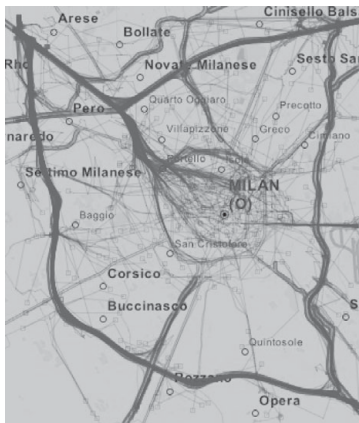
1. Transformations based on temporal cycles: Depending on the data and application, trajectories can be projected in time onto a single year, season, month,



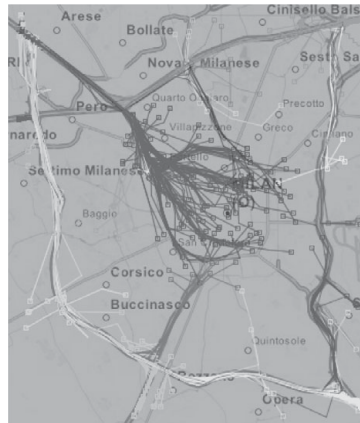
(a)



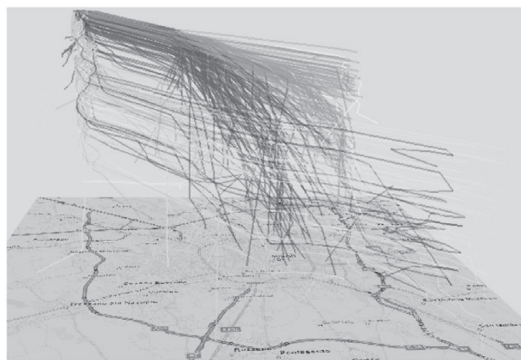
(b)



(c)



(d)



(e)

Figure 8.2 Interactive progressive clustering of trajectories. (a) The car trajectories have been clustered according to the destinations. (b) The noise is hidden. (c) One of the clusters is selected. (d) Clustering by route similarity has been applied to the selected cluster; the noise is hidden. (e) The clusters by route similarity are shown in an STC. (See color plate.)

week, or day. This allows the user to uncover and study movement patterns related to temporal cycles, for example, find typical routes taken in the morning and see their differences from the routes taken in the evening.

2. Transformations with respect to the individual lifelines of trajectories: Trajectories can be shifted in time to a common start time or a common end time. This facilitates the comparison of dynamic properties of the trajectories (particularly, spatially similar trajectories), for example, the dynamics of the speed. Aligning both the start and end times supports comparison of internal dynamics in trajectories irrespective of the average movement speed.

An example of time-transformed trajectories is shown in Figure 8.2e. The STC shows the route-based clusters of car trajectories ending in the northwest. The times in the trajectories have been transformed so that all trajectories have a common end time. This allows us to see that, although the routes within each cluster are similar, the dynamics of the movement may differ greatly. The speeds can be judged from the slopes of the lines. Fast movement is manifested by slightly inclined lines (which means more distance traveled in less time); steep lines signify slow movement. Vertical line segments mean staying in the same place. In the STC in Figure 8.2 we can very clearly observe the movement dynamics in the red cluster: the cars moved slowly while being in the city center but could move quickly after reaching the diagonal motorway. The orange cluster is divided in two parts. One part consists of nearly straight, slightly tilted lines indicating uniformly high speed along the whole route. The other part consists of trajectories with steep segments at the beginning. This means that there were times when the movement in the eastern part of the northern motorway was obstructed and the cars could not reach high speed. We can interactively select the trajectories with the steep segments and find out the times of the obstructed traffic: from about 06:00 A.M. till 01:00 P.M.; the most difficult situation was after 10:30 A.M. Making such observations could hardly be possible with the trajectories positioned in the STC according to their original times.

8.3 Looking inside Trajectories: Attributes, Events, and Patterns

The methods described in the previous section deal with trajectories as wholes, that is, treat them as atomic objects. Here we consider methods operating on the level of points and segments of trajectories. They visualize and analyze the variation of movement characteristics (speed, direction, etc.) and other dynamic attributes associated with trajectory positions or segments. The most obvious way to visualize position-related attributes is by dividing the lines or bands representing trajectories on a map or in a 3D display into segments and varying the appearance of these segments. Attribute values are usually represented by colouring or shading of the segments.

Position-related dynamic attributes can also be visualized in separate temporal displays such as a time graph or a time bars display. An example of a time bars display is given in Figure 8.3a. The horizontal axis represents time. Each trajectory is represented by a horizontal bar such that its horizontal position and length correspond to the start time and duration of the trajectory. Note that temporal zooming has been applied: a selected interval from 06:30 A.M. till 12:00 P.M. is stretched to the full available width. The vertical dimension is used to arrange the bars, which can be sorted based on one or more attributes of the trajectories (start time in our example). Coloring of bar segments encodes values of some user-selected dynamic attribute associated with the positions in the trajectories. This may be an existing (measured) attribute or an attribute derived from the position records, that is, coordinates and times. Examples of such derivable attributes are speed, acceleration, and direction. To represent attribute values by colors, the value range is divided into intervals and each interval is assigned a distinct color or shade. In Figure 8.3a, shades of red and green represent speed values; red is used for low speeds and green for high. The legend on the left explains the color coding. Interactive linking between displays allows the user to relate attribute values to the spatial context: when the mouse cursor points on some element within the time bars display, the corresponding spatial position is marked in the map by crossing horizontal and vertical lines and the trajectory containing it is highlighted (Figure 8.3b). In this example we see that the car whose trajectory is highlighted moved at 06:54 A.M. to the northeast with a speed of 1.2 km/h.

The use of this kind of dynamic link is limited to exploration of one or a few particular trajectories. To investigate position-related dynamic attributes in a large number of trajectories, the analyst can apply filtering of trajectory segments according to attribute values. Figure 8.3c–d illustrate how such filtering can be done in a highly interactive way. The color legend on the left of the time bars display is simultaneously a filtering device: the user can switch off and on the visibility of any value interval by clicking on the corresponding colored rectangle in the legend. In Figure 8.3c, the user has switched off all intervals except for that with speeds from 0 to 5 km/h. As a result, the trajectory segments with the speed values higher than 5 km/h have been hidden. The filter affects not only the time bars display but also the map (Figure 8.3d). It is possible to combine several segment filters based on values of different attributes.

The points satisfying filter conditions can be extracted from the trajectories into a separate data set (information layer) consisting of spatial events, that is, objects located in space and time. This data set can be visualized and analyzed independently from the original trajectories or in combination with them. In Figure 8.3e, the yellow circles represent 19,339 spatial events constructed from the points of the car trajectories where the speeds did not exceed 5 km/h. The filtering of the trajectory segments has been canceled so that

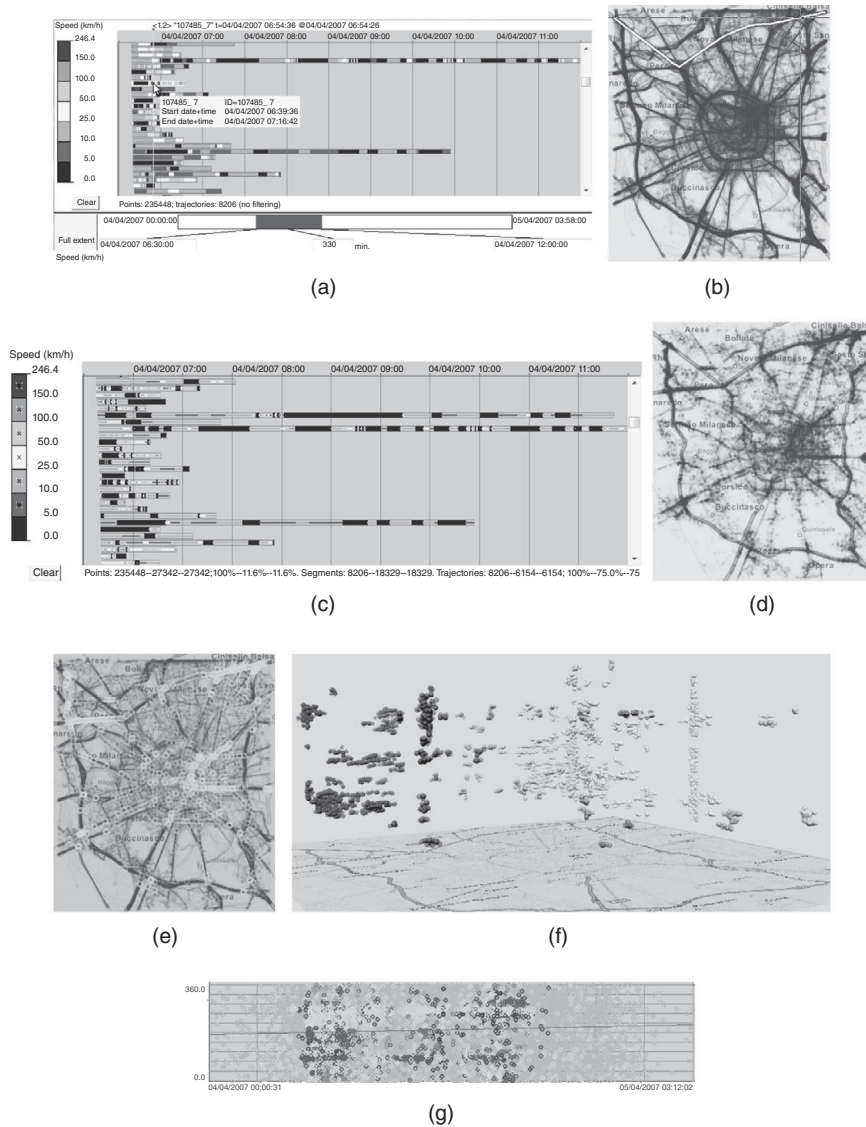


Figure 8.3 (a) A time bars display shows the speeds by color-coding. Mouse-pointing highlights the trajectory and marks the pointed position in a map (b). (c) Trajectory segments are filtered according to the speed values. (d) Only the segments satisfying the filter are visible on the map. (e) Low-speed events have been extracted from the trajectories according to the segment filter. (f) Density-based spatio-temporal clusters of the low speed events are shown in a space-time cube. (g) A scatterplot shows the times (horizontal dimension) and movement directions (vertical dimension) of the low-speed events. (See color plate.)

the whole trajectory lines are again visible. As could be expected, there are many low speed events in the center of the city. However, there are also visible concentrations of such events in many places on the motorways and their entrances/exits. These events are very probable to have occurred due to traffic congestions.

To investigate when and where traffic congestions occurred, we apply density-based clustering to the set of extracted events in order to find spatio-temporal clusters of low speed events. We look for dense spatio-temporal clusters because standalone low-speed events may be unrelated to traffic jams. The distance function we use is spatio-temporal distance between events. The STC in Figure 8.3f displays the clusters we have obtained; the noise (15,554 events) is hidden. The clusters are colored according to the geographical positions. We see a vertically extended cluster in light green on the east of the city. More precisely, it is located at the Linate airport. Most probably, the reason for these low-speed events is not traffic congestions but car parking or disembarking/embarking of passengers. The clusters in the other locations are more probable to be related to traffic jams. Some clusters on the northwest (blue) and northeast (cyan) are quite extended spatially, which means that the traffic was obstructed on long parts of the roads. The existence times of the clusters can be more conveniently seen in a 2D display, such as the scatterplot in Figure 8.3g, where the times of the events (horizontal axis) are plotted against the movement directions. It is possible to select the clusters one by one and see when they occurred and in which direction the cars were moving. For instance, two large clusters of slow movement westward occurred in the far northeast in the time intervals 05:38–06:50 and 10:20–12:44.

Generally, there are many possible ways in which events extracted from trajectories can be further analyzed and used. Interested readers are referred to papers by [Andrienko et al. \(2011b,c\)](#).

8.4 Bird's Eye on Movement: Generalization and Aggregation

Generalization and aggregation enable an overall view of the spatial and temporal distribution of multiple movements, which is hard to gain from displays showing individual trajectories. Besides, aggregation is helpful in dealing with large amounts of data. There are two major groups of analysis tasks supported by aggregation:

- Investigation of the presence of moving objects in different locations in space and the temporal variation of the presence.
- Investigation of the flows (aggregate movements) of moving objects between spatial locations and the temporal variation of the flows.

8.4.1 Analyzing Presence and Density

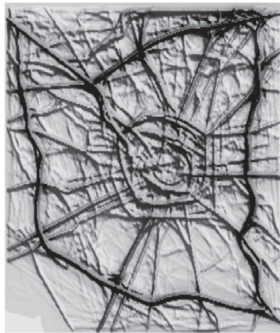
Presence of moving objects in a location during some time interval can be characterized in terms of the count of different objects that visited the location, the count of the visits (some objects might visit the location more than once), and the total time spent in the location. Besides, statistics of various attributes describing the objects, their movements, or their activities in the location may be of interest. To obtain these measures, movement data are aggregated spatially into continuous density surfaces or discrete grids. Density fields are visualized on a map using color coding and/or shading by means of an illumination model (Figure 8.4a). Density fields can be built using kernels with different radii and combined in one map to expose simultaneously large-scale patterns and fine features, as demonstrated in Figure 8.4a.

An example of spatial aggregation using a discrete grid is given in Figure 8.4b. The irregular grid has been built according to the spatial distribution of points from the car trajectories. The darkness of the shading of the grid cells is proportional to the total number of visits. Additionally, each cell contains a circle with the area proportional to the median duration of a visit. It can be observed that the median duration of staying in the cells with dense traffic (dark shading) is mostly low. Longer times are spent in the cells in the city center and especially at the Linate airport in the east. There are also places around the city where the traffic intensity is low while the visit durations are high.

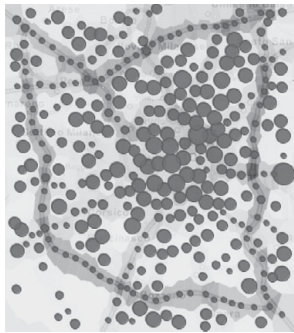
To investigate the temporal variation of object presence and related attributes across the space, spatial aggregation is combined with temporal aggregation, which can also be continuous or discrete. The idea of spatial density can be extended to spatio-temporal density: movement data can be aggregated into density volumes in a 3D space-time continuum, which can be represented in an STC.

For discrete temporal aggregation, time is divided into intervals. Depending on the application and analysis goals, the analyst may consider time as a line (i.e., linearly ordered set of moments) or as a cycle, for example, daily, weekly, or yearly. Accordingly, the time intervals for the aggregation are defined on the line or within the chosen cycle. The combination of discrete temporal aggregation with continuous spatial aggregation gives a sequence of density surfaces, one per time interval, which can be visualized by animated density maps. It is also possible to compute differences between two surfaces and visualize them on a map, to see changes occurring over time (this technique is known as a change map).

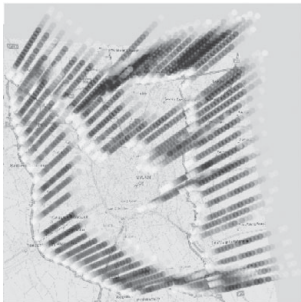
The combination of discrete temporal aggregation with discrete spatial aggregation produces one or more aggregate attribute values for each combination of space compartment (e.g., grid cell) and time interval. In other words, each space compartment receives one or more time series of aggregate attribute values.



(a)



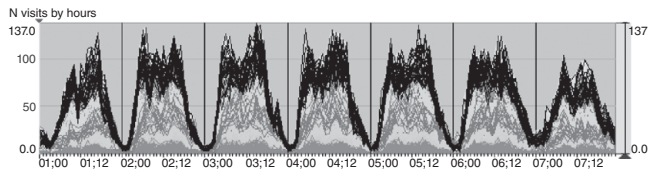
(b)



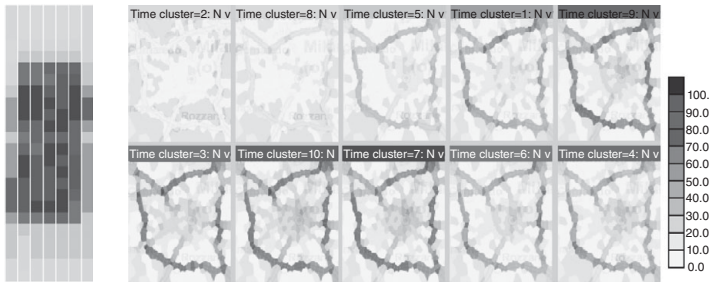
(c)



(d)



(e)



(f)

(g)

Figure 8.4 (a,b) Car tracks aggregated in a continuous density surface (a) and by discrete grid cells (b). (c) STC shows the variation of car presence over a day in the most visited cells. (d) The cells clustered by similarity of the presence time series shown on a time graph in (e). (f) Hourly time intervals clustered by similarity of the spatial distributions of car presence, which are summarized in (b). (See color plate.)

Visualization by animated density/presence maps and change maps is possible as in the case of continuous surfaces. There are also other possibilities. The time series may be shown in an STC by proportionally sized or shaded or colored symbols, which are vertically aligned above the locations; Figure 8.4c gives an example; the color legend is given in the lower right corner of Figure 8.4. Occlusion of symbols is often a serious problem in such a display; therefore, we have applied interactive filtering so that only the data for the most intensively visited cells (1,000 or more visits per day) are visible.

When the number of the space compartments is big and the time series is long, it may be difficult to explore the spatio-temporal distribution of object presence using only visual and interactive techniques. It is reasonable to cluster the compartments by similarity of the respective time series and analyze the temporal variation cluster-wise, that is, investigate the attribute dynamics within the clusters and do comparisons between clusters. Figure 8.4d demonstrates the outcome of k -means clustering of grid cells according to the time series of car presence obtained by aggregating the car movement data from the whole time period of one week by hourly intervals (hence, the time series consists of 168 time steps). Distinct colors have been assigned to the clusters and used for painting the cells on the map. The same colors are used for drawing the time series lines on the time graph in Figure 8.4e. The colours are chosen by projecting the cluster centroids onto a 2D continuous color map; hence, clusters with close centroids receive similar colors and, vice versa, high difference in colors signifies much dissimilarity between the clusters. Figure 8.4e shows a prominent periodic variation of car presence in the grid cells over the week. Interactive tools allow us to select the clusters one by one or pairs of clusters for comparison and see only these clusters on the displays. We find out that the clusters differ mainly in the value magnitudes and not in the temporal patterns of value variation, with the exception of the bright red and orange clusters. The value ranges in these clusters are very close. The main difference is that the red cluster has higher values in the afternoons of Sunday and Saturday. This may have something to do with people spending their leisure time near lakes, which are located to the north of the city.

Spatially referenced time series is one of two possible views on a result of discrete spatio-temporal aggregation. The other possibility is to consider the aggregates as a temporal sequence of *spatial situations*. The term “spatial situation” denotes spatial distribution of aggregate values of one or more attributes in one time interval. Thus, in our example, there are 168 spatial situations, each corresponding to one of the hourly intervals within the week. Temporal variation of spatial situations can also be investigated by means of clustering. In this case, the spatial situations are considered as feature vectors characterizing different time intervals. Clustering groups the time intervals by similarity of these feature vectors.

In Figure 8.4f, we have applied k -means clustering to the 168 spatial situations in terms of car presence and built a time mosaic display where each hourly interval is represented by a square. As in the previous case, different colors have been assigned to the clusters. The squares in the time mosaic are painted in these colors. The squares are arranged so that the columns, from left to right, correspond to the days, from Sunday (the first day in our data set) to Saturday, and the rows correspond to the hours of the day, from 0 on the top to 23 at the bottom. We see that the working days (Columns 2–6) have quite similar patterns of coloring, which means similarity of the daily variations of the situations. The patterns on Sunday (Column 1) and Saturday (Column 7) are different. The multimap display in Figure 8.4g shows summarized spatial situations: each small map represents the mean presence values in the respective time cluster (the color coding is the same as in the STC in Figure 8.4c; see the legend in the lower right corner). It is seen that the shades of cyan, which occur in the night hours, correspond to very low car presence over the city and the shades of red, which occur in the working days from 5 till 17 o'clock, to high presence, especially on the belt roads around the city. Red also occurs in the afternoon of Sunday (from 15 till 17) and in the morning of Saturday (from 8 till 9).

To deal with very large amounts of movement data, possibly not fitting in RAM, discrete spatio-temporal aggregation can be done within a database or data warehouse. The aggregates can then be loaded in RAM for visualization and interactive analysis.

8.4.2 Tracing Flows

In the previous section, we have considered spatial aggregation of movement data by locations (space compartments). Another method of spatial aggregation is by pairs of locations: for two locations A and B, the moves (transitions) from A to B are summarized. This can result in such aggregate attributes as number of transitions, number of different objects that moved from A to B, statistics of the speed, and transition duration. The term “flow” is often used to refer to aggregated movements between locations. The respective amount of movement, that is, count of moving objects or count of transitions, may be called “flow magnitude.”

There are two possible ways to aggregate trajectories into flows. Assuming that each trajectory represents a full trip of a moving object from some origin to some destination, the trajectories can be aggregated by origin-destination pairs, ignoring the intermediate locations. A well-known representation of the resulting aggregates is the *origin-destination matrix* (*OD matrix*) where the rows and columns correspond to the locations and the cells contain aggregate values. OD matrices are often represented graphically as matrices with shaded or colored cells. The rows and columns can be automatically or interactively reordered for

uncovering connectivity patterns such as clusters of strongly connected locations and “hubs,” that is, locations strongly connected to many others. A disadvantage of the matrix display is the lack of spatial context.

Another way to visualize flows is the flow map where flows are represented by straight or curved lines or arrows connecting locations; the flow magnitudes are represented by proportional widths and/or coloring or shading of the symbols. Since lines or arrows may connect not only neighboring locations but any two locations at any distance, massive intersections and occlusions of the symbols may occur, which makes the map illegible. Several approaches that have been suggested for reducing the display clutter either involve high information loss (e.g., due to filtering or low opacity of lesser flows) or work well only for special cases (e.g., for showing flows from one or two locations).

The other possible way of transforming trajectories to flows is to represent each trajectory as a sequence of transitions between all visited locations along the path and aggregate the transitions from all trajectories. Movement data having sufficiently fine temporal granularity or allowing interpolation between known positions may be aggregated so that only neighboring locations (adjacent spatial compartments) are linked by flows. Such flows can be represented on a flow map without intersections and occlusions of the flow symbols. To summarize movement data in this way, the space can be tessellated into larger or smaller compartments, for example, using the method suggested in [Andrienko and Andrienko \(2011\)](#), to achieve higher or lower degree of generalization and abstraction. This is illustrated in Figure 8.5a–c. The same trajectories of cars (a one-day subset from Wednesday) have been aggregated into flows using fine, medium, and coarse territory tessellations. The flows are represented by “half-arrow” symbols, to distinguish movements between the same locations in the opposite directions. Minor flows have been hidden to improve the display legibility; see the legends below the maps. The exact values of the flow magnitudes and other flow-related attributes can be accessed through mouse-pointing on the flow symbols. Flow maps can also be built using predefined locations or space partitioning, as demonstrated in Figure 8.5f, where the flow map is built based on a division of the territory of Milan into 13 geographic regions.

Flow maps can serve as expressive visual summaries of clusters of similar trajectories. To obtain such summaries, aggregation is applied separately to each cluster.

When movement data are aggregated into flows by time intervals, the result is time series of flow magnitudes. These can be visualized by animated flow maps or by combining flow maps with temporal displays such as a time graph. Flows may be clustered by similarity of the respective time series (Figure 8.5d,e) and the temporal variation analyzed clusterwise, as was suggested for time series of presence indicators in the previous section. Note that the spatial patterns visible on the map and the periodic patterns of flow variation visible on the time

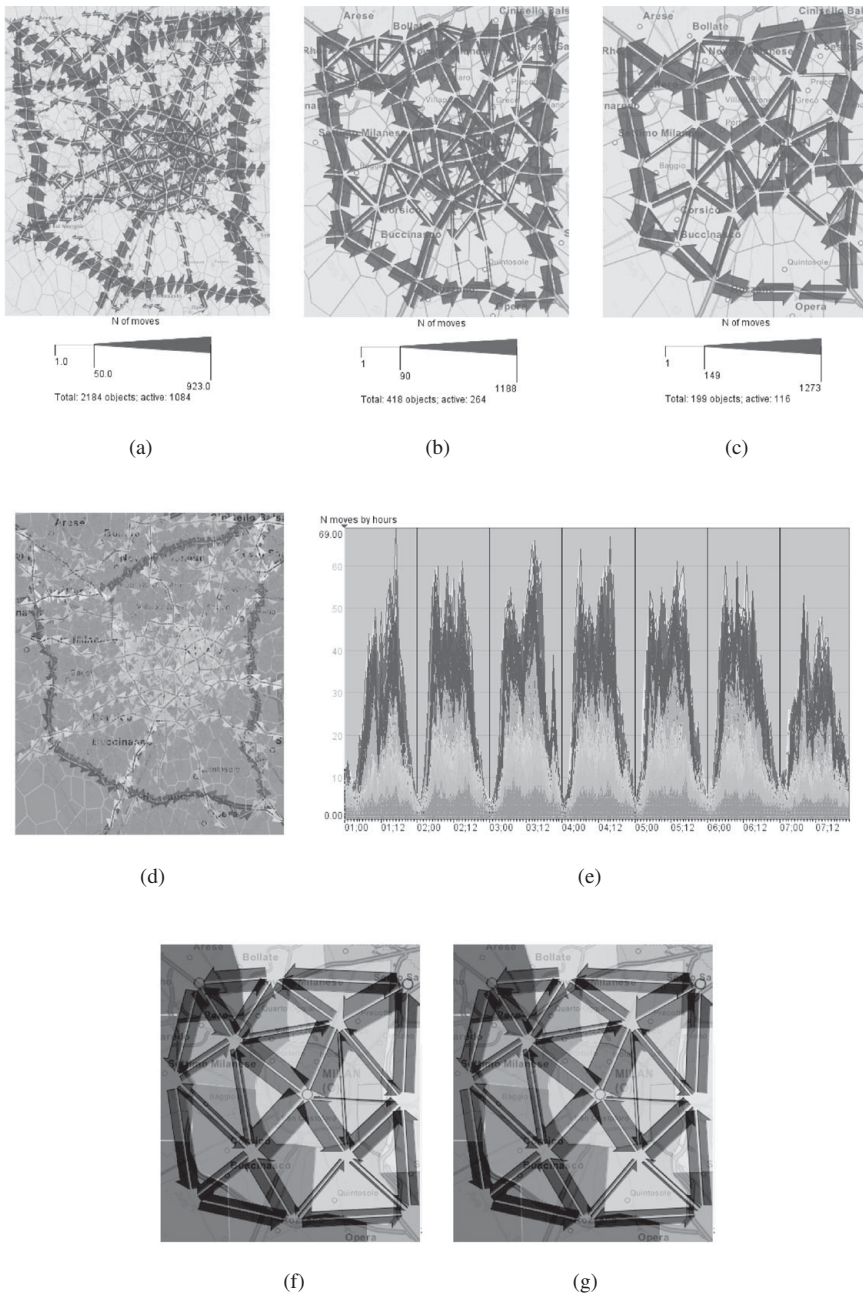


Figure 8.5 (a,b,c) Flow maps based on fine, medium, and coarse territory divisions obtained automatically. (d,e) Clustering of flows based on the time series of flow magnitudes. (f) Flows between predefined regions. (g) Investigation of movements between the regions over time adjusted to individual lifetimes of the trajectories. (See color plate.)

graph are similar to those that we observed for the presence (Figure 8.4d,e). However, we see that symmetric flows (i.e., flows between the same locations in opposite directions) may have different patterns of temporal variation. Thus, on the east and south of the city, symmetric flow symbols are coloured in blue and in magenta, that is, the respective time series belong to different clusters. The flows in the magenta cluster achieve higher magnitudes in the afternoons of all days, except Friday (day 6).

Aggregation of movement data into transitions between locations does not allow investigation of paths and movement behaviors where more than two locations are visited. The visualization technique demonstrated in Figure 8.5g aggregates trajectories in such a way that movement behaviors can be traced (Bremm et al., 2011). This is an abstract display where the horizontal axis represents time and colors represent different locations. The map in Figure 8.5f shows the geographic regions of Milan filled in different colors. The same colors are used in Figure 8.5g.

In this example, we investigate the movements of 4,634 cars that spent at least 6 hours on the territory under study on Wednesday (i.e., we have selected the trajectories with a duration of at least 6 hours); the flow map in Figure 8.5f summarizes the movements of these cars. The trajectories have been aligned in time to common start and end times, as mentioned in Section 8.2.3. The resulting time units are thousandths (also called “per mill”) of the total trajectory duration. Then the transformed time has been divided into 50 intervals of the length 20 per mills, or 2 percent. The temporal display in Figure 8.5g represents time intervals by vertical bars divided into colored segments proportionally to the number of cars that visited the regions in these intervals. Aggregated transitions between the regions are represented by bands drawn between the bars. The widths of the bands are proportional to the counts of the objects that moved. Gradient coloring is applied to the bands so that the left end is painted in the color of the origin location and the right end in the color of the destination location.

The colored bars are shown not for all time intervals but for a subset of intervals selected interactively or automatically. In our example, we have selected the first 3 intervals, the last 3 intervals, and each 10th interval (i.e., 100 per mills, 200 per mills, and so on). The small rectangles at the bottom of the display represent all time intervals. The greyscale shading encodes the amount of change in each interval with respect to the previous interval, that is, how many objects moved to different locations. We can observe that the most intensive movements of the selected cars occurred in the first 2 percent and in the last 2 percent of the total trajectory lifetime. Between the time intervals 100 and 900 the cars mostly stayed in the same regions. The most visited region was center. There were higher presence and more movements in the northern part of the city than in the southern part. The most intensive flows at the beginning of the trips were to the center and inner northeast and at the end to the outer northeast.

By interacting with the display, it is possible to explore not only direct transitions between locations but also longer sequences of visited locations. When the user clicks on a bar segment, the movements of the corresponding subset of objects are highlighted in the display (i.e., shown by brighter colors). It is possible to see which locations were visited and when. Thus, we can learn that from the 994 cars that were in the center in the interval 500 (i.e., in the middle of the trip time) 489 cars were in this region during the whole time and the remaining cars came to the center mainly from the northeast (133), southwest (132), northwest (74) and southeast (62) in the first 2 percent of the time. At the end, these cars moved back. Analogously, the user can click on bands connecting segments to select the objects participating in the respective transitions and trace their movements.

8.5 Investigation of Movement in Context

The spatio-temporal context of the movement includes the properties of different locations (e.g., land cover or road type) and different times (e.g., day or night, working day or weekend) and various spatial, temporal, and spatio-temporal objects affecting and/or being affected by the movement. The methods discussed so far seem to deal with movement data alone and not address the context of the movement, at least in an explicit way. However, the context is always involved in the process of interpreting what is seen on visual displays. Thus, the analyst always tries to relate visible spatial patterns to the spatial context (e.g., the highest car traffic density is on motorways) and visible temporal patterns to the temporal context (e.g., the traffic decreases on weekends).

The cartographic map is a very important provider of information about spatial context; therefore, maps are essential in analyzing movement data. It is not very usual, although it is possible, to include information about temporal context in temporal displays such as a time graph. A space-time cube may show spatio-temporal context, but occlusions and projection effects often complicate the analysis. Besides the context items that are explicitly represented on visual displays, the analyst also takes relevant context information from his/her background knowledge. Visual displays, especially maps, help the analyst in doing this since things that are shown can facilitate recall of related things from the analyst's mind. After noticing a probable relationship between an observed pattern and some context item, group of items, or type of items, the analyst may wish to check it, which can be supported by interactive visual tools.

The analyst may not only attend to the movement context for interpreting results of previously done analysis. It may also be a primary goal of analysis to detect and investigate particular relationships between the movement and a certain specific context item or group of items. For example, the goal may be to investigate how cars move on motorways or in traffic congestions. To do the

analysis, one may need special techniques that support focusing on the context items and relationships of interest.

Position records in movement data may include some context information, but this is rarely the case. In any case, movement data cannot include all possible context information. Typically, the source of relevant context information is one or more additional data sets describing some aspect(s) of the movement context. We shall shortly call such data “contextual data.” Context data may result from previous analyses of movement data. In our previous examples we have demonstrated derivation of spatial events, event clusters, as well as classes (clusters) of locations and of time moments. Such derived data can be considered as context data and used in further analysis of movement data.

The general approach is to derive contextual attributes for trajectory positions by joint processing of movement data and contextual data and then visualize the attributes to observe patterns and determine relationships. The derived attributes may characterize the environment (such as weather conditions) at the positions of the moving objects or relations (such as spatial distance) between the positions and context items in focus. Values of these attributes are defined, as a rule, for all trajectory positions. The analyst looks for correlations, dependencies, or, more generally, stable or frequent correspondences between the contextual attributes and movement attributes.

Besides stable relationships between movement and its context, the analyst may also be interested in transitory spatial, temporal, and spatio-temporal relationships occurring between moving objects and context items during the movement and lasting for limited time. This includes, in particular, relative movements of two or more moving objects such as approaching, meeting, passing, and following, and relative movements with respect to other kinds of spatial objects. Such occurrent relationships can be regarded as spatial events since they exist only at certain positions in space and in time.

Many types of relationships can be expressed in terms of spatial and/or temporal distances. This includes proximity between moving objects, visiting of certain locations or types of locations, and being in the spatio-temporal neighborhood of a spatial event. Spatial and/or temporal distances from moving objects to context items can be computed and attached to trajectory positions as new attributes, which can be visualized and/or used in further analyses. Particularly, they can be used for filtering and event extraction as described in Section 8.3.

As an example of analyzing movement in context, we shall investigate how the speed of car movement on motorways is related to the distances between the cars. Hence, there are two aspects of the movement context in which we are interested: type of location (specifically, motorway) and other cars (specifically, distances to them). The distances between the cars can be determined directly from the trajectory data; no additional data are needed. This can be done using

a computational procedure that finds for each trajectory position the closest position in another trajectory within a given time window, for example, of 1 minute length (from -30 to $+30$ seconds with respect to the time of the current position).

The location types could be taken from an additional data set describing the streets; however, we have no such data set for Milan. We shall demonstrate the use of previously derived data. Earlier we made a tessellation of the territory (Figure 8.4); moreover, the clustering according to the temporal variation of the car presence (Figure 8.4d) separates quite well the cells on motorways from the other cells. We create a suitable classification of the cells, as in Figure 8.6d, by editing the clusters. Here the yellow filling corresponds to the cells on motorways. We select this class of cells and compute the distances from the trajectory positions to the selected cells; for each position the nearest cell is taken. The computed distances are attached to the position records as a new attribute, which can now be used for filtering. By filtering, we extract the points and segments of the trajectories with zero distances to the selected cells (Figure 8.6d).

We compute also the distance from each position to the nearest position of another car within the 1-minute time window. This makes one more attribute attached to the position records. Then we use an additional filter according to values of this attribute to sequentially select the trajectory points with the distances to the nearest neighbor in three different ranges: below 20 m, from 20 to 50 m, and over 50 m. For each subset of points, we produce a frequency histogram of the respective speeds. The histograms are shown in Figure 8.6a–c. They have the same height and bar width. The latter corresponds to a speed range of approximately 5 km/h. Hence, despite the differing sizes of the point subsets, the shapes of the distributions can be compared. There are many points with low speeds (0–10 km/h) in each subset but the relative number of such points is the highest in the first subset and the lowest in the third subset. In all subsets, there is a smaller peak of frequencies for the speeds 80–90 km/h, but this peak is the lowest for the first subset and the highest for the third subset. Hence, we observe that smaller distances between cars on a motorway correspond to lower movement speeds.

To demonstrate investigation of occurrent relationships between moving objects and items of the context, we extract from the car trajectories the events where the car is on a motorway and its distance to the nearest neighbor car is at most 10 m while the movement speed is not more than 10 km/h. These events reflect occurrent proximity relationships of cars to motorways and other cars while the low speeds indicate that these occurrences may be related to traffic congestions. As we did in Section 8.3, we find spatio-temporal clusters of these events; some of them are shown in the STC in Figure 8.6e. We build spatio-temporal convex hulls around the event clusters (the yellow shapes in Figure 8.6e). We assume that each convex hull represents a traffic jam. Hence,

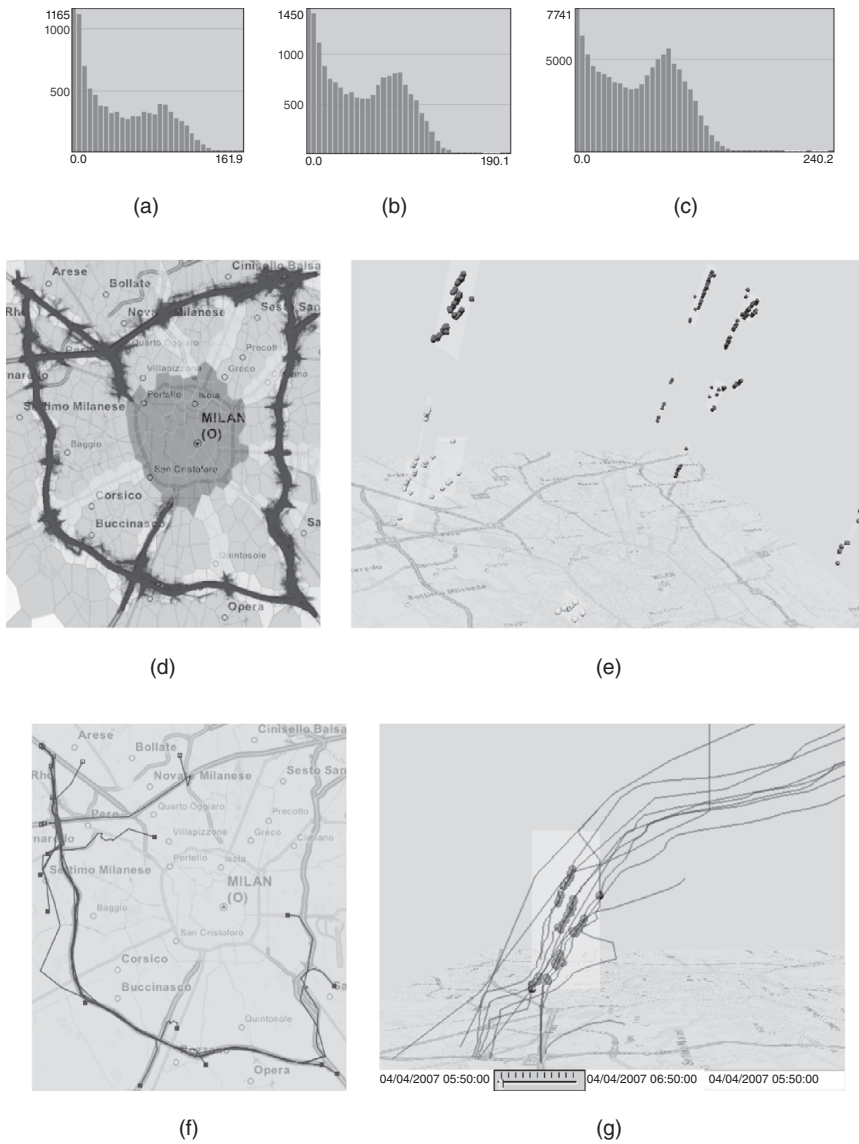


Figure 8.6 (a,b,c) Frequency distributions of car speeds on motorways in different ranges of distance to the nearest neighbor car: (a) below 20 m, (b) 20–50 m, (c) over 50 m. (d) Trajectory segments on or near motorways selected by means of segment filter. (e) Spatio-temporal clusters of low-speed events on motorways where the distance to the nearest neighbor is 10 m or less. Yellow shapes represent spatio-temporal convex hulls of the clusters. (f) Trajectories that passed through one of the convex hulls are selected by filtering. (g) The selected trajectories and respective low speed events in a STC. (See color plate.)

we have obtained an additional data set with spatio-temporal boundaries of traffic jams on motorways. It may, in turn, be considered as contextual data and used in further analysis. Thus, Figure 8.6f shows selected trajectories passing through one of the traffic jams, which have been used as a filter for trajectory selection. We can closely investigate the movement of the cars affected by this traffic jam by means of an STC (Figure 8.6g).

Sections 8.2–8.4 show that movement can be analyzed at different levels: whole trajectories, elements of trajectories (points and segments), and high-level summaries (densities, flows, etc.). In principle, analyzing movement in context can also be done at these levels. A comprehensive set of visual analytics methods addressing all these levels and different types of context items does not exist yet, which necessitates further research in this direction.

8.6 Conclusions

Movement data link together space, time, and objects positioned in space and time. They hold valuable and multifaceted information about moving objects and properties of space and time, as well as events and processes occurring in space and time. Visual analytics has developed a wide variety of methods and tools for analysis of movement data, which allow an analyst to look at the data from different perspectives and perform diverse analytical tasks. Visual displays and interactive techniques are often combined with computational processing, which, in particular, allows analysis of larger amounts of data than would be possible with purely visual methods. Visual analytics leverages methods and tools developed in other areas related to data analytics, particularly statistics, machine learning, and geographic information science. The main goal of visual analytics is to enable human understanding and reasoning. We have demonstrated by examples how understanding of various aspects of movement is gained by viewing visual displays and interacting with them, possibly after appropriate data transformations and/or computational derivation of additional data.

8.7 Bibliographic Notes

Keim et al. (2008) give a general definition of visual analytics and describe the scope of this research field. Andrienko et al. (2011a) suggest a conceptual framework defining the concepts of movement data, trajectories, and events, and possible relationships between moving objects, locations, and times. It shows that movement data hold valuable information not only about the moving objects but also about properties of space and time and about events and processes occurring in space and time. To uncover various types of information hidden in movement data, it is necessary to consider the data from different perspectives and to perform a variety of analytical tasks. The paper defines the possible foci

and tasks in analyzing movement data. Furthermore, it defines generic classes of analytical techniques and links the types of tasks to the classes of techniques that can support fulfilling them. The techniques include visualizations, data transformations, and computational analysis methods developed in several areas: visualization and visual analytics, geographic information science, database research, and data mining.

Readers interested in visualization of trajectories and techniques for interaction with the displays can be referred to the papers by [Kapler and Wright \(2005\)](#) describing a nice implementation of the space-time cube, [Bouvier and Oates \(2008\)](#) suggesting original interaction techniques for marking moving objects on an animated display and tracing their movements, and [Guo et al. \(2011\)](#) showing the use of several coordinated displays and interactive query techniques specifically designed for trajectories, such as sketching for finding trajectories with particular shapes.

[Rinzivillo et al. \(2008\)](#) talk about visually supported progressive clustering of trajectories. The paper argues for the use of diverse distance functions addressing different properties of trajectories, describes several distance functions, and demonstrates the use of progressive clustering by example.

[Andrienko et al. \(2011b,c\)](#) refer to “looking inside trajectories” (Section 8.3). The first paper describes visual displays that show temporal variation of dynamic attributes associated with trajectory positions. The second paper gives a structured list of position-related attributes that can be computationally derived from movement data alone and from a combination of movement data and contextual data. These attributes characterize either the movement itself or possible relationships between the moving objects and the movement context. Both papers deal with extraction of spatial events from movement data. The first paper introduces a conceptual model where movement is considered as a composition of spatial events of diverse types and extents in space and time. Spatial and temporal relations occur between movement events and elements of the spatial and temporal contexts. The model gives a ground to a generic approach based on extraction of interesting events from trajectories and treating the events as independent objects. The paper also describes interactive techniques for extracting events from trajectories. The second paper focuses more on the use of extracted events in further analysis. Thus, it considers density-based clustering of movement-related events, which accounts for their positions in space and time, movement directions, and, possibly, other attributes. The clustering allows extraction of meaningful places. The further analysis involves spatio-temporal aggregation of events or trajectories using the extracted places.

[Andrienko and Andrienko \(2010\)](#) give an illustrated survey of the aggregation methods used for movement data and the visualization techniques applicable to the results of the aggregation. These methods and techniques are also presented in a more formal way by [Andrienko et al. \(2011a\)](#). [Willems et al. \(2009\)](#)

describe aggregation of trajectories into a continuous density surface using a specially designed kernel density estimation method, which involves interpolation between consecutive trajectory points taking into account the speed and acceleration. Density fields built using kernels with different radii can be combined into one field to expose simultaneously large-scale patterns and fine features. [Andrienko and Andrienko \(2011\)](#) suggest a method for the tessellation of a territory used for discrete spatial aggregation of movement data and generation of expressive visual summaries in the form of flow maps. The method divides a territory into convex polygons of desired size on the basis of the spatial distribution of characteristic points extracted from trajectories. It uses a special algorithm for spatial clustering of points that produces clusters of user-specified spatial extent (radius). Depending on the chosen radius, the data can be aggregated at different spatial scales for achieving lower or higher degree of generalization and abstraction.

An example of visualization of flows between locations in the form of an origin-destination matrix can be found in [Guo \(2007\)](#). The rows and columns can be automatically or interactively reordered for uncovering connectivity patterns such as clusters of strongly connected locations and “hubs,” that is, locations strongly connected to many others.

To deal with very large amounts of movement data, possibly not fitting in RAM, discrete spatio-temporal aggregation can be done within a database or a data warehouse as described by [Raffaetà et al. \(2011\)](#). Only aggregated data are loaded in RAM for visualization and interactive analysis. Using roll-up and drill-down operators of the warehouse, the analyst may vary the level of aggregation.

[Andrienko and Andrienko \(2012\)](#) give a comprehensive review and extensive bibliography of methods, tools, and procedures for visual analysis of movement data.

9

Mobility Data and Privacy

Fosca Giannotti, Anna Monreale, and Dino Pedreschi

9.1 Introduction

Mobility data represent an invaluable source of information that can be recorded thanks to mobile telecommunications and ubiquitous computing where the locations of mobile users are continuously sensed. However, the collection, storage, and sharing of these movement data sets raise serious privacy concerns. In fact, position data may reveal the mobility behavior of the people: where they are going, where they live, where they work, their religion and so on. All this information refers to the private personal sphere of a person and therefore the analysis of mobility data may potentially reveal many facets of his or her private life. As a consequence, these kinds of data have to be considered personal information to be protected against undesirable and unlawful disclosure.

In the specific case of mobility scenarios, there exist two major different contexts in which the location privacy problem has to be taken into consideration: online location-based services and offline data analysis context. In the first case, a user communicates to a service provider his or her location to receive on-the-fly a specific service. An example of LBS is *find the closest point of interest (POI)*, where a POI could be a restaurant. Privacy issues in the context of online location-based services have been already addressed in Chapter 2. In the second case, large amounts of mobility data are collected and can be used for offline data mining analysis able to extract reliable knowledge useful to understand and manage intelligent transportation, urban planning, and sustainable mobility, as already highlighted in previous chapters.

Many PETs (privacy-enhancing technologies) for mobility data have been proposed by the scientific community. The most representative methods are presented in Section 9.3 of the present chapter by highlighting how the privacy models initially proposed for relational databases (presented in Section 9.2), are extended to spatio-temporal data. A common point of view among all these

techniques is that, unfortunately, obtaining privacy protection is becoming more and more difficult because of the complex nature of movement data: it is easy to show that privacy cannot simply be accomplished by deidentification (i.e., by removing the direct identifiers contained in the data). As an example, consider the deidentified GPS trajectory of a user driving in a city for a specific period. Using simple analytical tools, capable of visualizing the trajectory with its geographical context, it is possible to infer important and sensitive information about the user, such as the regions most commonly visited by the user. Moreover, analyzing the timeline with respect to the different regions it is possible to infer which region, among the most frequent locations, corresponds to the user's home since he or she usually stays there for the night, and the region corresponding to the work place, because he or she usually goes there every day at the same time, and stays there all the day. Clearly, by discovering the group of people living in the identified home and those working in that identified work place it is possible to identify the user as the person who belongs to both groups. This is possible checking publicly available information such as web pages.

In general, the data privacy problem requires finding an optimal trade-off between privacy and data utility. From one side, one would like to transform the data in order to avoid the reidentification of individuals and/or locations. Thus, one would like to publish safely the data for mining analysis or to communicate locations for receiving an online service without risks (or with negligible risk) for each data subject. From the other side, one would like to minimize the loss of information that can reduce the effectiveness of the underlying data when it is given as input to data mining methods and can cause bad quality of the received location-based service. Therefore, the goal is to maintain the utility of the data as much as possible. In order to measure the information loss introduced by the data transformation process it is necessary to define measures of utility; analogously, it is necessary to quantify the risks of privacy violation. Privacy by design, in the research field of privacy-preserving data analysis, is a recent paradigm that promises a quality leap in the conflict between data protection and data utility (Section 9.4). Recent applications of this paradigm for the design of privacy-preserving frameworks for movement data prove that it is possible to achieve reasonable and measurable privacy guarantees and a good quality of the analytical results.

9.2 Basic Concepts for Data Privacy

The analysis and disclosure of personal information to the general public or to third parties such as data miners is subject to the limitations imposed by the regulations for privacy protection. Nevertheless, if this information was rendered anonymous, these limitations would not apply, hence making it possible to share and analyze the information without explicit user agreement. In the last ten years,

Quasi-Identifier attributes			Sensitive attribute
Gender	Date of Birth	ZIP Code	Disease
F	1988	561*	Flu
F	1988	561*	Flu
F	1988	561*	Flu
M	1990	910*	Heart Disease
M	1990	910*	Cold
M	1990	910*	Flu

Figure 9.1 A 3-anonymous database.

different models have been proposed by the scientific community to achieve privacy protection while sharing and analyzing personal sensitive information. The most important privacy models are: k -anonymity, l -diversity, t -closeness, randomization, and cryptography-based models.

k -anonymity

The k -anonymity model was introduced in the context of relational databases, where data are stored in a table and each row of this table corresponds to one individual. The basic idea of the k -anonymity model is to guarantee that the information of every data subject cannot be distinguished from the information of other $k - 1$ data subjects. This model is based on the assumption of the existence of the following kind of attributes in the user's record: *identifiers*, which explicitly identify data owners, such as name and social security number (SSN); *quasi-identifiers*, which could identify data owners or a small groups of them (e.g., gender and zip code); *sensitive* attributes, which represent sensitive person-specific information (e.g., disease and salary) to be protected. Based on this classification, the privacy requirement defined by k -anonymity is that for each released record (e.g., a record is a row in the table in Figure 9.1) there must be at least other $k - 1$ records with the same quasi-identifier values. A set of records that have the same values for the quasi-identifiers is called *equivalence class*. The techniques adopted in the literature to enforce k -anonymity involve the removal of explicit identifiers and the generalization (e.g., date of birth is changed to the year of birth) or suppression (e.g., removing the date of birth), or microaggregation (clustering and averaging) of quasi-identifiers. It is evident that these techniques reduce the accuracy of the disclosed information.

l -diversity

The weakness of the k -anonymity model is that it can allow the disclosure of sensitive information. In other words, it only protects the identity of a user. Indeed, if a group of k records all have the same quasi-identifiers values and the same value of the sensitive attribute, it is not able to protect the sensitive information. As an example, consider the table in Figure 9.1. Suppose that the

adversary knows that Alice was born in 1988, lives in the area with ZIP code 56123 and is in the database. He knows that Alice's record is one of the first three in the table. Since all of those patients have the same medical condition (flu), the adversary can identify Alice's disease.

To overcome this weakness the *l-diversity* model requires obtaining groups of data subjects with indistinguishable quasi-identifiers and with an acceptable diversity of sensitive information. In particular, the main idea of this method is that every *k*-anonymous group should contain at least *l* different values for the attributes containing personal information.

***t*-closeness**

The problem with *l*-diversity is that it can be insufficient to prevent the disclosure of private information when the adversary knows the distribution of the private values. Indeed, if the adversary has prior belief about the private information of a data subject, he or she can compare this knowledge with the probability computed from the observation of the disclosed information. In order to avoid this weakness, the *t-closeness* model requires that, in any group of quasi-identifiers, the distribution of the values of a sensitive attribute be close to the distribution of the attribute values in the overall table. The distance between the two distributions should be no more than a threshold *t*. Clearly, this limits the information gain of the adversary after an attack.

Randomization

The *randomization* model is based on the idea of perturbing the data to be published by adding a noise quantity. More technically, this method can be described as follows. Denote by $X = \{x_1 \dots x_m\}$ the original data set. The new distorted data set, denoted by $Z = \{z_1 \dots z_m\}$, is obtained by drawing independently from the probability distribution a noise quantity n_i and adding it to each record $x_i \in X$. The set of noise components is denoted by $N = \{n_1, \dots, n_m\}$. The original record values cannot be easily guessed from the distorted data as the variance of the noise is assumed large enough. Instead, the distribution of the data set can be easily recovered.

Cryptography-Based Models

The basic idea of the privacy models based on cryptography techniques is to compute analytical results without sharing the data in such a way that anything is disclosed except the final result of the analysis. In general, the application of these models allows one to compute functions over inputs provided by multiple parties without sharing the inputs. This problem is addressed in cryptography in the field of secure multi-party computation. As an example, consider a function f of n arguments and n different parties. If each party has one of the n arguments a *protocol* is needed that allows exchanging information and computing

the function $f(x_1, \dots, x_n)$, without compromising privacy. There exist some methods that allow transforming data mining problems into secure multi-party computation problems. In the literature, many protocols have been proposed for the computation of the secure sum, the secure set union, the secure size of set intersection and the scalar product. These protocols can be used as data mining primitives for secure multiparty computation in case of horizontally and vertically partitioned data sets.

9.3 Privacy in Offline Mobility Data Analysis

In the context of offline mobility data analysis, large amounts of collected mobility data can be used for extracting reliable knowledge useful for the understanding of very complex and interesting phenomena. Indeed, these data can be used for various data analyses that allow improving systems for city traffic control, mobility management, and urban planning, as evidenced in Chapters 6, 7, and 10. Unfortunately, mobility data provide detailed movement information of individuals and thus this information could be used for their identification and sometimes for inferring personal sensitive information about them. Therefore, when spatio-temporal data have to be analyzed and/or published, it is fundamental to guarantee individual privacy protection of the respondents represented in the data.

The privacy models for relational data described in the previous section have been widely adopted to achieve privacy protection in the context of the offline analysis of spatio-temporal data. However, the different and more complex nature of mobility data with respect to relational tabular data sometimes rendered it difficult to apply these privacy models directly and this has led to the definition of some suitable variants. The inadequacy of the aforementioned models for trajectory data depends on the fact that these data pose new challenges due to the following characteristics: time dependency, location dependency, and data sparseness. The location and time components of the mobility data make it harder to enforce privacy protection. Indeed, both the information alone or in combination with external sources could be used by an attacker to reidentify individuals and discover sensitive information about them. As a consequence, a privacy defense has to take into consideration this fact and apply a data transformation able to eliminate the privacy threats that derive from the two sources of information. Moreover, the problem is made more difficult by the sparseness of this large amount of data. Indeed, usually an individual visits few locations with respect to the total number of locations available in the territory, therefore the trajectories are relatively short and it is difficult to find overlapping of locations among different trajectories, thus causing the sparseness problem. Additionally, the time component makes the situation more complicated because the same location can be visited by different individuals in different time periods.

All this makes mobility data very sparse and in this setting, it is clearly difficult to identify and to group together trajectories for enforcing, for example, traditional k -anonymity.

The next section shows how the basic data privacy notions presented in Section 9.2 have been adapted to address the new challenges posed by spatio-temporal data in offline data analysis. We present three categories of PETs: PETs for mobility data publishing, PETs for distributed mobility data mining, and PETs for knowledge hiding in mobility data.

9.3.1 PETs for Publishing of Trajectory Data

Mobility data publishing includes sharing the mobility data with specific recipients such as data miners and releasing the data for public download. In both cases, the recipients could potentially be adversaries who try to associate sensitive information in the published data with a known person. The privacy-preserving techniques for mobility data publishing have the goal to transform spatio-temporal data to make them anonymous; in other words, they provide suitable formal safeguards against reidentification of individuals represented in the data by their movements.

In the literature, most of the proposed PETs for mobility data publishing use privacy models that are suitable variants of the classical k -anonymity model. They consider adversaries that use location-based knowledge for the reidentification of users. As explained in Section 9.2, an adversary can use *quasi-identifier* attributes (e.g., age, gender, and ZIP code) representing public knowledge and can use them as key elements for the reidentification of individuals. Similarly, in spatio-temporal databases the attackers could identify the person corresponding to a given trajectory by using pairs of locations and timestamps that work as quasi-identifiers. In this context the challenge often is the definition of realistic and reasonable quasi-identifiers. Two important questions need to be answered when we have to consider quasi-identifiers in spatio-temporal databases: (1) *Can we assume the same set of quasi-identifiers for all the individuals in the database?* (2) *Where and how should the knowledge of quasi-identifiers be obtained?*

Concerning the first question, in the literature some works argue that, unlike in relational microdata, where every tuple has the same set of quasi-identifier attributes, in spatio-temporal data it is very likely that various individuals have different quasi-identifiers and clearly this fact should be taken into consideration in modeling adversary knowledge. Unfortunately, allowing different sets of quasi-identifiers for different individuals makes the anonymization problem more challenging because the anonymization groups may not be disjoint.

Concerning the second question typically we have different possibilities: (a) the quasi-identifiers may be part of the users' personalized settings; (b) they

may be provided directly by the users when they subscribe to the service; and c) the quasi-identifier may be found by statistical data analysis or data mining.

Given that in the real world the definition of quasi-identifiers in movement data is not trivial, most anonymization approaches do not use any information about the quasi-identifiers of trajectories during the anonymization process. In Section 9.3.1 we present the details of a typical technique of this category.

Anonymization without Quasi-Identifiers

A spatio-temporal technique that does not take into consideration any knowledge about the quasi-identifier of trajectories implicitly assumes that an adversary may identify a user in any location at any time. Clearly, this is a very conservative setting and under this assumption the anonymized data sets are composed of anonymization groups, each one containing at least k identical or very similar trajectories. This typically is achieved by the application of clustering-based approaches.

The application of classical k -anonymity notion in spatio-temporal data is hard because it is necessary to take into account some problems that are specific in this context. As an example, in the definition of the privacy model one should consider the inaccuracy of the positioning device that introduces possible location imprecision in the collection of data. This leads to the definition of a variant of the k -anonymity notion called (k, δ) -anonymity suitable for moving objects databases, where δ represents the possible location imprecision. This novel concept is based on colocalization that exploits the inherent uncertainty of the moving object's whereabouts. Intuitively, the trajectory is considered as a cylindrical volume with some uncertainty. In other words, the position of a moving object in the cylinder then becomes uncertain. Figure 9.2 illustrates a graphical representation of an uncertain trajectory.

Two trajectories moving within the same cylinder are indistinguishable; this leads to the definition of (k, δ) -anonymity model:

Definition 9.1. Given an anonymity threshold k and a radius parameter δ , a (k, δ) -anonymity set is a set of at least k trajectories that are colocalized with respect to δ . \square

A set of trajectories S , with $|S| \geq k$, is a (k, δ) -anonymity set if and only if there exists a trajectory t_c such that all the trajectories in S are possible motion curves of t_c within an uncertainty radius of $\frac{\delta}{2}$. Given a (k, δ) -anonymity set S , we obtain the trajectory t_c by taking, for each $t \in [t_1, t_n]$, the point (x, y) that represents the center of the minimum bounding circle of all the points at time t of all trajectories in S (Figure 9.3).

The (k, δ) -anonymity framework requires transforming a trajectory database D in D' in such a way that for each trajectory $t \in D'$ a (k, δ) -anonymity set

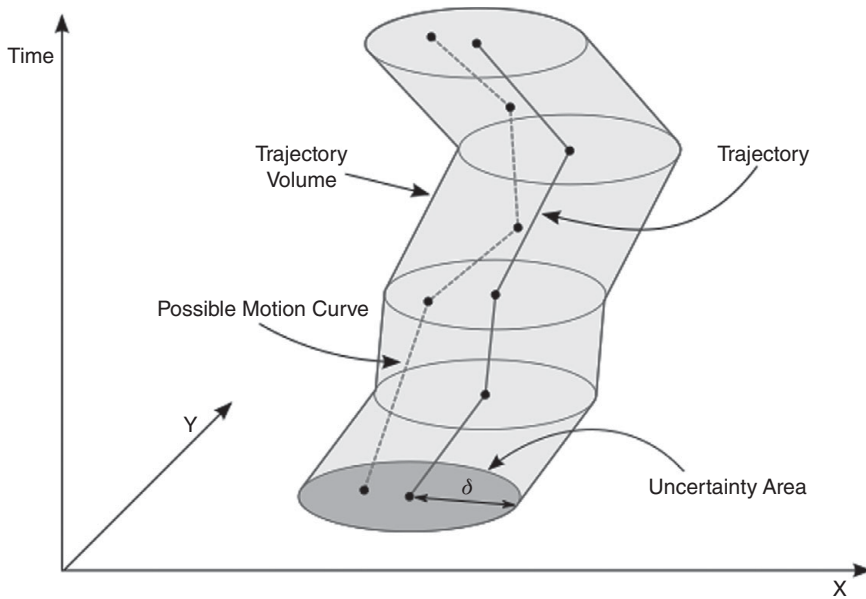


Figure 9.2 Uncertain trajectory: uncertainty area, trajectory volume, and possible motion curve.

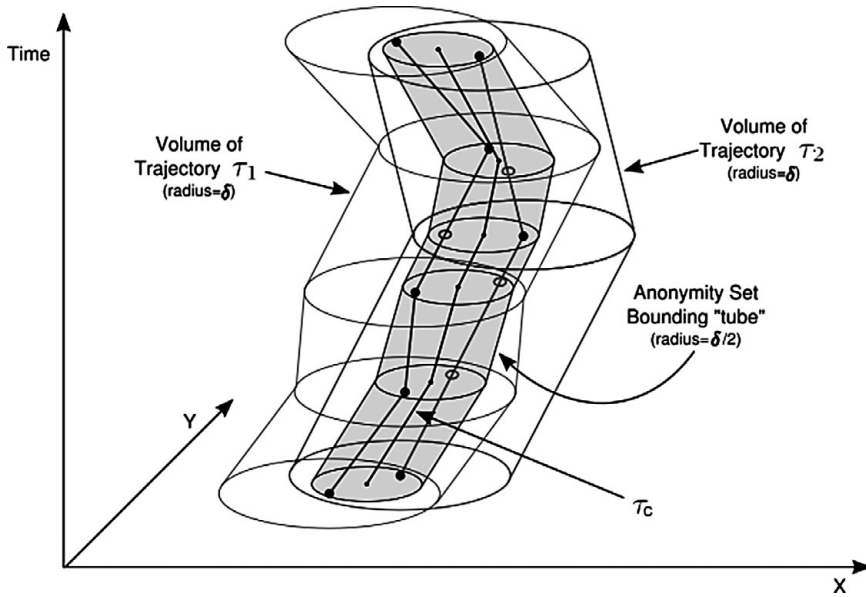


Figure 9.3 A $(2, \delta)$ -anonymity set formed by two co-localized trajectories, their respective uncertainty volumes, and the central cylindrical volume of radius $\frac{\delta}{2}$ that contains both trajectories.

$S \subset D'$ with $t \in S$ exists, and the distortion between D and D' is minimized. To achieve (k, δ) -anonymous data sets, we can apply a method based on trajectory clustering and spatial translation that is a form of perturbation. In particular, it consists of three main steps:

1. **Preprocessing step:** The goal of this phase is to find a partition of the original database in equivalence classes with respect to the time span. In other words, each equivalence class contains trajectories with the same starting time and ending time. This step is necessary because the algorithm has to compute the Euclidean distance between trajectories and when it is computed on the input raw data could lead to the generation of very small equivalence classes.
2. **Clustering step:** In this phase the trajectories, obtained by the preprocessing step, are clustered by using a greedy approach. This step iteratively selects a pivot trajectory as cluster center and assigns its nearest $k - 1$ trajectories to the cluster. The clusters must have a radius not larger than a given threshold to guarantee a certain compactness of the groups of trajectories. So, if this criterion of compactness is not satisfied then the process is repeated selecting a different pivot trajectory. Clearly, when a remaining trajectory cannot be added to any cluster without violating the compactness constraint, then it is trashed because it is considered as an outlier.
3. **Space transformation step:** The aim of this step is to transform each cluster into a (k, δ) -anonymity set. This is achieved perturbing each trajectory by the spatial translation that allows putting all the trajectories within a common uncertainty cylinder.

9.3.2 Other PETs for Offline Mobility Data Analysis

Although PETs for mobility data publishing represent an important part of the literature on privacy in mobility data analysis, there are other interesting techniques that consider different scenarios and different settings and apply different privacy models, such as techniques suitable for analyzing and mining data in distributed environments and techniques that allow hiding models considered sensitive in a database to be published.

Distributed Privacy-Preserving Mobility Data Mining

The methods belonging to this group aim at analyzing data sets that are partitioned and distributed among several parties that do not want to (or cannot) share the data or certain corporate information that is represented in the data, but are interested in developing global models of common interest. Therefore, the main assumption in this scenario is that multiple data holders want to collaboratively perform data mining on the union of their data without revealing their sensitive information. The question addressed in these cases is how to compute the results

without sharing the data, so that nothing is disclosed except the final result of the data mining process. This problem is addressed in cryptography in the field of *secure multi-party computation*. An example of a problem tackled by this kind of approach is privacy-preserving clustering in horizontally partitioned spatio-temporal data. Here, each horizontal partition contains trajectories of distinct moving objects collected by separate sites, which want to cluster these trajectories without releasing sensitive location information to the other data holders. At the end of the protocol the global clustering results will be available to each data holder. The method used to achieve this goal is to construct the dissimilarity matrix of the trajectories in a privacy preserving manner, which can be the input of any hierarchical clustering algorithm. In this setting there is a third party that has the following tasks: (1) managing the communication among data holders; (2) constructing a global dissimilarity matrix; (3) clustering the trajectories by using the dissimilarity matrix; and (4) releasing the final result to the data holders. Each party involved is considered semitrusted, in the sense that they follow the protocol as expected to, but cannot store any information to infer sensitive data. Moreover, parties do not share any sensitive information with each other.

As an example application of this technique, consider the case of a traffic control office that wants to solve traffic congestion by analyzing data from a mobile operator who cannot share these data with other entities for privacy issues. The traffic congestion problem assumes the use of a clustering algorithm, therefore the best solution is to apply a privacy-preserving clustering algorithm for horizontally partitioned data that avoids sharing of the spatio-temporal data.

Knowledge Hiding in Mobility Data

Knowledge hiding refers to the activity of hiding patterns considered sensitive in a database before being published. In fact, if the data are published as they are, the sensitive patterns may be surfaced by means of data mining techniques. Knowledge hiding involves a process of *sanitization* of the database in such a way that the sensitive knowledge can no longer be inferred, while the original database is changed as little as possible. This problem is particularly interesting in the context of spatio-temporal patterns in a database of trajectories. Mobility data contain the description of typical mobile behaviors (i.e., frequent patterns) that are considered sensitive for political or security reasons. It is therefore necessary to have a method capable of hiding such sensitive patterns before the disclosure of the database. A valid hiding technique in this context should take into consideration the road network, modeled as a directed graph, and therefore consider trajectories of objects moving over a background road network. A privacy solution should sanitize the input trajectory database D in such a way that a set of sensitive spatio-temporal patterns P is hidden while most of the information in D is maintained. The resulting database D' , which is the released version, is consistent with the background road network. The privacy

solution avoids creating unreal trajectories in the sanitization process, since the road network is publicly available knowledge and thus unreal trajectories can be easily identified. Moreover, all sensitive patterns are hidden in D' , that is, they have a support no more than the given disclosure threshold ψ . Finally, the last requirement is that D' is kept as similar as possible to D .

9.4 Privacy by Design in Data Mining

As shown in the previous sections, several techniques have been proposed by the scientific community to develop technological frameworks for countering the threats of undesirable and unlawful effects of privacy violation, without obstructing the knowledge discovery opportunities of data mining technologies. However, the common result obtained is that no general method exists that is capable of both dealing with “generic personal data” and preserving “generic analytical results.” The ideal solution would be to inscribe privacy protection into the knowledge discovery technology by design, so that the analysis incorporates the relevant privacy requirements from the very beginning. We evoke here the concept of “privacy by design,” coined in the 1990s by Ann Cavoukian, the Information and Privacy Commissioner of Ontario, Canada. In brief, privacy by design refers to the philosophy and approach of embedding privacy into the design, operation, and management of information-processing technologies and systems.

The articulation of the general “by design” principle in the data mining domain is that higher protection and quality can be better achieved in a goal-oriented approach. In such an approach, the data mining process is designed with assumptions about:

- The sensitive personal data that are the subject of the analysis;
- The attack model, that is, the knowledge and purpose of a malicious party that has an interest in discovering the sensitive data of certain individuals;
- The category of analytical queries that are to be answered with the data.

Under these assumptions, it is conceivable to design a privacy-preserving analytical process able to:

1. Transform the data into an anonymous version with a quantifiable privacy guarantee – that is, the probability that the malicious attack fails;
2. Guarantee that a category of analytical queries can be answered correctly, within a quantifiable approximation that specifies the data utility, using the transformed data instead of the original ones.

In the next sections we present two frameworks that offer two different instances of the privacy by design paradigm in the case of personal mobility trajectories (obtained from GPS devices or cell phones). The first one is suitable

for the privacy-aware publication of movement data enabling clustering analysis useful for the understanding of human mobility behavior in specific urban areas. The released trajectories are made anonymous by a suitable process that realizes a generalized version of the original trajectories. The second framework is suitable when it is required that the released data set of trajectories contains the real locations contained in the original data. In fact, this framework applies to the original data a process of transformation capable of maintaining unchanged this information, even if the data become anonymous.

The application of this methodology requires one to understand: the specific properties of the trajectories to be protected; which characteristics it is necessary to preserve to guarantee a good quality of the analyses that have to be performed on these data; and which adversary's knowledge the attacker may use for the user reidentification. Clearly, this information is fundamental for the design of a data transformation technique.

9.4.1 Trajectory Anonymization by Spatial Generalization

In this section, we show the design of a privacy-preserving framework for the publication of movement data, while preserving clustering analysis. The framework is based on a data-driven spatial generalization of the data set of trajectories. The results obtained with the application of this framework show how trajectories can be anonymized to a high level of protection against reidentification while preserving the possibility of mining clusters of trajectories, which enables novel powerful analytic services for infomobility or location-based services.

Attack Model

In this framework the *linkage attack model* is considered, that is, the ability to link the published data to external information, which enables some respondents associated with the data to be reidentified. In relational data, linking is made possible by *quasi-identifiers*, that is, attributes that, in combination, can uniquely identify individuals, such as birth date and gender (see Section 9.2). The remaining attributes represent the respondent's private information, which may be violated by the linkage attack. In privacy-preserving data publishing techniques, such as *k*-anonymity, the goal is precisely to find countermeasures to this attack, and to release person-specific data in such a way that the ability to link to other information using the quasi-identifier(s) is limited. In the case of spatio-temporal data, where each record is a temporal sequence of locations visited by a specific person, the dichotomy of attributes into quasi-identifiers (QI) and private information (PI) does not hold any longer: here, a (sub)trajectory can play both the role of QI and the role of PI. To see this point, consider that the attacker may know a sequence of places visited by some specific person *P*: for example, by shadowing *P* for some time, the attacker may learn that *P*

was in the shopping mall, then in the park, and then at the train station. The attacker could employ such knowledge to retrieve the complete trajectory of P in the released data set: this attempt would succeed, provided that the attacker knows that P 's trajectory is actually present in the data set, if the known trajectory is compatible with (i.e., is a subtrajectory of) just one trajectory in the data set. In this example of a linkage attack in the movement data domain, the subtrajectory known by the attacker serves as QI, while the entire trajectory is the PI that is disclosed after the reidentification of the respondent. Clearly, as the example suggests, is rather difficult to distinguish QI and PI: in principle, any specific location can be the theater of a shadowing action by a spy, and therefore any possible sequence of locations can be used as a QI, that is, as a means for reidentification. Put another way, distinguishing between QI and PI among the locations means putting artificial limits on the attacker's background knowledge; on the contrary, it is required in privacy and security research to have assumptions on the attacker's knowledge that are as liberal as possible, in order to achieve maximal protection.

As a consequence of this discussion, it is reasonable to consider the radical assumption that any (sub)trajectory that can be linked to a small number of individuals is a potentially dangerous QI and a potentially sensitive PI. Therefore, in the *trajectory linkage attack*, the malicious party M knows a subtrajectory of a respondent R (e.g., a sequence of locations where R has been spied on by M) and M would like to identify in the data the whole trajectory belonging to R , that is, learn all places visited by R .

Privacy-Preserving Techniques

How is it possible to guarantee that the probability of success of the above attack is very low while preserving the utility of the data for meaningful analyses? Consider the source trajectories represented in Figure 9.4, obtained from a massive data set of GPS traces (17,000 private vehicles tracked in the city of Milan, Italy, during a week).

Each trajectory is a deidentified sequence of timestamped locations, visited by one of the tracked vehicles. Albeit deidentified, each trajectory is essentially unique – very rarely are two different trajectories exactly the same given the extremely fine spatio-temporal resolution involved. As a consequence, the chances of success for the trajectory linkage attack are not low. If the attacker M knows a sufficiently long subsequence S of locations visited by the respondent R , it is possible that only a few trajectories in the data set match with S , possibly just one. Indeed, publishing raw trajectory data such as those depicted in Figure 9.4 is an unsafe practice, which runs a high risk of violating the private sphere of the tracked drivers (e.g., guessing the home place and the work place of most respondents is very easy). Now, assume that one wants to discover the trajectory clusters emerging from the data through data mining, that

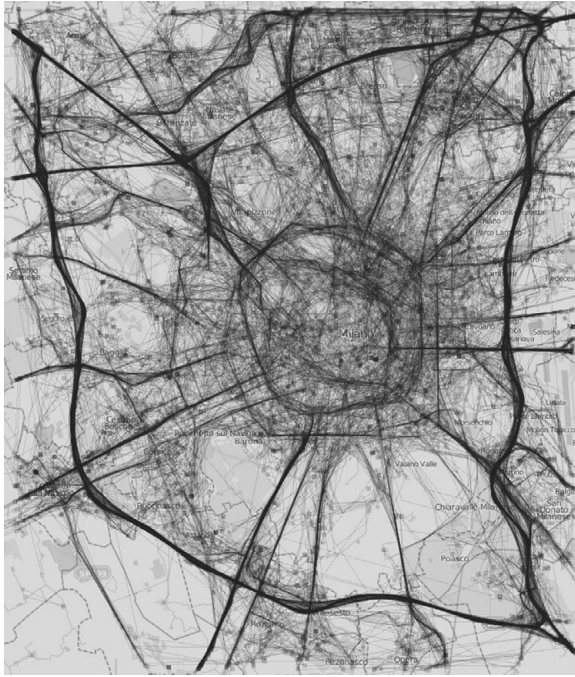


Figure 9.4 Milan GPS trajectories.

is, the groups of trajectories that share common mobility behavior, such as the commuters that follow similar routes in their home–work and work–home trips. An anonymizing transformation of the trajectories consists of the following steps:

1. Characteristic points are extracted from the original trajectories: starting points, ending points, points of significant turn, points of significant stop (Figure 9.5a);
2. Characteristic points are clustered into small groups by spatial proximity (Figure 9.5b);
3. The central points of the groups are used to partition the space by means of Voronoi tessellation (Figure 9.5c);
4. Each original trajectory is transformed into the sequence of Voronoi cells that it crosses (Figure 9.5d).

As a result of this data-driven transformation, where trajectories are generalized from sequences of points to sequences of cells, the probability of re-identification already drops significantly. Further techniques can be adopted to lower it even more, obtaining a safe theoretical upper bound for the worst case (i.e., the maximal probability that the linkage attack succeeds), and an extremely

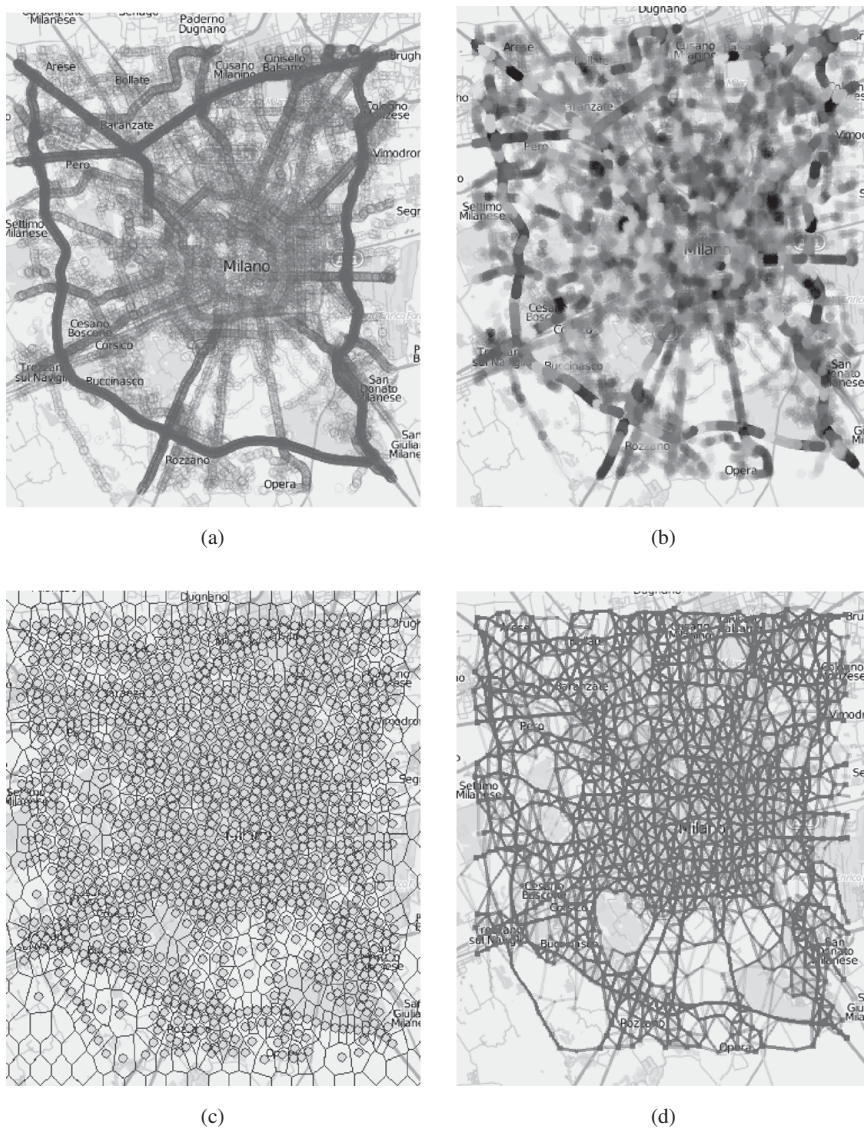


Figure 9.5 Anonymization steps. (a) Characteristic points. (b) Spatial clusters. (c) Territory tessellation. (d) Generalized trajectories. (See color plate.)

low average probability. A possible technique is to ensure that for any subtrajectory used by the attacker, the reidentification probability is always controlled below a given threshold $1/k$; in other words, ensuring the k -anonymity property in the released data set. Here, the notion of k -anonymity proposed is based on the definition of k -harmful trajectory, that is, a trajectory occurring in the database with a frequency less than k . Therefore, a trajectory database D^* is

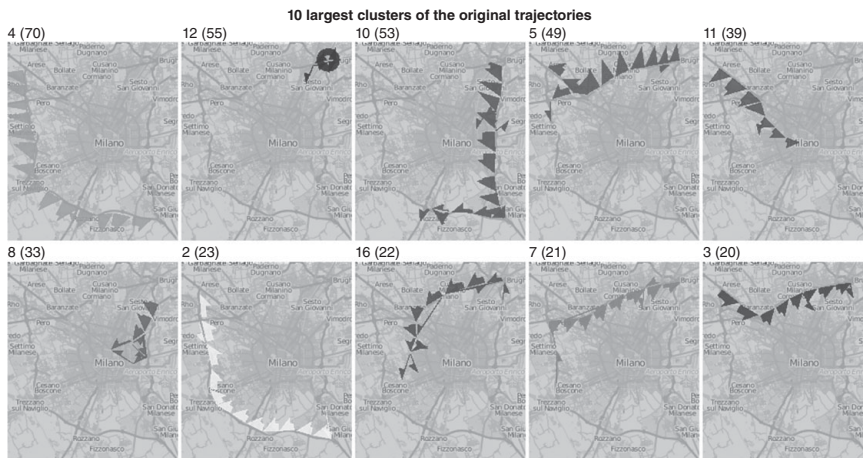


Figure 9.6 Ten largest clusters of the original trajectories.

considered a k -anonymous version of a database D if: each k -harmful trajectory in D appears at least k times in D^* or if it does not appear in D^* any longer. To achieve this k -anonymous database, the generalized trajectories, obtained after the data-driven transformation, are transformed in such a way that all the k -harmful subtrajectories in D are not k -harmful in D^* .

In the example in Figure 9.4, the probability of success is theoretically bounded by $1/20$ (i.e., 20-anonymity is achieved), but the real upper bound for 95% of the attacks is below 10^{-3} .

Clustering Analysis

The above results indicate that the transformed trajectories are orders of magnitude safer than the original data in a measurable sense: *but are they still useful to achieve the desired result, that is, discovering trajectory clusters?*

Figures 9.6 and 9.7 illustrate the most relevant clusters found by mining the original trajectories and the anonymized trajectories, respectively.

A direct effect of the anonymization process is an increase in the concentration of trajectories (i.e., several original trajectories are bundled on the same route); the clustering method will thus be influenced by the variation in the density distribution. The increase in the concentration of trajectories is mainly caused by the reduction of noisy data. In fact, the anonymization process tends to render each trajectory similar to the neighboring ones. This means that the original trajectories, initially classified as noise, can now be “promoted” as members of a cluster. This phenomenon may produce an enlarged version of the original clusters. To evaluate the clustering preservation quantitatively, the F-measure is adopted. The F-measure is usually adopted to express the combined values of precision and recall and is defined as the harmonic mean of the two

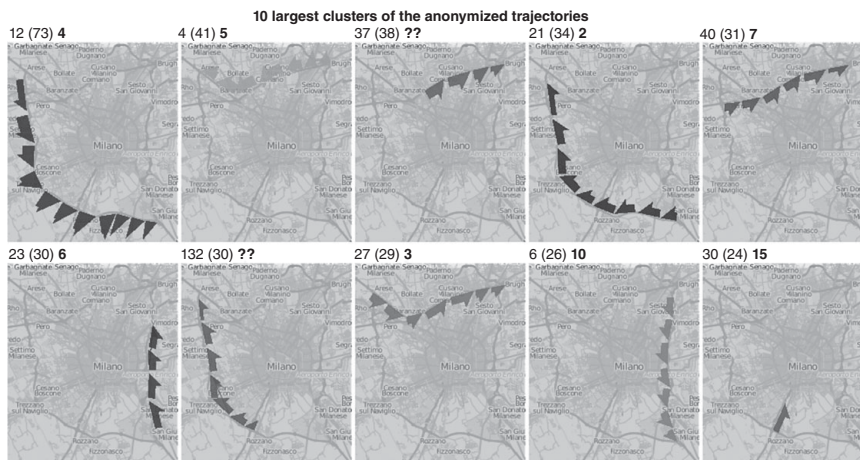


Figure 9.7 Ten largest clusters of the anonymized trajectories.

measures. Here, the recall measures how the cohesion of a cluster is preserved: it is one if the whole original cluster is mapped into a single anonymized cluster, and it tends to zero if the original elements are scattered among several anonymized clusters. The precision measures how the singularity of a cluster is mapped into the anonymized version: if the anonymized cluster contains only elements corresponding to the original cluster its value is one, otherwise the value tends to zero if there are other elements corresponding to other clusters. The contamination of an anonymized cluster may depend on two factors: (1) there are elements corresponding to other original clusters, or (2) there are elements that were formerly noise and have been promoted to members of an anonymized cluster.

The immediate visual perception that the resulting clusters are very similar in the two cases in Figures 9.6 and 9.7 is also confirmed by various cluster comparisons by F-measure, redefined for clustering comparison (Figure 9.8).

The conclusion is that in the illustrated process the desired quality of the analytical results can be achieved in a privacy-preserving setting with concrete formal safeguards and the protection with respect to the linkage attack can be measured.

9.4.2 Trajectory Anonymity by Microaggregation and Perturbation

The previous technique is not suitable when it is necessary to obtain anonymous data preserving real locations in the data. When this requirement has to be satisfied it is possible to use the anonymization methods called *SwapLocations* and *ReachLocations*, which allow anonymizing trajectories composed of original locations.

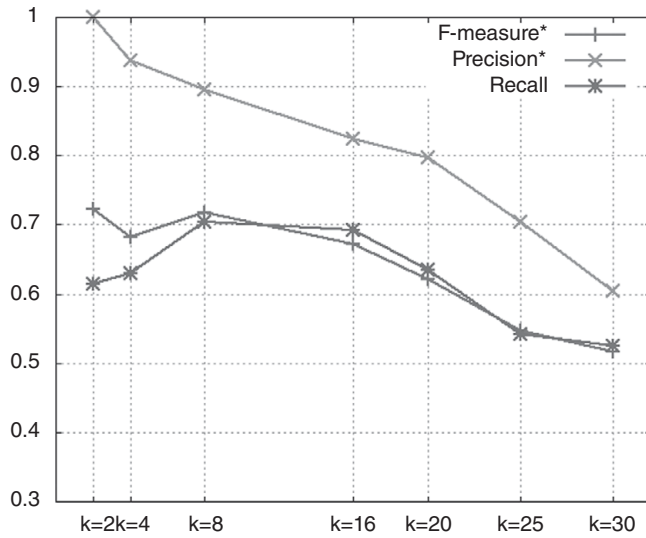


Figure 9.8 Clustering comparison by F-measure.

Attack Model

The aim of these two methods is to protect individuals in trajectory data against the *linkage attack model*. In other words, the attack model is similar to that described in the previous section. The only difference is that the adversary knows that each location in the anonymized trajectories must be in the original data. This is an important point because the linkage of a location with a specific user could reveal the exact location rather than the generalized one. Therefore, it is possible to identify two attacks: (a) finding an anonymized version of a specific real trajectory; and (b) determining if a location belongs to a specific trajectory.

Privacy-Preserving Techniques

How is it possible to guarantee that the probability of success of the attack just described is very low while preserving the utility of the data for meaningful analyses? The countermeasure against the attack in point (a) uses microaggregation to partition the set of trajectories into several clusters, by minimizing the sum of the intracluster distances. The cardinality of each cluster must be between $k + 1$ and $2k - 1$. The purpose of setting k as the cluster size is to obtain trajectory k -anonymity. Given a cluster, the algorithm takes a random trajectory and attempts to cluster each unswapped location l of this trajectory with another $k - 1$ unswapped locations. These locations must belong to different trajectories and the following properties have to be satisfied: (1) the time stamps of these locations differ by no more than a specific time threshold; (2) the spatial coordinates differ by no more than a space threshold. Given a cluster, random swaps

of locations are performed. In the case in which no $k - 1$ suitable locations can be found for creating a cluster, l is removed. As a result, no original location is unswapped in a cluster of k trajectories; as a consequence an adversary is not able to link a true trajectory to an anonymized one with probability higher than $\frac{1}{k}$. The countermeasure against the attack in point (b) has to guarantee that from a given location, only those locations at a distance below a threshold following a path in an underlying network are considered to be directly reachable. Each location is k -anonymized independently using the whole set of locations of all trajectories. Specifically, given a location l , a cluster with at least other $k - 1$ locations is constructed in such a way that the locations belong to k different trajectories and the location respects a specific spatial and temporal distance with respect to the location l . Then, the spatial coordinates of the location l are swapped with the spatial coordinates of some random location in the cluster. The process stops when all locations appear swapped at least once. The result of this transformation is that a location l of a true trajectory appears in its anonymized version with a probability at most of $\frac{1}{k}$ (*location k -diversity*).

Data Utility Analysis

The above techniques provide trajectories with a formal guarantee of protection; but now an important question is if the transformed data are still useful to achieve the desired analytical results. A suitable evaluation showed that the anonymization of trajectories by the two techniques causes an acceptable space distortion and makes the anonymized trajectories suitable for range queries by providing low distortion for every value of k . The range query measures evaluate the relative position of a moving object with respect to a region.

9.5 Conclusions

Mobility data represent an important source of knowledge but sharing of these data can raise serious privacy concerns: mobility data may potentially reveal many facets of a person's private life. Mobility data privacy problems have to be addressed in two different scenarios: online location-based services and offline data analysis context. Many recent research works have focused on the study of privacy protection in spatio-temporal data and many privacy-enhancing technologies have been proposed, which essentially aim at finding an acceptable trade-off between data privacy on the one hand and data utility on the other. So far, the common result obtained is that no general method exists that is capable of both dealing with "generic personal data" and preserving "generic analytical results." A recent paradigm, called *privacy by design*, promises a quality leap in the conflict between data protection and data utility. The application of this paradigm in mobility data mining showed that the desired quality of the analytical results can be achieved in a privacy-preserving setting with concrete formal

safeguards, so that the protection with respect to the linkage attack can be measured. The implication of this finding is far reaching; once an analytical process has been found and specified, it can be deployed and replicated with the mentioned privacy-preserving safeguards in order to perform mobility data analyses in different periods of time, in different cities, in different contexts: once deployed, it is a safe service that generates knowledge of the expected quality starting from truly anonymous data.

9.6 Bibliographic Notes

The literature on privacy in mobility data is becoming extensive. In the following, we will provide an essential list of bibliographic references for the reader, including those describing the problems and the solutions discussed in the chapter.

Privacy issues in mobility data mining were deeply discussed by [Giannotti and Pedreschi \(2008\)](#). [Monreale et al. \(2010\)](#) present an overview on the main privacy-preserving data publishing and mining techniques proposed by the data mining community and by the statistical disclosure control community. This contribution also discusses the privacy issues in complex domains, focusing the attention on the context of spatio-temporal data and describing some approaches proposed for anonymity of this type of data.

The k -anonymity model was introduced by [Samarati and Sweeney \(1998\)](#), and then [Machanavajjhala et al. \(2007\)](#) and [Li et al. \(2007\)](#) proposed l -diversity and t -closeness to overcome the weaknesses of k -anonymity. This privacy model and its variants have been widely adopted to achieve privacy in movement data, especially in privacy-preserving publishing of trajectories. A recent survey on this topic is presented by [Bonchi et al. \(2011\)](#).

The problem of hiding sensitive spatio-temporal patterns in trajectory data was studied in [Abul et al. \(2010\)](#), while a privacy-preserving clustering method in horizontally partitioned spatio-temporal data was described by [Inan and Saygin \(2006\)](#).

The *privacy by design* paradigm in data mining was introduced by [Monreale \(2011\)](#). This PhD thesis proposed this novel methodology to address the privacy issues in complex data with a particular focus on data with a sequential nature such as trajectory data.

Lastly, techniques for trajectory anonymity based on microaggregation and perturbation were introduced in a recent work by [Domingo-Ferrer and Trujillo-Rasua \(2012\)](#).

PART III

MOBILITY APPLICATIONS

10

Car Traffic Monitoring

Davy Janssens, Mirco Nanni, and Salvatore Rinzivillo

10.1 Traffic Modeling and Transportation Science

Transportation science, together with its related research fields, is a key discipline of today's society, due to its potential impact on several levels of societal organization and resource usage. In this chapter we will discuss some of the main issues of next generation transportation solutions, and traffic models in particular, and describe case studies where mobility data analysis can help provide some answers.

Relevance of Traffic Modeling

In a research report by the United Nations in 2001, it was postulated that the transport sector accounts for about 25% of the total commercial energy consumed worldwide and that it consumes approximately 50% of the total oil produced. The International Energy Agency (IEA) predicts that the transport sector will overtake industry as the largest energy user by 2020. Unfortunately, that has major negative economic, social, and environmental side effects. At the environmental level, transport has proven to be the source of nitrogen oxides, sulfur oxides, and other volatile organic compounds, all which have negative environmental and health implications. Pollution, environmental degradation, space consumption, and greenhouse gases are receiving increasing attention as the immediately detectable externalities of transport and land-use development patterns. At the economic level, accidents and congestions, traffic gridlocks, stress from pedestrian and vehicular conflict, inefficient public transport, and urban sprawl are all associated with unsustainable transport systems that indirectly represent costs to society. At the social level, research reports seem to suggest that in areas where public transport is often second-rate or absent and where the levels of car ownership are significantly lower, a higher degree of risk for social exclusion is perceived. Whereas a good transport system increases the

opportunities to satisfy interaction needs, a poorly connected transport system limits economic and social development (Ortúzar and Willumsen, 2002). The transport system thus allows individuals to trade time for space when moving to (activity) locations (Miller, 2003; Rietveld, 1994).

Traffic Modeling Standard

Rising concerns over these increasingly intolerable externalities have generated particular interest in how transport planning policies might at least moderate the pressures in growth in personal mobility and support the principles of sustainable development (Barrett, 1996; Salomon et al., 1993). Originally, transport planning policies focused on mastering the massive growth in car mobility. These policies were adopted in an immediate response to the predicted growth in (car) mobility. The estimation and forecasting of travel demand and behavior were handled by a standard methodological approach commonly referred to as the four-step modeling approach consisting of trip generation, trip distribution, mode choice, and assignment of travel demand to highway and transit networks. In the trip generation stage, the goal is to predict the total number of trips generated and attracted to each zone of the study area. In the second stage, the question is how to distribute trips among destinations. The result of this step is a 2D array of cells (matrix) where rows and columns represent each of the zones in the study area and the cells contain the number of trips that go from the origin zone (in the rows) to the destination zone (in the columns). The latter is also known as an *origin–destination matrix*. Next, in the third step the transport mode is chosen. The output of this step is typically an origin–destination matrix that represents the number of trips that are carried out by the different transport modes. While the previous three steps mainly deal with the demand side of travel, the last step in the four-stage methodology is mainly related with the supply side. In this step, the supply side of the transportation system – which is made up of a road network and is represented by links and costs – is confronted with the demand side of travel that has been estimated in the first three steps. The result of this step is the amount traffic projected on the road network, typically represented as number of vehicles on road segments.

Toward Data-Driven/Aware Models

In parallel with the traffic science evolution discussed above, there is growing literature and research available, originating from the field of mobility data mining (see Chapter 6 of this book), which emerged only recently, during the last decade. While the overall goal is the same, that is, to help policy makers to deal with traffic-related questions, the techniques used and the processes adopted are completely different. The main difference is the fact that most of the techniques are fully data driven and therefore also less policy sensitive. It can be very interesting to see how both domains could complement each

other, and hopefully evolve into next-generation traffic modeling systems able to better capture the dynamics of real human mobility. The following sections try to trace some connections between the two fields, providing examples of a mobility data mining approach to deal with some basic questions that naturally arise when dealing with traffic understanding and modeling. Moreover, the remaining part of the chapter will provide a series of real analytical scenarios enabled by the methods and techniques presented in the previous chapters of this book.

10.2 Data-Driven Traffic Models

Mobility phenomena are sensed by means of several data collections and monitoring. For example, traditional transportation methods use inductive loops, cameras, sensors, and counters to measure specific arc roads of the network. All these observations are merged and integrated within existing models in order to refine and fit the model parameters. Thus, the integration of the mobility models extracted from real mobility data is crucial. There is a strong need for an accurate mobility demand evaluation that calls for a data-driven approach to obtain better estimations of mobility phenomena. In this chapter we will show how to cope with a set of problems that provide the analyst with a particular view on specific mobility behaviors. At the base of such a process there is a large preprocessing step with the duty of integrating and merging different data sources. For the objective of this chapter we assume that this step has already been performed and all the data are available for the analysis in the correct format. We show how to master the complexity of the mobility knowledge discovery process by means of an organic analytical framework centered on the concept of *trajectory*. In particular, we show how the semantic deficiency of big mobility data can be bridged by their size and precision. To this purpose, we describe the key results obtained in a large-scale experiment conducted with the mobility analysts of the cities of Milan and Pisa, on the basis of real life GPS tracks sensed from tens of thousands of private cars. We show how it is possible to find answers to challenging analytical questions about mobility behavior, which are not supported by the current generation of commercial systems, such as:

1. What are the most popular itineraries followed from the origin to the destination of people's travels? What are the routes, timing, and volume for each such itinerary?
2. How do people leave the city toward suburban areas (or vice versa)? What is the spatio-temporal distribution of such trips?
3. How can we understand the accessibility to key mobility attractors, such as large facilities, railway stations, or airports? How do people behave when approaching an attractor?

4. How can we detect an extraordinary event and understand the associated mobility behavior? How and when do people reach and leave the event's location? What is the spatio-temporal distribution of such (portion of) trips?
5. What will be the areas with highest traffic volume in the next hour(s)? To what extent are our predictions accurate?
6. Are there geographic borders that emerge from the way people use the territory for their daily activities? If so, how do we define such borders? Do these borders match the administrative ones?

More than just examples, these questions are paradigmatic representatives of the analysts' need to disentangle the huge diversity of individual whereabouts and discover the subgroups of travels characterized by some common behavior or purpose. It is no surprise, then, that finding answers to these questions is beyond the limits of the current generation of commercial systems, and cannot even be accomplished by simply applying single known research prototypes, such as the mobility data mining methods presented in Chapter 6. There is the need for a *mobility knowledge discovery process* aimed at *discovering interesting subgroups of vehicles and travels characterized by some common movement behavior*. To perform this kind of analysis, a complete querying, analysis, and mining system is needed, able to support the overall knowledge discovery process centered around the trajectory concept. In this chapter we will provide analytical answers based on the tools and the knowledge discovery process handled by an analytical framework named M-Atlas, already introduced in Chapter 7. A general analytical process on mobility data follows several steps. First the data are explored by the analyst to understand and comprehend the several dimensions of the observed phenomena. In Section 10.3 we present a set of statistical methods that have a twofold objective: on one hand they serve to assess the general validity of the data with respect to background knowledge; on the other hand they provide insight into the internal distribution of data dimensions. Once the analyst has acquired a deep understanding of the data, he or she can proceed with the exploration. Section 10.4 provides a set of analytical scenarios where different mobility data mining methods are used to find answers to the questions we have proposed. The methods used have already been presented in Chapter 6, thus we refer the reader to that chapter and we will not give further details on the internal functionalities of each algorithm.

To present a paradigmatic mobility knowledge discovery process we concentrate on massive, real-life GPS data sets, obtained from tens of thousands of private vehicles with on-board GPS receivers. The owners of these cars are subscribers to a pay-as-you-drive car insurance contract, under which the tracked trajectories of each vehicle are periodically sent (through the GSM network) to a central server for antifraud and antitheft purposes. This data set has been

donated for research purposes by *Octo Telematics Italia S.p.A.*,¹ the leader for this sector in Europe. We use two GPS data sets: the first, *Milano2007*, describes approximately 17,000 cars tracked during one week (from April 1 through April 7, 2007) of ordinary mobile activity in the urban area of the city of Milan (a $20 \text{ km} \times 20 \text{ km}$ square). The second, *Pisa2010*, contains approximately 40,000 cars tracked during 5 weeks (from June 14 through July 18, 2011) in coastal Tuscany, a $100 \text{ km} \times 100 \text{ km}$ square centered on the city of Pisa. The average sampling rate of the GPS receivers is 30 seconds. Globally, *Milano2007* consists of approximately 2 million observations and *Pisa2010* of approximately 20 million observations, each consisting of a quadruple $(id, lat, long, t)$, where id is the car identifier, $(lat, long)$ are the spatial coordinates, and t the time of the observation. The car identifiers are pseudonymized, in order to achieve a basic level of anonymity

The resolution of the spatial coordinates is at 10^{-6} degrees, and the error of the positioning system is estimated at 10–20 m in normal conditions. The temporal resolution is in seconds. All the observations of the same car id over the entire observation period are chained together in increasing temporal order into a global *trajectory* of car id . Using the trajectory reconstruction techniques presented in Chapter 2, we obtained approximately 200,000 different travels in *Milano2007*, and approximately 1,500,000 different travels in *Pisa2010*.

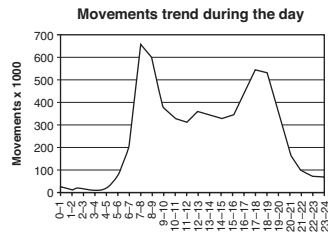
10.3 Data Understanding

Since the data we can use for analysis are a sample of the real population, as a first step we need to evaluate their representativeness and statistical significance. We do that through a set of statistical evaluations that analyze the distributions of typical movement dynamics properties, such as speed, length of each trip, and temporal location. In some cases these same measurements are estimated also by traditional transportation methods, therefore a comparison is possible in order to assess meaningfulness of the data sample as proxy of real mobility phenomena. For the *Milano2007* data set, we compared it against the survey data collected in 2005–6 by the local mobility agency AMA,² although the two data sources differ in both the sampled population and the kind of collected information: mobility reports are obtained through a survey campaign and include flows of private vehicles but also public transportation and pedestrians.

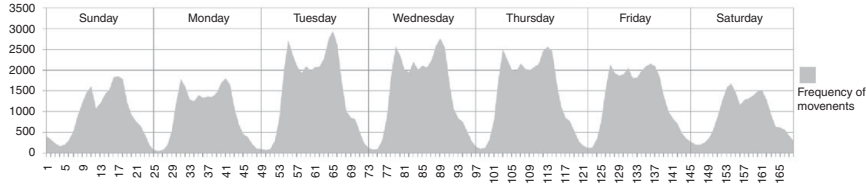
Since the basic components of mobility data are the spatial and temporal dimensions, we focus on the statistical analysis of these dimensions separately. First, we try to understand when people are moving during the day. In particular,

¹ <http://www.octotelematics.it>

² <http://www.ama-mi.it/english>



(a)



(b)

Figure 10.1 Movement distribution by hour: (a) representative weekday in MilanoSurvey and (b) entire week in Milano2007.

we measure the number of moving vehicles in every hour of the day, and create a histogram over the entire week. The result is shown in Figure 10.1 together with a typical day distribution provided by the MilanoSurvey.

The two distributions match significantly, especially for the days from the second to the fifth of the week, which actually represent *regular* working days, from Monday to Thursday. Background knowledge and domain expertise, in this case, help to explain the anomaly of Friday. Indeed, it was Easter Friday, which explains the different shape with respect to previous weekdays. Within working days, the most relevant deviation from the survey data is a higher volume of movement between the two peaks in the rush hours and (to a minor extent) the later part of the day. The assessment with the mobility agency revealed that the results are coherent, and actually in this specific case GSP data prove to be more robust than surveys. Indeed, the latter are known to underestimate the movements in these periods of the day, due to the fact that interviewed people tend not to report their occasional mobility, such as going to the dentist or visiting a friend. Also, GPS data contain mobile activity of people who do not live in the greater metropolitan area, while the survey focuses on Milano residents.

The second dimension of analysis is the spatial component. Here we can try to estimate the presence of population on the territory through GPS data, and compare it with correspondent results obtained from the declared places of residence on the surveys (see Figure 10.2a). A similar estimate was obtained on Milano2007 by (1) partitioning the space into a regular grid and (2) counting for each cell the number of vehicles that were stationary in the cell for each time

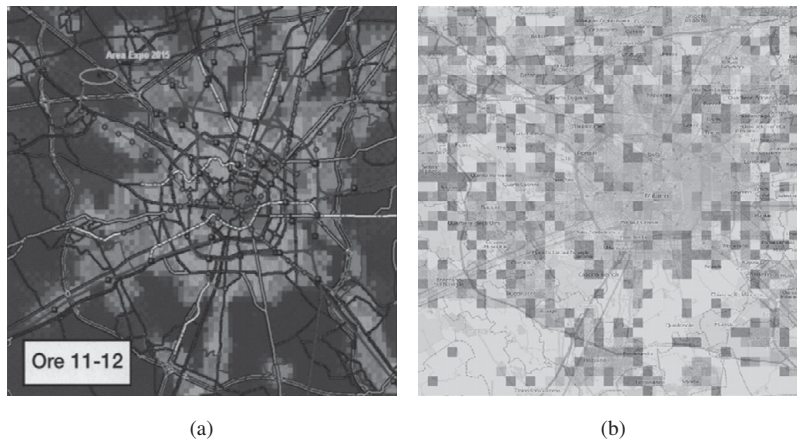


Figure 10.2 Presence distribution between 11 A.M. and noon, (a) survey, (b) GPS data; frequent locations plotted with lighter shades. (See color plate.)

interval. Such values were averaged over all (regular) working days available. Figure 10.2b shows the results.

The two distributions match well in most locations, including some particular areas along main streets and suburban residential areas, confirming again the coherence of results obtained with survey and mobility data. The main deviation occurs in the inner city center, where a high density spot found by surveys is significantly lower in Milano2007: this is explained by the strong access restrictions on private cars in the city center, as well as by the limited capacity of roads and traffic, which causes an underrepresentation in the GPS data of the people who reach their workplaces in the center with public transportation.

These two first analyses are useful to have a first insight of the data. The next aspect to analyze is the exploration of the movement dynamics, that is, identifying the movement quantities represented in the trajectory data sets: the length of a trip, and the duration of a trip, the correlation of length and speed of trips.

Trip Length and Duration

Figure 10.3a shows the distribution of trip length (in km), as estimated from GPS trajectories. The heavy-tailed distribution of trip length highlights how there are many short trips of a few kilometers, and few, but nonnegligible, very long trips of tens or even hundreds of kilometers; a similar consideration applies to the distribution of trip duration, shown in Figure 10.3b. The lesson learned here confirms how mobility is a complex phenomenon, where a simple notion of *average behavior* may be misleading. In fact the variance of the distribution is so large that the representativeness of the average value is limited.

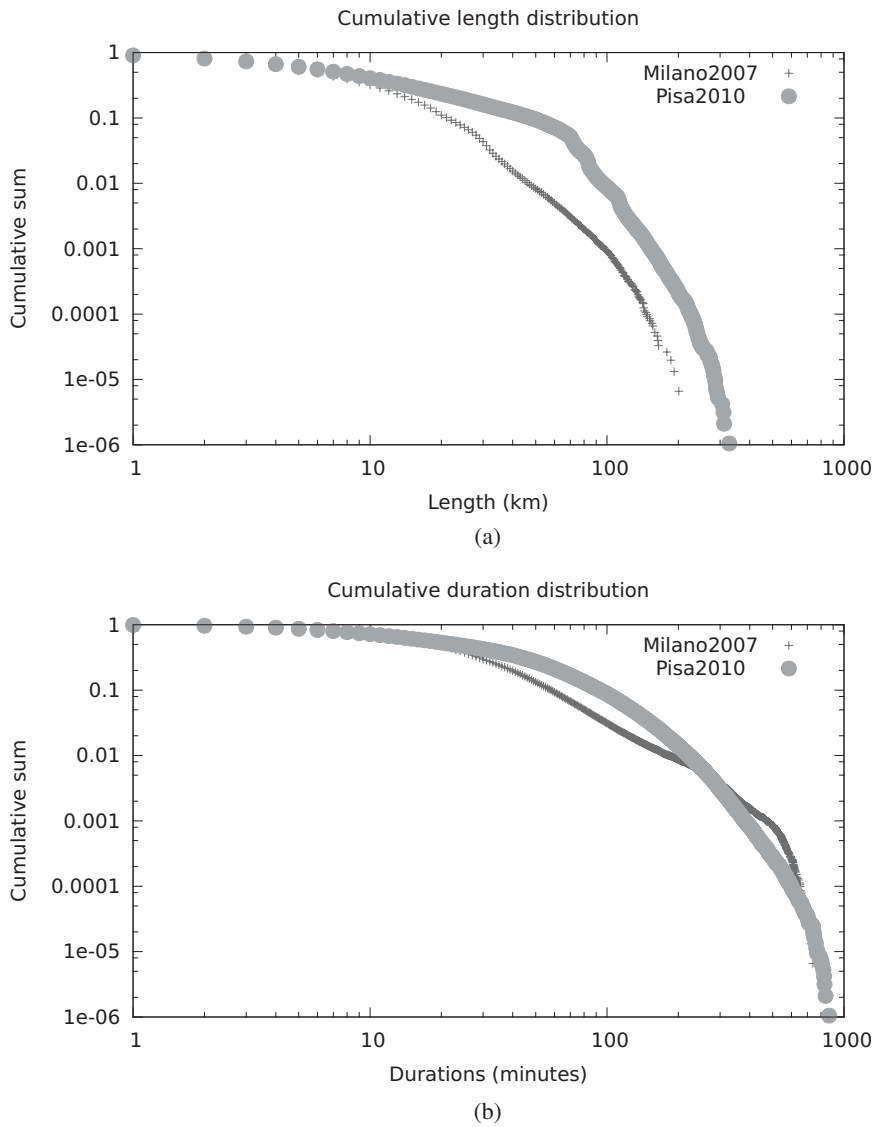


Figure 10.3 (a) Trip length cumulative distribution in log-log scale; (b) trip duration cumulative distribution in log-log scale.

Correlation of Length and Speed of Trips

Figure 10.4 shows the correlation plots of trip length (in km) and speed (in km/h). For each speed value s , the plot reports the distribution of distance traveled by all trips with average speed s . For each value of speed the box plot reports median, 25th, 75th, and 99th percentiles. Notice that with this specific

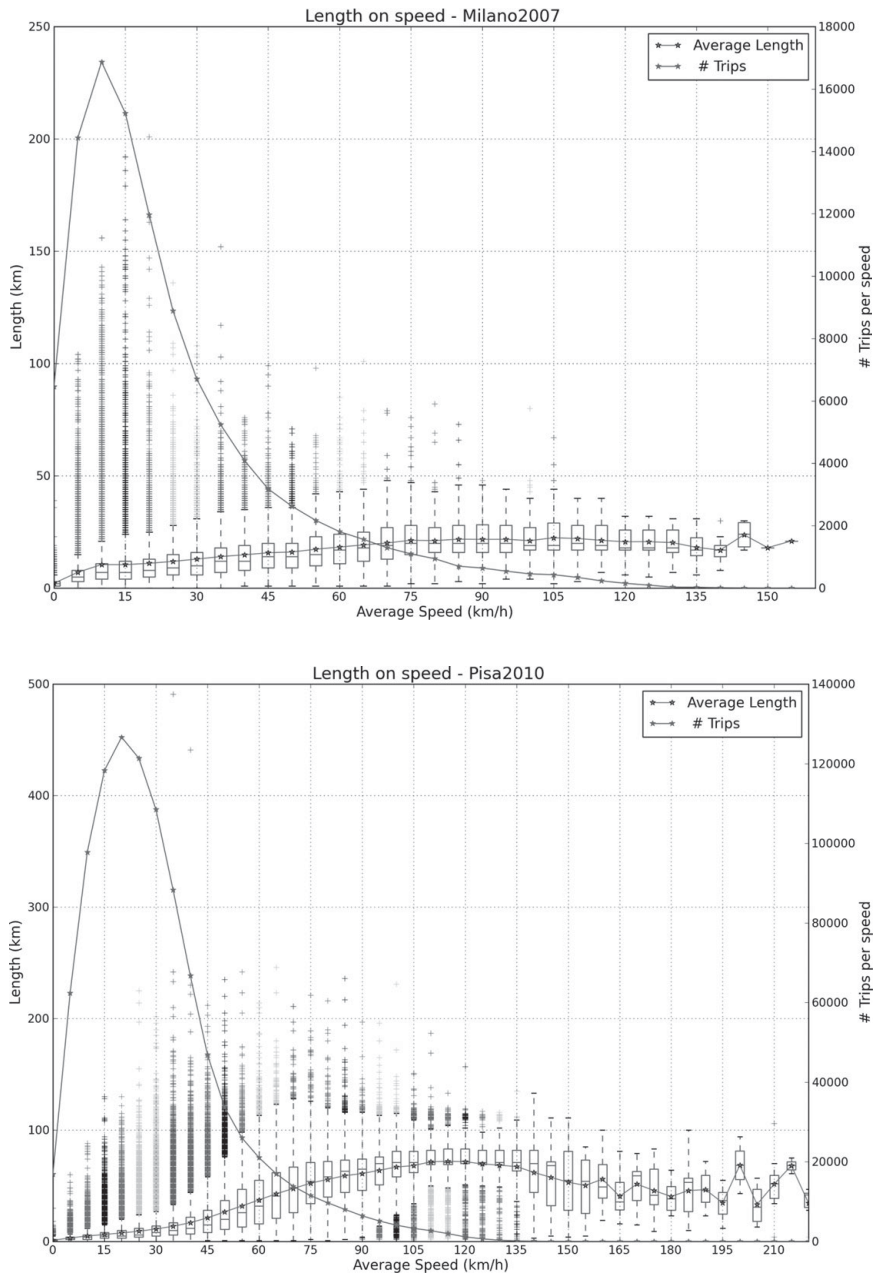


Figure 10.4 Correlation plot of length and average speed of trips and number of trips per speed for the Milano2007 and Pisa2010 data sets.

selection of movements (in this case the trajectories with a specific average speed s) the average length observed seems to capture well the behavior of the observed trips, because the variance is low. In the Milano2007 data set, the plot shows how the distance traveled grows linearly with speed, as expected, only up to 80 km/h, while it decreases for higher speed. In the Pisa2010 data set, the distance traveled grows linearly up to 110 km/h, with a low slope between 20 and 40 km/h. The plots show also the number of trips for each speed value: the high diversity of lengths for speeds beyond 130 km/h (the highest speed limit in Italy) is due to the low number of travels with that velocity and can be considered as noise, coherently with the intuition that very fast trips take place in particular situations of light traffic, typically at night.

We learned two lessons from our basic analytical explorations. First, all statistics confirmed that there is a huge complexity represented in the data, a wide variability of individual mobility behaviors that cannot be fully understood in their diversity by looking only at macroscopic, global measures and laws. Second, we realized that the basic spatio-temporal statistics are not well suited to support the discovery and analysis of *movement patterns*, because the very nature of a trajectory requires a deep understanding of the internal dynamics of movement and their relations with the context. For these particular aspects, we exploit the mobility data mining methods introduced in Chapter 6.

10.4 Analysis of Movement Behavior

To answer the questions proposed in Section 10.2, a complete mobility knowledge discovery process centered around the trajectory concept is needed. Such a process should be powered by a suitable system with the aim of supporting interactive, iterative visual exploration of the analytical results, thus enabling the analyst to combine different forms of knowledge and drive the analysis toward the discovery of interesting movement patterns. An instance of a mobility knowledge discovery process has been introduced in Chapter 7. In this section we show how the mobility data analysis tools are able to provide answers to the questions discussed.

10.4.1 Origin-Destination Matrix Exploration

As stated in Section 10.1, origin–destination (OD) matrix models provide a simple and compact representation of traffic dynamics, by abstracting detailed actual movements by means of aggregation in flows between two regions. While the traditional OD matrices are modeled by statistical analysis of surveys, sample observations, and continuous refinements of the original models, the large availability of sensed tracks from real vehicles enables the automatic extraction of

OD matrices. The suggested procedure is based on the traditional transportation science approach where a spatial tessellation is used to generalize and summarize movements. Starting from a given spatial tessellation, each GPS-tracked movement is mapped to its corresponding origin and destination, that is, the former is the region where the trip begins and the second is the region where the trip stops. This kind of representation loses the focus on *how* people move, that is, by which routes, and maintains only the information of the origin and the destination. Depending on the time interval considered, it is possible to reconstruct a OD matrix for different time periods, allowing a precise characterization of the evolution of traffic demand during time.

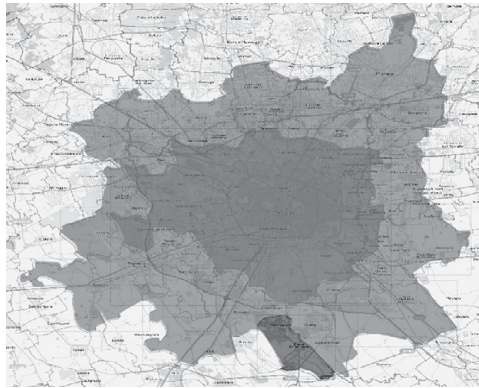
For the mobility data analyst, the OD matrix represents a valuable tool to explore mobility data of a region, since it helps to reveal relevant flows and time intervals. For example, to explore the main flows from the city center toward the suburbs, we start by considering the administrative borders of Milan and its adjacent municipalities (see Figure 10.5a). A visual interface may enable the analyst to disentangle the complexity of the model by exploring relevant flows on the screen. There exist several methods to visualize and explore OD matrices (see Chapter 8 for a review of visualization methods for flows); as an example, Figure 10.5b shows the visual interface provided by the M-Atlas system. In our analysis, we focus on the flows leaving the city center of Milan toward the north east suburbs.

10.4.2 Most Popular Itineraries from the City Center to Suburban Areas

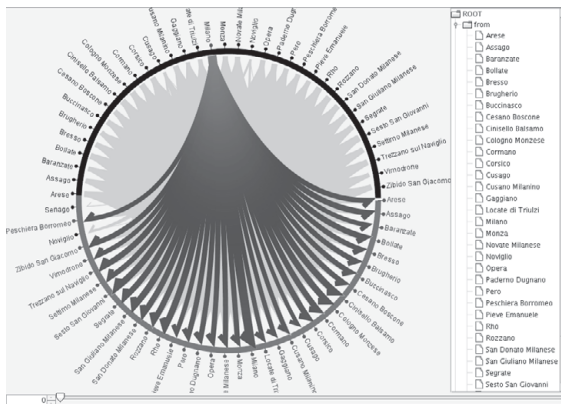
Once we have selected a relevant set of flows, we can focus the analysis on the individual trips associated to them.

The resulting trajectories are presented in Figure 10.6a. Despite the fact that all these trips originate in the city center and end in the northeast suburbs, a broad diversity is still evident. In order to understand which are the most popular itineraries followed by the selected travels, we apply an algorithm that automatically detects significant groups of similar trips. In particular we use the density-based clustering algorithm with the *Spatial Route* distance function introduced in Chapter 6. Given two trajectories, the route similarity function returns a numeric estimation of their diversity: if the trajectories are equal it tends to zero, otherwise it tends to infinity. A route is relevant for the mobility analyst if it is followed by many vehicles. The clustering algorithm selects effectively groups of trajectories with similar paths and thus provides a selection of frequent routes. Trajectories that do not belong to any group are labeled as noise, and the user might decide to discard them or, in some particular cases, to analyze them separately from the others.

The clustering algorithm produces a set of clusters, each of which can be visualized by means of a thematic rendering where the trajectories in the same



(a)

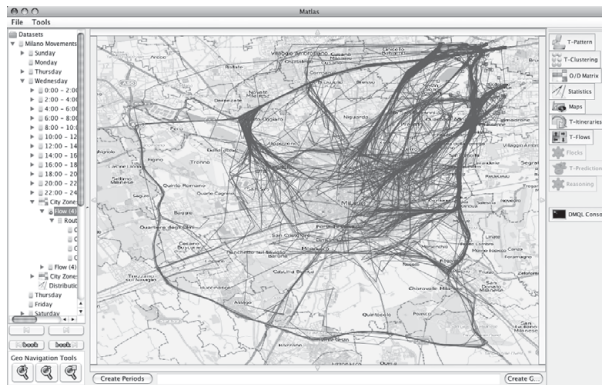


(b)

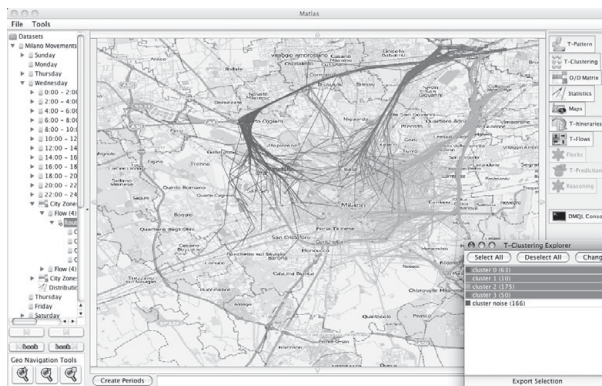
Figure 10.5 The resulting OD matrix model for Milano2007 on a specific weekday (Wednesday, April 3rd). (a) The regions used as input for the model: the center region contains the administrative borders of Milan and the adjacent cells represent neighbouring cities; (b) The visual interface to browse the OD matrix: each region is represented with a node, nodes are displayed in a circular layout. The arc connecting two nodes represents the flow, i.e., the number of trips from the origin to the destination node; the arc width is proportional to the flow. The analyst visually browses the OD matrix, either selecting some specified origins and/or destinations or highlighting the main flows by setting a minimum support threshold.

cluster are drawn with the same color. Figure 10.6b shows how the most popular clusters highlight the main routes used by drivers to leave the center toward the northeast.

The frequent behaviors highlighted by the clustering process followed above might in some cases be characteristic of some specific time period (for instance,



(a)



(b)

Figure 10.6 The result of clustering from the trajectories moving from the center to the northeast area. (a) The input data set for the clustering algorithm: the trajectories moving from the center to the northeast area. (b) The resulting clusters using the *Route Similarity* distance function. The clusters are visualized using a themed color, and the analyst can select and browse them separately.

it might arise only on Monday) or might have a general validity. In order to distinguish these two cases, we need to measure how the population of the clusters is distributed over the days of the week, and this task can be accomplished using the clustering as an unsupervised classification model. More precisely, after the clusters have been extracted for a specific day, one or more representatives, named specimens, for each cluster are computed and such representatives are used to classify the trajectories in other days of the week: every new (unseen) trajectory T is classified by assigning it to the closest specimen (and therefore to the cluster it represents). If the distance between T and such a specimen is too high, however, the trajectory is assigned to the *noise*. Figure 10.7 shows how

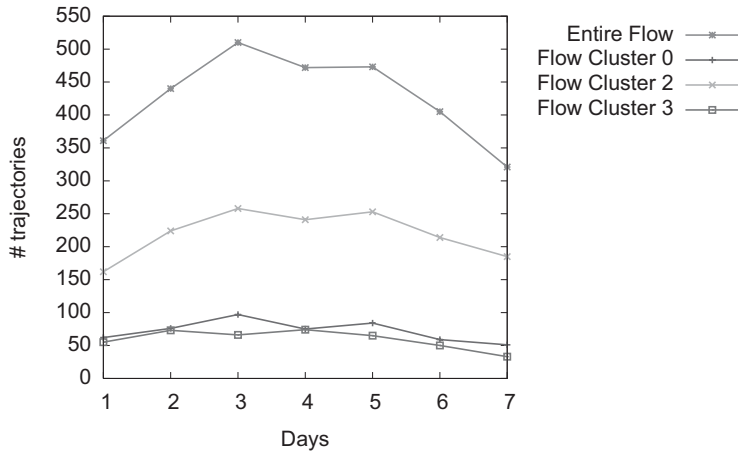


Figure 10.7 Distribution of estimated cardinality of three main clusters and number of all travels from the city center to NE suburbs over the week April 1st (Sun) – 7th (Sat). Clusters 0 and 3 are essentially constant with a small decrease during the weekend (days 1 and 7), while Cluster 2 has a shape similar to the general flow, with a significant decrease during the weekend.

the distribution of the estimated population of the three clusters varies during the week. The figure highlights that Clusters 0 and 3 are stable over the entire week, while the most popular cluster, 2 (green), is stable over weekdays only, suggesting that it is composed mainly of outbound commuters who travel during working days.

The next question is to determine if the commuters of Cluster 2 travel from home to work or vice versa. The answer can be explored by analyzing the temporal distribution of the trips of the cluster over the hours of a weekday (see Figure 10.8b).

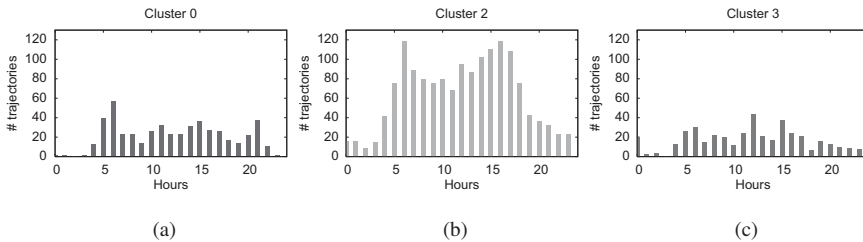


Figure 10.8 Temporal distribution of the trajectories in the clusters of Figure 10.6b on the hours of weekdays. Cluster 0 and Cluster 3 (a, c) do not exhibit significant peaks, while Cluster 2 (b) has a peak in the morning and one in the afternoon. The temporal profile of Cluster 2 captures two commuting behaviors: a group leaving the city in the morning (commuters going to work outside), and a larger group leaving the city in the late afternoon (commuters coming back home in the suburbs after work).

The Analytical Process in a Mobility Data Mining Platform

To give an idea of how the analytical process we described above concretely maps into a dedicated analytical framework, we show here how the M-Atlas system allows us to handle such complexity by means of an SQL-like language called DMQL. All the analysis presented in this chapter can be expressed in such language. Due to space limits, we cannot show here the corresponding queries for all examples. We show how the previous process can be described through DMQL. A more detailed introduction to the system is presented in Chapter 7.

First, the OD matrix is extracted according to a spatial tessellation and then the trajectories between a given origin and several destinations are retrieved from the data set:

```
CREATE MODEL MilanODMatrix AS MINE ODMATRIX
FROM (SELECT t.id, t.trajectory FROM TrajectoryTable t),
      (SELECT orig.id, orig.area FROM MunicipalityTable orig),
      (SELECT dest.id, dest.area FROM MunicipalityTable dest)

CREATE RELATION CenterToNESuburbTrajectories USING ENTAIL
FROM (SELECT t.id, t.trajectory
      FROM TrajectoryTable t, MilanODMatrix m
      WHERE m.origin = Milan AND
            m.destination IN (Monza, ..., Brugherio))
```

The selected trajectories are then clustered to extract groups of trips with similar characteristics. In the following query the route similarity function is used:

```
CREATE MODEL ClusteringTable AS MINE T-CLUSTERING
FROM (Select t.id, t.trajectory from CenterToNESuburbTrajs t)
SET T-CLUSTERING.FUNCTION = ROUTE_SIMILARITY AND
     T-CLUSTERING.EPS = 400 AND
     T-CLUSTERING.MIN_PTS = 5
```

The extraction of cluster specimens from a specific day of the week and the classification of new trajectories are performed by the following queries:

```
CREATE MODEL WednesdaySpecimens AS MINE SPECIMENS
FROM (SELECT id, trajectory, cid FROM WedTrajsToClusters)
SET SPECIMENS.MAX_DISTANCE = 750 AND
     SPECIMENS.METHOD = ROUTE_SIMILARITY

CREATE TRANSFORMATION ClassifiedTrajectories
USING SPECIMENS_CLASSIFIER
```

```

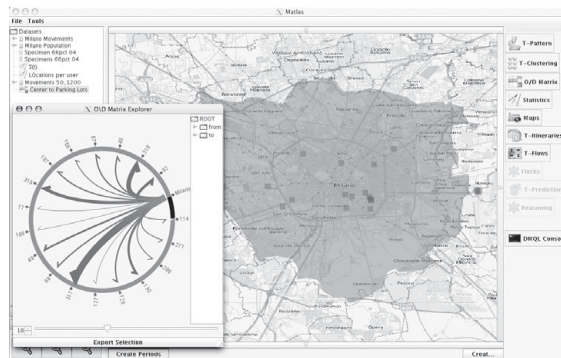
FROM (SELECT id, trajectory FROM TrajectoryTable)
SET SPECIMENS_CLASSIFIER.SPECIMENS =
  (SELECT * FROM WednesdaySpecimens) AND
  SPECIMENS_CLASSIFIER.METHOD = ROUTE_SIMILARITY

```

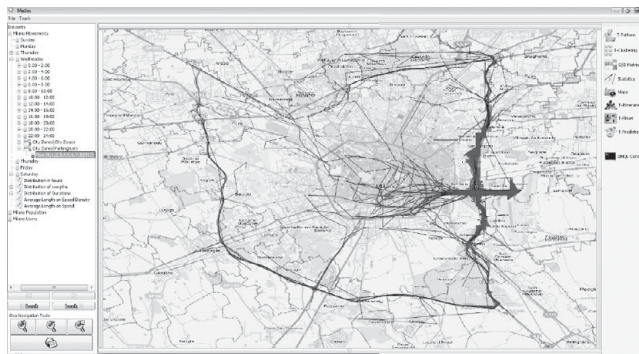
This approach to the management of the mobility knowledge discovery process allows the interoperability of models and data, and it also provides a clear tool to summarize and formally define the analytical process.

10.4.3 Access to Key Mobility Attractors

To understand how users access big mobility attractors, we focus on the travels ending in a specific parking lot of the city. An advanced knowledge of the dynamics of use of a parking lot allows the mobility agency to plan specific fares or to notify the users of extraordinary events or interruption of services. For this case study we have selected the parking lots of the Linate Airport. Figure 10.9 shows the set of trajectories that start in Milan and end in the airport parking lot, selected by means of a OD matrix selection. It is evident that vehicles start from a broad diversity of locations, but converge toward the parking lot. Our goal is to characterize the typical behaviors of vehicles when approaching the attractor, a task that cannot be directly addressed by clustering, due to the fact that clustering generally works at the level of whole trajectories, while the behaviors might emerge just on shorter subtrajectories. Also, simply predefining a set of directions of approach and counting how many trips reach the attractor from each of them answers our request only partially, as we want to characterize behaviors, which might include not only incoming directions but also particular paths followed (e.g., common shortcuts or detours). As an example, we focus here on frequent segments of trips that are followed by a significant volume of vehicles, a feature that can be directly detected by mining trajectory patterns (see Chapter 6). We recall that trajectory patterns describe sequences of regions that appear frequently in the data, together with their typical transition times. Figure 10.9b is a visual summary of the trajectory patterns that are supported by at least 5% of the travels to Linate. As we can see, they allow us to characterize the three main routes to approach the attractor, together with the different travel times. Figure 10.10 focuses on the three most frequent trajectory patterns. Observe how the trajectory patterns approaching the airport from north are longer than those from south, highlighting that the northern travels tend to concentrate on the outer ring earlier than the southern travels, which instead use a small segment of the ring. This behavior suggests the presence of more alternative routes to get to the proximity of the airport from the south and city center than from the north.

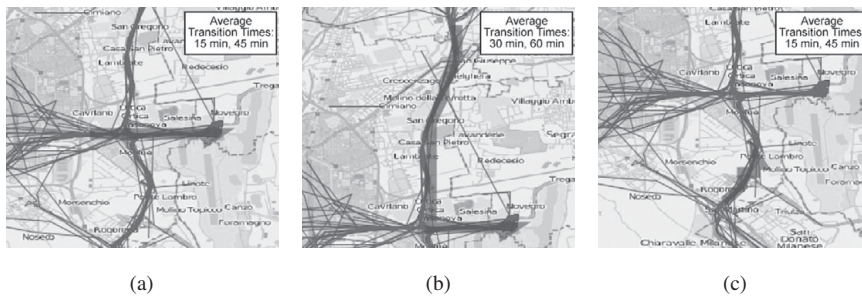


(a)



(b)

Figure 10.9 Accessibility to parking lots. (a) Asymmetric OD matrix from Milano (origin) toward parking lots (destinations). The highest fluxes to parking lots are highlighted by adjusting the frequency threshold slide bar (bottom left). The biggest attractor is parking lot 317 (Linate airport). (b) Travels from Milano to the Linate Airport parking lot, and summary of associated trajectory patterns, characterizing how the travels approach the final destination.



(a)

(b)

(c)

Figure 10.10 Most significant trajectory patterns for traffic directed to Linate airport: (a) from the city center, (b) from north ring, (c) from south ring. Transition times are reported in the insets.

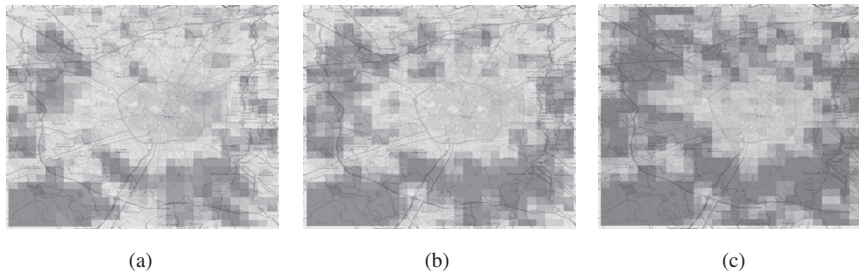


Figure 10.11 Distribution of presence on Tuesday, April 3rd, in three contiguous time slots of 2 hours: (a) from 6 P.M. to 8 P.M., (b) from 8 P.M. to 10 P.M., (c) from 10 P.M. to midnight. An evident hot spot emerges between 8 P.M. and 10 P.M., and disappears afterwards. The location (immediately west of city center) is that of Stadio Meazza, the main soccer arena.

10.4.4 Extraordinary Events

Extraordinary events have a large impact on mobility. They can include big planned rendezvous, such as concerts and sport competitions, which set the destination of many individual trips toward a small area (the event location) where many people concentrate for the event duration; but also unexpected events, either natural or human-generated, such as car accidents or floods, that perturb the regular traffic flow producing (often undesired) concentrations of vehicles in some specific locations. Even if not known a priori, big events can be easily detected by localizing exceptionally high concentrations of presence in specific areas at specific time intervals. The reader may refer to Chapter 8 for a wide presentation of event detection in movement data. Density maps for stationary cars can be used for visual exploratory analysis of abnormal concentrations of presence. A density map can be generated by using a spatial grid and a count of vehicles for each cell for each time interval of interest. For example, in this analysis we use a grid with cells of size $0.5 \text{ km} \times 0.5 \text{ km}$ and compute, for each grid cell and for every interval of two hours of each day, the number of cars that are stationary in the cell.

A sample of the results obtained from Milano2007 is shown in Figure 10.11. The location of the hot spot – the main soccer arena and surrounding parking areas – suggests that a big sport event occurred in such location. It is easy to check that the Milan A.C. versus Bayern Munich quarter-final match of the UEFA Champions League took place in the exact location and time, attended by approximately 77,700 spectators.³ The detection of such hot spots can be easily automatized by an iterative procedure that selects every cell C and time interval h (8–10 P.M. in our case) such that the population of cell C during h is above the 90th percentile in the distribution of the population of (C, h) over the entire observation period.

³ Source http://en.wikipedia.org/wiki/UEFA_Champions_League_2006-2007

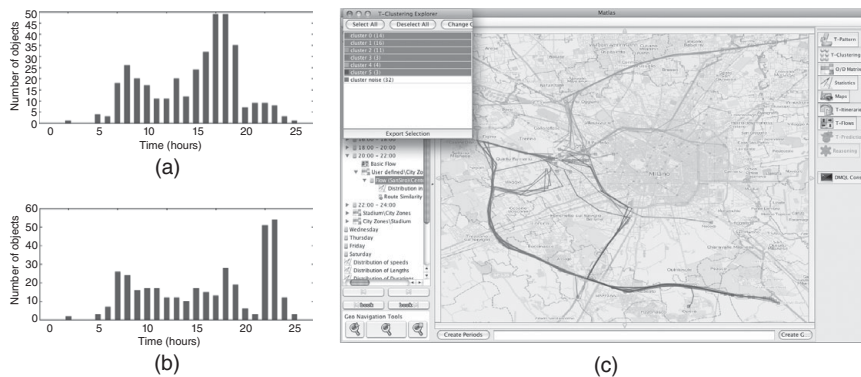


Figure 10.12 Temporal distribution of (a) arrivals to and (b) departures from the arena area: arrivals peak from 5 P.M. through 8 P.M., and departures peak from 10 P.M. through midnight. Arrivals are spread over several hours, while departures occur soon after the end of the match. (c) clusters of trips leaving the arena after the football match. Clusters are highlighted by shades. The largest cluster performs short range trips or take the road ring, either toward the northeast or southwest.

Going deeper in the analysis, we might want to understand when and how attendees reached and left the event location. First, the arrival and departure time of the each car v parked in the arena area during the day is approximated considering, respectively, the ending point of the incoming trajectory and the starting point of the outgoing trajectory of v . The distribution of arrivals and departures during the day is depicted in Figure 10.12a,b. We further analyze the return travels of the attendees after the match, in order to detect the main escape routes – notice that they might differ from the routes planned (for example) by public authorities, either in shape, frequency, or timing. We apply clustering to the trajectories leaving the arena area between 10 P.M. and midnight, obtaining the clusters shown in Figure 10.12. The detected escape routes are relevant information for a mobility manager to enact countermeasures to prevent possible congestion.

10.4.5 Mobility Prediction

The prediction of traffic congestions represents a challenging task for urban mobility managers. The following experiments are aimed at showing how to predict future areas of dense traffic that may lead to traffic congestions. For this task we use the *WhereNext* location prediction algorithm (introduced in Chapter 6) and run the experiment on the Pisa2010 data set, which covers a larger area and a longer temporal interval compared with the Milano2007 data set. This is particularly useful in prediction tasks because the training and test phases use a richer data set. Here, we selected a subset of the entire Pisa2010

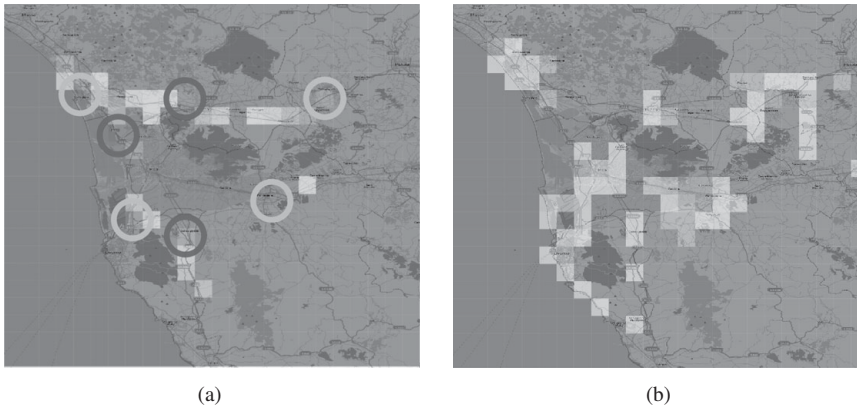


Figure 10.13 Distribution of presence: (a) with predicted trajectories, (b) with the real trajectories. As highlighted on (a), the predictor is able to correctly guess the most dense locations (green circles), though it introduces some false positives (red circles). (See color plate.)

data set that includes trajectories from five working days (from Monday, July 5 to Friday, July 9) restricted to the morning peak hours (8 A.M.–10 A.M.). This selection resulted in about 10,000 trajectories for the training set. Then, we selected, as test set, the trajectories of Monday, July 12th (in the same temporal interval), leading to a total of around 4,000 trajectories. From them, the algorithm was able to predict the next location of about 3,000 trajectories focused on 29 regions. Five of them contain more than 150 trajectories. Scaled to the global number of circulating vehicles this corresponds to about 7,500 vehicles predicted to converge to these areas in the two-hour interval. Figure 10.13 reports the results of the prediction compared with the ground truth obtained by computing the density map of the real GPS trajectories moving during the predicted period.

It is worth pointing out that the interpretation of the predicted zones suggests further, deeper analysis. Indeed, the dense regions do not necessarily indicate traffic problems in those areas. These regions represent dense movements of cars, which can hint the possibility of traffic jams or congestions. Further analysis, focused on these specific areas, are needed to have a more precise indication of possible traffic problems.

10.4.6 Borders of Human Mobility

Here, we address the problem of finding the borders of human mobility at the lower spatial resolution of municipalities or counties. The aim of discovering borders at a mesoscale is motivated by providing decision-support tools for policy makers, capable of suggesting optimal administrative borders for the

government of the territory. We apply social network analysis techniques to mobility data. Our aim is to reach a better understanding of human mobility patterns, using a different perspective based not on the interactions of humans themselves, but rather on the underlying, hidden connections that reside among different places. To do so, we apply community discovery algorithms to the network of geographic areas (i.e., where each node represents a cell or region of movements), with the aim of finding areas that are densely connected by the visits of different users. A community discovery algorithm takes as input a graph and determines a partition of its nodes into communities. Thus, to apply such a method, it is necessary to extract a network model from the mobility data. As in Section 10.2, we adopt census sectors to generalize movement description. In particular, each trajectory is generalized by the sequence of census sectors it crosses during the movement.

Generalized movements can be described by means of a weighted, directed graph $G(V, E)$ as follows. Each census sector is mapped to a vertex $v \in V$. A directed edge $(u, v) \in E$ is placed if there exists at least a movement from u to v , $u, v \in V$. The weight w of the edge corresponds to the number of movements from u and to v . The graph has an edge $(u, v) \in E$ if at least one trip has two consecutive points such that the first is mapped to census sector u and the second to v .

Once the mobility network has been extracted, a community discovery algorithm may be applied to discover groups of nodes, and hence sectors, that can be aggregated. In particular, we adopt here one of the best performing nonoverlapping community discovery algorithms, namely Infomap. Once the communities have been discovered, it is possible to link the nodes back to the geography and define the region covered by each community.

The clustering contains eleven clusters, which are shown in Figure 10.14. The clusters determined by the Infomap algorithm are rendered with distinct colors: the census sectors grouped within the same clusters are drawn with the same color. As a reference for the actual administrative partition, we have plotted the boundary of each town. It is worth noting how the cohesion of the sectors within the same city is preserved. In fact, there are very few episodes of sectors of a city that are scattered among several clusters, and this happens more frequently for rural regions. The zones belonging to the urban centers maintain a strong cohesion. This phenomenon is due to a larger proportion of intracity trips rather than long-range movements: while the main highways are intuitively associated with very dense movement, the local movement within each city is greater than the flow registered in the outer road network. In fact, all the clusters are centered around the big urban regions, which serve as attractors for the surrounding mobility. In the few cases where a sector is associated to a cluster of a different city, it happens that the “misclassified” sector is located near the administrative

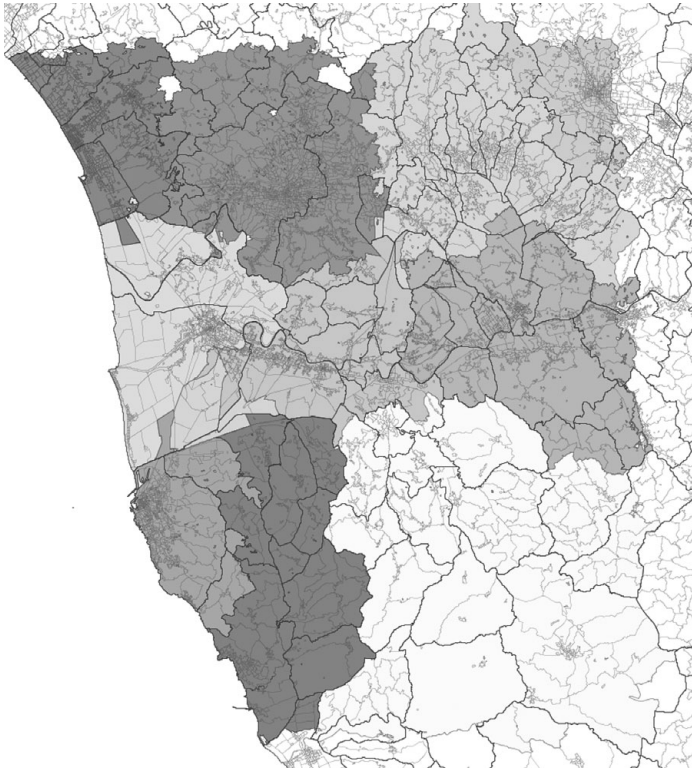


Figure 10.14 Visualization of the clusters determined from the mobility network. As a reference to the existing administrative borders, the perimeter of each town is drawn with a thicker line. The clusters are determined considering the communities derived from Infomap: regions within the same cluster are themed with the same shade.

border of the city. This misclassification is not necessarily a symptom of a error, but rather proves that the sector is attracted by the adjacent city.

Another relevant property of the clustering results is an empirical proof that a single city cannot be considered an “island.” On the contrary, the mobility of a city strictly depends on the mobility of the surrounding towns. In fact, each cluster can be described as an enumeration of a series of cities. Moreover, the cohesion property described above allows the definition of a partition of the territory where each group, that is, each cluster, can be exploited to develop combined mobility policies and planning. Finally, it is important to note that all the clusters present geographically adjacent census. Although this constraint has not been imposed to the community discovery algorithm, the strong cohesion of regions is yielded by the high volume of internal mobility that enables the regions to attract each other. Thus, as discussed earlier, the local short-ranged movements dominate the long-range trips.

10.5 Conclusions

In this chapter we have shown how mobility data mining tools can be very helpful in supporting policy decision makers and transportation planners to answer some very interesting research questions such as the typical access routes to a city, the dynamics of people aggregating and scattering to/from a relevant place, the detection of extraordinary events, and the plotting/modeling of origin destination matrices.

In order to effectively implement and analyze policies for travel demand management (TDM), which constitute one of the main final objectives of transportation science, an increasing amount of awareness has emerged with respect to the need for improved understanding of travel behavior. Indeed, while information such as origin destination matrices that are derived from mobility data mining methods may give a nice overall picture of mobility, nothing is said about the reasons/activities behind these traffic flows. This clearly resulted in a need for travel demand models that embody a realistic representation and understanding of the decision-making processes of individuals and that are responsive to a wider range of transport policy measures. *Activity-based travel analysis* approaches have received attention in recent years as a potential replacement for trip-based approaches because they analyze travel from a theoretical perspective that takes into account the demand for activity participation, interrelationships among trips, and interactions among household members. In the context of the activity-based framework, human activity is a result of actions that are motivated to satisfy needs and desires of the household and its members and travel is undertaken by individuals on their own behalf or as household members to fulfill their needs and desires to participate in these activities. Scientific research related to the field of activity-based modeling is motivated by the importance of improving our understanding of human behavior on the one hand and to use this understanding to provide better predictions of the impact of societal changes and both travel and broader social policies on the future use of transport systems on the other hand. Over the last decade, several of those micro-simulation models of activity-travel demand have become operational.

Current activity-based models are based on either traditional surveys or on full (activity) diaries to model the individual behavior of the agent in the system. Collecting these data either in paper-and-pencil format or by means of computer-aided technology such as small, hand-held computers is a demanding and burdensome task for respondents. The reason for this is that data about the principal choice dimensions underlying the simulation model have to be collected. Typically, a temporal and spatial component always needs to be questioned. And this is exactly where larger GPS and GSM data sets, such as the ones adopted in the research described in this chapter, could be used. However, there is a very long way to go from raw data of individual trajectories to high-level

collective mobility knowledge, implemented in activity-based models capable of supporting the decisions of mobility and transportation managers.

10.6 Bibliographic Notes

The analytical scenarios presented in this chapter are linked to the techniques and methods presented in the previous chapters. In this section we will provide a general list of references to the scientific literature for the reader.

The analytical process for mobility data is based on a specific instance of the knowledge discovery process (Giannotti and Pedreschi, 2008), where analytical methods and algorithms are composed by means of an SQL-based language (Trasarti et al., 2011), introduced in Chapter 7, and integrated in the analytical framework of M-Atlas (Giannotti et al., 2011).

The estimation of travel demand by means of the *four-step model* is presented in Ruiter and Ben-Akiva (1978). The basic concept of this approach is the definition of an origin-destination matrix where rows and columns represent zones of origin and destination respectively and each cell estimates the flows between the two corresponding zones. This model has been extensively used in mobility data management to select, aggregate, and analyze specific traffic flows. Chapter 8 presents an overview of different methods to visualize and interact with OD matrices.

The mining algorithms, trajectory pattern, clustering, and *WhereNext* were introduced in Chapter 6. In this chapter, we adopted a clustering process based on the progressive clustering approach (Rinzivillo et al., 2008), where the clustering analysis is organized in a stepwise process.

The extraction of the borders of human mobility by means of network analytics methods was originally presented in Brockmann et al. (2006), where mobility flows are measured by observing the movements of banknotes. Successive works adopted a similar approach using telephone usage data (Ratti et al., 2010) and GPS data (Rinzivillo et al., 2012). The identification of groups of nodes within a network is performed with a community discovery method. An extensive presentation of the available community discovery methods is given in Coscia et al. (2011). Chapter 15 presents several techniques to analyze mobility by exploiting network analytics methods.

11

Maritime Monitoring

Thomas Devogele, Laurent Etienne, and Cyril Ray

11.1 Maritime Context

The maritime environment still represents unexploited potential for modeling, management, and understanding of mobility data. The environment is diverse, open but partly ruled, and covers a large spectrum of ships, from small sailboats to supertankers, which generally exhibit type-related behaviors. Similarly to terrestrial or aerial domains, several real-time positioning systems, such as the *Automatic Identification System* (AIS), have been developed for keeping track of vessel movements. However, the huge amounts of data provided by these reporting systems are rarely used for knowledge discovery. This chapter aims at discussing different aspects of maritime mobilities understanding. This chapter enables readers to, first, understand the intrinsic behavior of maritime positioning systems and then proposes a methodology to illustrate the different steps leading to trajectory patterns for the understanding of outlier detection.

11.1.1 Maritime Traffic

The maritime environment has a huge impact on the world economy and our everyday lives. Beyond being a space where numerous marine species live, the sea is also a place where human activities (sailing, cruising, fishing, goods transportation, etc.) evolve and increase drastically. For example, world maritime trade of goods volume has doubled since the seventies and reached about 90% of global trade in terms of volume and 70% in terms of value. This ever increasing traffic leads to navigation difficulties and risks in coastal and crowded areas where numerous ships exhibit different movement objectives (sailing, fishing, etc.), which can be conflicting. The disasters and damages caused in the event of sea collisions can pose serious threats to the environment and human lives.

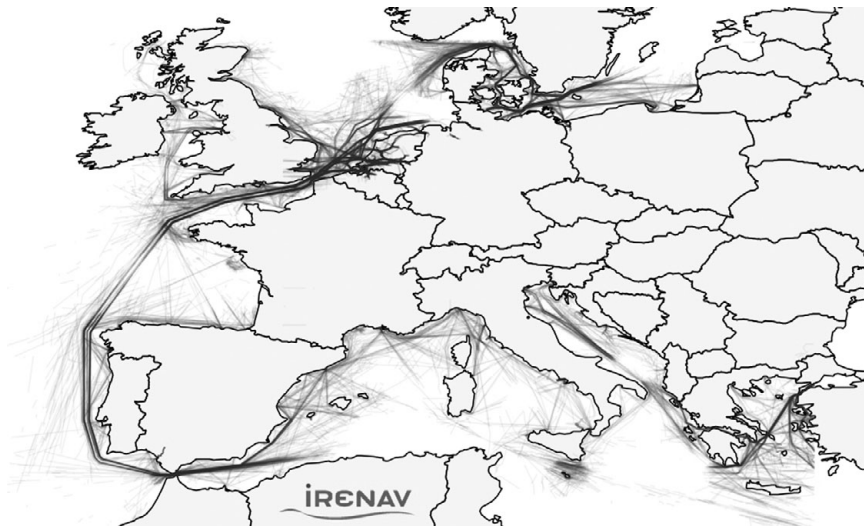


Figure 11.1 Ships' trajectories, density map in Europe during one month (AIS positions, December 2010).

Such disasters and damages often lead to highly negative effects on maritime ecosystems and are threats not only for the important populations of marine protected and endangered species, but also for economic, scientific, and cultural sectors. Safety and security have therefore become a major concern, especially in Europe.

Consideration of this security issue by the International Maritime Organization (IMO) has partly evolved in the last decade from ship design, education, and navigational rules (e.g., International Regulations for Preventing Collisions at Sea: COLREGS), to technical answers for traffic monitoring. Nowadays, ships are fitted out with almost real-time position report systems whose objective is to identify and locate vessels at distance. Figure 11.1 shows, for instance, ships' trajectories obtained through the AIS in Europe during one month.

The maritime environment, represented in Figure 11.1, is diverse and open, but partly ruled. Regulation is ensured by Traffic Separation Schemes (TSS) set up in order to split and regulate the traffic in crowded spaces into traffic-lanes, and by the definition of exclusion areas and Particularly Sensitive Sea Areas (PSSA) the ships have to avoid (e.g., biodiversity areas). Trajectories in such an open space are very typical; ships often behave similarly, traveling in straight lines, leading to visually noticeable trends and patterns. This naturally favors the analysis of aggregated behaviors in order to detect maritime routes, dense areas, evolution of the traffic, and finally, at individual levels, abnormal trajectories and locations.

11.1.2 Maritime Positioning Systems

Two of the most successful systems used in maritime navigation and positioning are the Automatic Radar Plotting Aid (ARPA) and the AIS. Both are used by vessels and Vessel Traffic Services on shore (VTS) in order to facilitate navigation decisions and warn about possible collisions. Vessel traffic services also take advantage of their higher computing and networking resources to store data locally and share them at national and worldwide levels (e.g., program SafeSeaNet of the European Maritime Safety Agency).

Marine Radar

with automatic radar plotting aid tracks vessels using radar contacts. A radar transmitter generates very short pulses of radio waves. When the radio waves of one of these pulses encounter any obstacle, such as a ship, shore line, or big sea waves, part of the radiated energy is reflected and received by the emitting radar. The reflected pulse constitutes a radio echo. The time between the pulse and the echo can be accurately measured and used to calculate the distance between the radar and the echo. The direction of the echo reflects the direction of the pulse. When a target echo appears on a radar screen, an operator plots the relative motion of the echo in order to determine the target's course and speed. The maximum range of an object detected is affected by the height of the radar antenna as well as the height of the object due to the curvature of the earth. In the same way, mountainous sea lines cause blind areas, and objects behind these areas cannot be detected. Bad weather conditions can also affect significantly the effectiveness of radar tracking. Thus, any target should be acquired and confirmed in at least five of ten scans over a period of 2 minutes in order to be brought to the attention of the operator with an identifier and coordinates.

Automatic Identification System

has been recently implemented and made a mandatory standard on commercial and passenger ships. This system, whose objective is to identify and locate vessels at distance, automatically broadcasts location-based information through self organised wireless communications (VHF). AIS usually integrates a transceiver system, a GPS receiver, and other navigational sensors on board, such as a gyrocompass and a rate of turn indicator. An AIS transponder runs in an autonomous and continuous mode, and regularly broadcasts a position report according to the ship's behavior. The information is broadcast, within a range of 35 nautical miles, to surrounding ships and maritime authorities on the ground. There are two different classes of AIS that can be found on ships, search and rescue aircrafts, and base stations on ground: Mandatory AIS (class A) for large vessels and low-cost AIS (class B), which has been introduced for smaller

vessels. Devices from these two classes broadcast information at different time intervals (Table 11.1) and at different ranges (typically 20–40 miles for class A and generally 5–10 miles for class B).

Enhanced Worldwide Positioning Systems

are emerging especially to address drawbacks of both systems, which are complementary but imperfect. On one hand, ARPA is useful to detect and track vessels that might not have AIS devices onboard. On the other hand, it brings limited information and cannot identify a mobile object, and its coverage includes blind areas. The automatic identification system is useful to obtain more complete information, but devices are not available on all ships and data can be falsified. The most important issue that guides evolutions concerns the limited tracking range of both systems, which is insufficient to follow ships engaged on international journeys. Satellite communications systems are going to be more intensively employed, in particular to enhance or replace the AIS. For instance, Long-Range Identification and Tracking (LRIT) reports vessels' positions to their flag administrations at least four times a day. Satellite-based AIS-monitoring service (S-AIS) uses satellite communications to broadcast AIS information. Nowadays, position reports for European coasts reach almost 1.5 million positions per day (about 72,000 ships). The ever-increasing data flows provided by this evolution are going to emphasize issues on maritime data integration, fusion, filtering, processing, and analysis.

Location-Based Data

While radar data are limited to a tuple composed of an identifier, a position, and a related time, the automatic identification system broadcasts a wide range of richer information. Information systems onboard or in vessel traffic services generally merge AIS and radar positions into a single accurate one. When a ship is not fitted with an AIS (typically small boats), the reported information for data analysis is only limited to the aforementioned tuple. From our perspective, this does not impact the data mining process and therefore motivate an analysis focusing on the AIS data more easily accessible. Transmitted AIS data come from twenty-seven different messages, each providing specific information either related to the behavior of the AIS system or to a ship's locations and characteristics. Positioning data defines point-based trajectories describing 2D routes on the sea surface. That is, an ordered series of locations expressed in WGS84 format (latitude λ , longitude φ , time t) of a given mobile object with t indicating the timestamp of the location (λ, φ) . Among all the received data, meaningful information that can be considered in a purpose of movement discovery and understanding can be classified in the three following categories:

Table 11.1 *AIS Shipborne Mobile Equipment Reporting Intervals.*

Ship's Dynamic Conditions – AIS Class A	Freq.
Ship at anchor or moored and not moving faster than 3 knots	3 m
Ship at anchor or moored and moving faster than 3 knots	10 s
Speed between 0 and 14 knots	10 s
Speed between 0 and 14 knots and changing course	$3 \frac{1}{3}$ s
Speed between 14 and 23 knots	6 s
Speed between 14 and 23 knots and changing course	2 s
Speed over 23 knots	2 s
Speed over 23 knots and changing course	2 s

- *Static*: MMSI number (Maritime Mobile Service Identity: a unique ID), name, type, International Maritime Organization code, call sign, dimension.
- *Dynamic*: Position (longitude, latitude), time, speed, heading, course over ground (COG), rate of turn (ROT), navigational status.
- *Trajectory-based*: Destination, estimated time of arrival (ETA), draught, dangerousness.

Quality of data is variable and depends, first, on the quality of the AIS device itself and the way it implements algorithms and protocols. Therefore, data like coordinates and speed can be more or less accurate. Longitude and latitude are normally given in 1/10,000 minute that should give 0.18 m. However, considering this quality factor and intrinsic behaviour of GPS, the International Maritime Organization only considers an accuracy of 10 m. The quality also depends on people onboard. Indeed, some data, such as MMSI, name, destination, or navigational status, are manually set and possibly wrong. Contextual information associated with geographic positions helps to understand ships' behaviors according to space, time, destination, and ships' types although they require error-detection and filtering processes.

Space and Time Gaps

Time is not part of position reports, as the AIS was initially designed for real-time purpose only. Each received message has to be timestamped by the receiver's clock. While it communicates on a regular basis, the automatic identification system does not send these position reports continuously. Transponders broadcast data to surrounding listeners at different sampling rates according to ships' behaviors. Table 11.1 presents sampling rates for AIS class A. Class B devices behave in a similar way but at different sampling rates. This variation of time intervals is very specific to the maritime domain and can vary from 2 seconds for a fast-moving ship to several minutes when anchored.

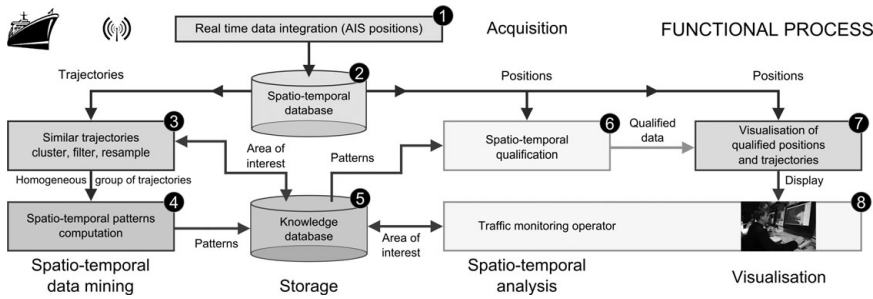


Figure 11.2 Data mining and trajectory qualification process.

The range covered by all VTS on shore is limited and coverage areas might not overlap everywhere. In such a context, the observation of the maritime traffic at a given time leads to a partial view due to space and time gaps. These received positions will mostly not correspond to the selected times for snapshots analysis (e.g., a ship communicated its position 10 seconds before the analysis time). This implies one should consider time intervals and the definition of trajectories for a successful analysis and understanding of the ships' behaviors. Let us note that these large and variable gaps between two position reports will affect significantly the way trajectories can be computed.

11.2 A Monitoring System Based on Data-Mining Processes

The increase of maritime location-based information brings opportunities for knowledge discovery on movement behaviors at sea over a long period of time. This section shows how maritime data can be processed and analyzed in order to qualify a given position or trajectory with computed patterns. This process allows one, for instance, to detect outliers including real-time traffic monitoring. It is based on data-mining principles presented in other chapters, especially Chapter 6. The methodology postulates that *normal* moving objects following a same itinerary at sea behave in a similar optimised way. Such a behavior illustrated in Figure 11.1 helps to compute accurate trajectory patterns.

Figure 11.2 presents the functional process used to extract spatio-temporal patterns from spatio-temporal databases and qualify ship positions and trajectories. First, an acquisition step (Step 1 in Figure 11.2) integrates AIS raw data from several monitoring systems into a structured spatio-temporal database (STDB). In this database, zones of interest (ZOI) define either an origin or a destination of a trip. Each identified ZOI is associated with its surface and linked to its neighbors (and stored in the spatio-temporal database). Then, trajectories are clustered (Step 3) according to their itineraries in order to obtain homogeneous groups of trajectories (HGT). A statistical analysis of these clusters gives the median trajectory of each cluster and spatio-temporal intervals around

them (Step 4). Median trajectories and intervals are combined together to define the spatio-temporal pattern of HGTs. These patterns are stored in a knowledge database (Step 5). They can be used either for geovisual analyses or to qualify in real-time ship positions and trajectories (Step 6).

This functional process has been experimented on and used in different contexts: real-time tracking of sailing races and maritime navigation in the coastal area of Brest: Processing and analysis of AIS raw data from Aegean, North, and East China seas, and from aggregated real-time data flows from NATO countries. A maritime case study based on passenger ships in Bay of Brest, France, illustrates, throughout this chapter, this qualification process for safety purpose (a sample data set is available at the ChoroChronos repository.¹

11.2.1 Platform, Database Model

This functional process (Figure 11.2) relies on a generic and scalable information system that has been designed for real-time monitoring and spatio-temporal analysis of different types of moving objects at sea. So far, the underlying platform developed is a Java-based computing system based on a PostgreSQL/PostGIS spatial database for data manipulation and storage. It has been designed with four tiers of client-server architecture, and organized through a distributed data and processing model. The information system is based on different functions depicted in Figure 11.2, as follows:

- Real-time integration of positioning information (Step 1),
- Spatio-temporal data mining (Steps 3–5),
- Spatio-temporal analysis (Step 6),
- Web-based visualisation (Step 7).

The data model set up in the PostGIS database relies on the aforementioned classification of AIS messages: static, dynamic, and trajectory based (Table 11.2). Table AISPositions stores all the dynamic position reports of ships. Table AISShips contains the static information, especially the ship's type, which can be used later to cluster trajectories of similar ships (e.g., cargo, passenger ships, sailing ships). Table AISTrips is used to store ships' trips, based on information such as a ship's destination and the type of goods it is carrying. In addition to these tables that contain raw information, some derived data can be added to the database. Table Trajectories is obtained from positions of the Table AISPositions and from AISTrips in order to link position reports of a same ship together and to reconstruct its path (Table 11.2, field trajectories.shape). As Table AISTrips gives information about ships' destinations, these destinations can be extracted as zones of interest (ZOI) and

¹ <http://www.chorochronos.org>

Table 11.2 *Database Model.*

Table	Description
Data Provided by AIS	
AISPositions	Position reports of each ship with additional dynamic information. <u>MMSI</u> (numeric), Time (timestamp), Heading (numeric), Speed (numeric), COG (numeric), ROT (numeric), Coordinates (geometry), Status (text)
AISShips	Static information about ships. <u>MMSI</u> , OMI_Number (numeric), Name (text), Callsign (text), Type (text), Length (numeric), Width (numeric)
AISTrips	Trajectory-based information. <u>MMSI</u> , Draught (numeric), Danger (Boolean), Destination (text), ETA (timestamp), ReportedTime (timestamp)
Derived Data Added to the Model	
Trajectories	Trajectories extracted from raw data. <u>MMSI</u> , BeginningTime (timestamp), EndTime (timestamp), Shape (geometry)
Zones	Zones of interest (ZOI). <u>ZID</u> (numeric), Name (text), Shape (geometry)
Itineraries	Itineraries between ZOI. <u>IID</u> (numeric), StartZoneID (numeric), EndZoneID (numeric)

stored in a new table, *Zones*. The zones of interest can also be manually defined by an operator according to various criteria such as regulations (waiting areas, traffic channels, restricted areas), geography (obstacles, isthmuses, straits, inlets), and economy (shops, loading sites, ports, fishing areas). These zones of interest, represented as spatial zones, can be connected together to define a *zone graph* in order to analyze ships' mobility and describe their itineraries (Table *Itineraries*).

For richer analysis, taking geographic information into account might also be of interest. The database could therefore include a large set of tables obtained from official S-57 vector charts that contain different kind of objects useful for spatial analysis:

- Points of interest: buoys, shipwrecks, containers at sea, etc.
- Lines of interest: coastlines, path, channels, crossing lines, etc.
- Zones of interest: oil spills, ports, restricted areas, PSSA, etc.

The zone graph of the Bay of Brest is illustrated in Figure 11.3b. The numerous dots shown in Figure 11.3a represent positions of ships. An itinerary *I* is

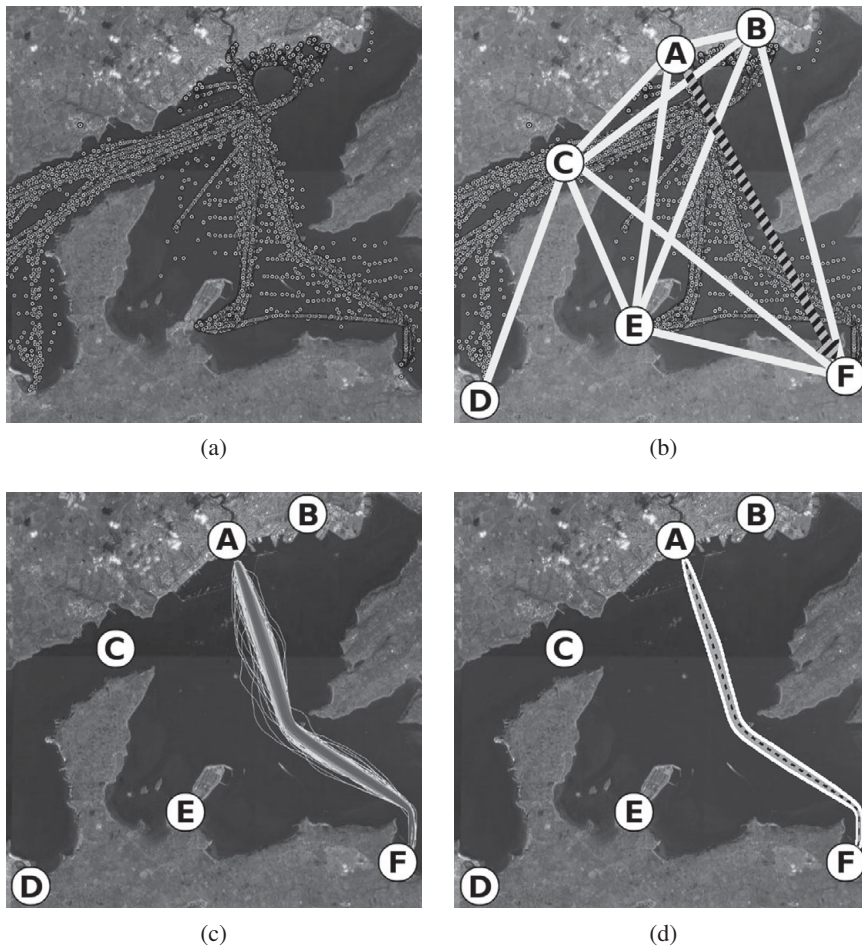


Figure 11.3 From raw data to trajectory pattern (Bay of Brest).

an arc between two zones of the graph. Figure 11.3c,d, illustrating trajectory patterns, will be presented in Sections 11.2.3 and 11.2.4.

11.2.2 From Raw Positions to Trajectories

As shown in Figure 11.3a, the numerous position reports of ships can be put together in order to build a trajectory and address point-based query limits. Point-based queries (strictly based on raw positions) exhibit two limits. First, a computing limit such as point-based spatial queries is very expensive in terms of computing cost. Second, it reaches a spatial limit as queries are applied on reported locations provided by the AIS (a ship passing through a narrow restricted area can report positions on both sides, due to AIS behavior and

sampling frequency, even if the trajectory of the ship crosses the zone). Therefore, it is difficult to identify whether a ship went through a narrow passage, entered a restricted area, or computed exact minimal distances to the coast (this requires interpolation and additional computing costs).

Trajectory features are required to query more correctly and efficiently the AIS database. Further, it allows for distance computation based on polylines instead of raw positions, route definitions, trajectory comparisons, and clear identification of passage through an area or a line. Due to the computing limit, the number of positions for each trajectory must be reduced using a filtering algorithm in order to apply spatial operators and functions to efficiently answer end users' questions. This trajectories production stage is located between Steps 2 and 3 of the data-mining and qualification process (Figure 11.2).

Many approaches can be considered to define a maritime trajectory and build such trajectories from a sequence of AIS positions. Let's consider the time-ordered sequence of all AIS positions of a given ship defined by $S = \{p_0, \dots, p_n\}$. A trajectory T of this ship can be defined as a subsequence of S so that $T \subset S \wedge T = (p_b, \dots, p_j, \dots, p_e)$ where p_b stands for the beginning position of the trajectory and p_e for the ending one.

The main matter consists in selecting the beginning and ending positions from S in order to create a set of trajectories. These particular positions (considered as stops) can be identified by the mobile object cinematic (e.g., null speed), its spatial position (inside a zone of interest), or the position report sampling rate (transmission gaps). As the position reports from the AIS itself are not regular and depend on the ship's behavior (Table 11.1), a simple time and spatial threshold might not be sufficient to properly detect gaps defining the beginning and ending positions and split sequences of raw positions into trajectories. So, dynamic spatial (δs) and temporal (δt) thresholds should be derived from the enriched information provided by the AIS, which contains heading H_p , speed S_p , acceleration A_p , and rate of turn R_p indications. Such an approach can rely on the number of missed frame(s) allowed (n_{mf}) and the reporting intervals expected by the AIS device onboard (Table 11.1) to define the time (δt) and spatial (δs) thresholds. The next position of a trajectory should be transmitted within δt and should be located within a maximum δs distance. Otherwise, the last position is considered as a stop and future positions of the sequence S will be associated with a new trajectory.

Another way to define these stops within a sequence of positions is to rely on zones of interest, which can be identified in cartographic information or manually defined by an expert (see Section 11.2.1). This inevitably changes the semantic of the trajectory with respect to the previous method. However, such an approach is suited better to the analysis of maritime mobilities as ships always have a small number of well-defined origins and destinations (harbor, mooring, or waiting area). For a more automatic process, such areas can also be created

automatically using a density analysis. In this context, a beginning position of a trajectory is a position that is inside a zone Z and whose next position of the trajectory is outside this zone. An ending position of a trajectory is a position that is inside a zone Z and whose previous position of the trajectory is outside this zone. Therefore, the sequence S of position of a given ship can then be split into a subset of trajectories $\Gamma = \{T_0, \dots, T_N\}$ such as $\Gamma \subseteq S$.

Once the positions are assigned to trajectories, a filtering process selects the key positions of a given trajectory. A position is considered as a key position when either the speed or the direction changes significantly. The other positions can be removed.

The algorithm initially introduced by Douglas and Peucker in 1973 is relevant as it performs well on typical straight trajectories of vessels. The principles of the original algorithm are as follows. The start and end points of a given polyline are connected by a straight line segment. Perpendicular offsets for all intervening end points of segments are then calculated from this segment, and the point with the highest offset is identified. If the offset of this point is less than the tolerance distance, then the straight line segment is considered adequate for representing the line in a simplified form. Otherwise, this point is selected, and the line is subdivided at this point of maximum offset. The selection procedure is then recursively applied to the two parts of the polyline until the tolerance criteria is satisfied. Selected points are finally chained to produce a simplified line.

This simplification algorithm for trajectory filtering could be adapted in order to be more efficient. Conversely to Meratnia and By (2004), who used Euclidean Distance between points at a same time, the Haversine distance can be used. This distance is the shortest distance (d_s) between two points measured along a path on the surface of a sphere. The perpendicular distance is therefore derived as a spatio-temporal distance d_{ST} and is as follows:

$$d_{ST}(T_i, T_j, t) = d_s(p_i(t) - p_j(t))$$

The spatio-temporal distances between position p_i of the trajectory T_j , and position p'_i of the interpolated trajectory T'_j taken at a same time (relative time from the departure) are computed. Let us note that these spatio-temporal distances are influenced by the speed and the direction of the mobile object. A tolerance distance should be defined appropriately. According to the GPS position accuracy, a tolerance of 10 meters is acceptable.

In order to exemplify this filtering process, three vessel trajectories have been selected for illustration purposes. The first trajectory concerns a passenger boat called *Bindy*, whose trajectory is smooth and speed is regular. The second trajectory is the one of a port pilot ship in the harbor of La Rochelle. This trajectory is very sinuous, and several loops appeared. The third trajectory is composed of long straight polylines made by the cargo ship *AB Valencia*.

Table 11.3 Results for Filtering Process with 10 m Tolerance.

Vessel	Trajectory Duration	% of Position Kept	% of Length Kept (km)
<i>Bindy</i>	28 m 01s	14.0% (32/229)	99.91% (11.284/11.294)
Port pilot boat	1 h 07 m 36 s	21.7% (122/562)	99.82% (24.846/24.892)
<i>AB Valencia</i>	7 h 04 m 20 s	12.0% (279/2316)	99.98% (175.07/175.109)

Table 11.3 summarizes the filtering result. One can note that their lengths are very close. This leads to a filtering process where more than 80% of the received positions can be filtered. The performance of the filtering process is likely to increase for large ships and decrease for small ships due to the intrinsic characteristics of their navigation.

11.2.3 Trajectory Clustering Process

Once the trajectory concept is defined, different trajectory clustering techniques can be used to determine homogeneous groups of trajectories. Some of them are presented in Chapter 6. Another technique based on the zone graph and itineraries can be used to extract clusters from trajectories following the same itinerary I . This set is called a homogeneous group of trajectories (HGT).

The first selection criterion of this approach is based on static information such as the type of mobile objects; this information is provided by AIS messages (Table 11.2). The second selection criterion is a geographical one. The first position of the trajectory (p_b) must be the only one within the departure zone (Z_D) of the itinerary, and the last position of the trajectory (p_e) must be the only one within the arrival zone (Z_A) of the itinerary. Taking into account the frequency of trajectory samples and the speed of the mobile object, trajectories that cross a zone of the graph should have at least one position within this zone. The last selection criterion used is time. Some moving objects can follow this itinerary periodically. These different trajectories can be distinguished using a time interval. Finally, the trajectory should not intersect any other zone of the graph G_Z that does not belong to the itinerary I . All valid trajectories previously extracted from the STDB compose the HGT to be analyzed.

Figure 11.3c illustrates the extraction of the HGT of 500 passenger ships' trajectories following the itinerary between Brest and Naval Academy (arc A-F of G_Z). Some density differences can be noticed on this HGT. This HGT

highlights the outlier trajectories represented in light grey (outside the darker grey dense area).

11.2.4 Spatio-Temporal Pattern Mining

Once the HGT clusters have been extracted and filtered, the next step aims at defining the pattern followed by most trajectories of each HGT. The main matter of this mining task is to deduce the median trajectory followed by the HGT and the spatial and temporal density distribution. Studies on several trajectory clusters showed that these data do not belong to any particular statistical distribution. Gaps between mean and median values are important. Density around these values changes frequently. For example, for the time dimension, it's easier for mobile objects to arrive late than early. For this kind of ordered set of data in descriptive statics, box plot series are very useful to describe the evolution of data according times. Box plots, proposed by John Tukey in 1977, graphically describe groups of numerical data through five important sample percentiles:

- The sample minimum (smallest observation),
- The lower quartile or the 1st decile,
- The median,
- The upper quartile or the 9th decile, and
- The sample maximum (largest observation).

In our maritime context, data lower than the first decile or higher than the ninth decile are considered as outliers. The idea is to enhance box plot series to produce 2D plus time patterns. Each pattern summarizes a cluster of trajectories (HGT) thanks to the median value, and the symmetry and dispersion of the data set.

First of all, a synthetic median trajectory (T_m) can be computed using an iterative refinement technique similar to the k -means algorithm. A trajectory from the HGT is chosen as initial T_m . T_m is an ordered set of positions: P_{m_i} . To optimize this algorithm, a trajectory with length and time duration close to median length and median time duration has to be chosen as initial T_m . Then, all positions of each trajectory of this HGT are assigned to one position of T_m using a matching process. Amongst existing algorithms, dynamic time warping (DTW) or Fréchet matching can be employed. They can align trajectories' positions in order to minimize the sum of the spatial distances between matched positions of two trajectories (DTW), or minimize the maximum distance between matched positions (Fréchet). They also take into account the temporal order of the positions of trajectories. Figure 11.4 illustrates the clusters of matched positions (Cm_{p_i}) between positions of trajectories of the HGT and the P_{m_i} in black. Light grey thin lines show links between matched positions.

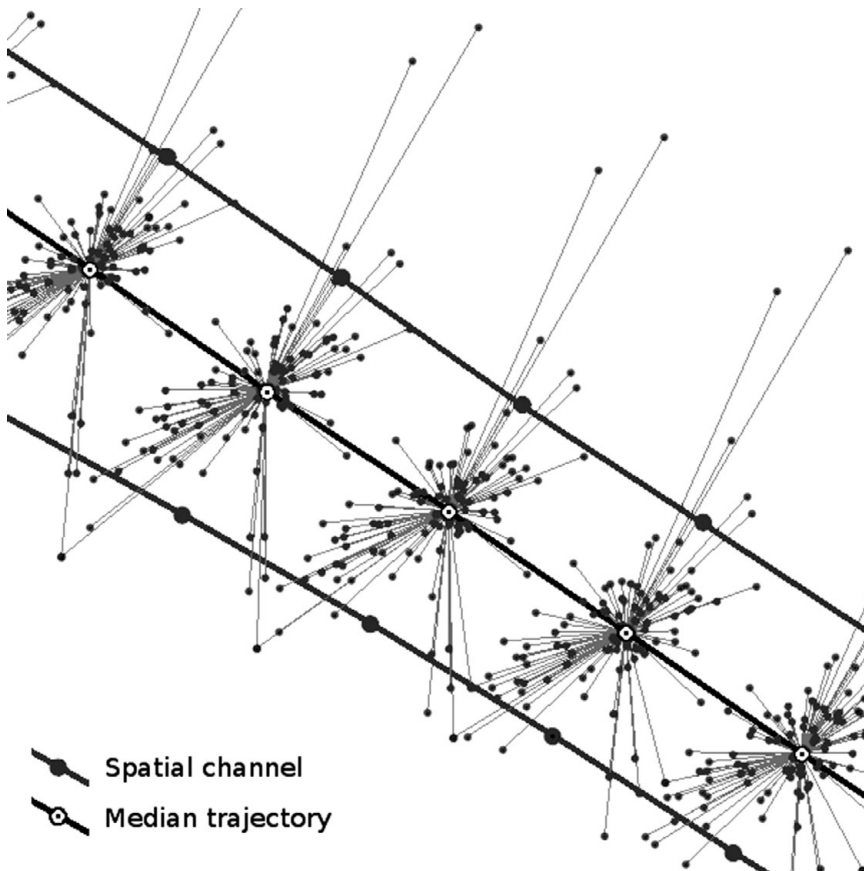


Figure 11.4 Clusters of positions and spatial pattern.

Once every position is matched, the coordinate and the timestamp of P_{m_i} are updated, by computation of median X (\tilde{X}), median Y (\tilde{Y}), and median timestamp (\tilde{t}). A medoid approach is also possible but requires more time for similar results. Assignment and update steps are repeated until the distance (Fréchet distance or average distance) between two consecutive points reaches a minimal threshold value.

As the studied mobile objects move in an open area, some of them can move away from the main trajectory. *Normal* temporal or spatial deviations must be distinguished from outliers. Two channels are computed to distinguish the spatio-temporal outliers. First, the spatial channel is defined. Once the median trajectory is computed, a statistical density analysis can be performed on every cluster of matched positions ($\mathcal{C}m_{p_i}$). These clusters are split into two subsets of positions, \mathcal{L}_{p_i} (left sided) and \mathcal{R}_{p_i} (right sided), according to their side to the median position P_{m_i} using the P_{m_i} heading. Then, spatial distances between positions from \mathcal{L}_{p_i} and the P_{m_i} are computed. After a statistical analysis, the

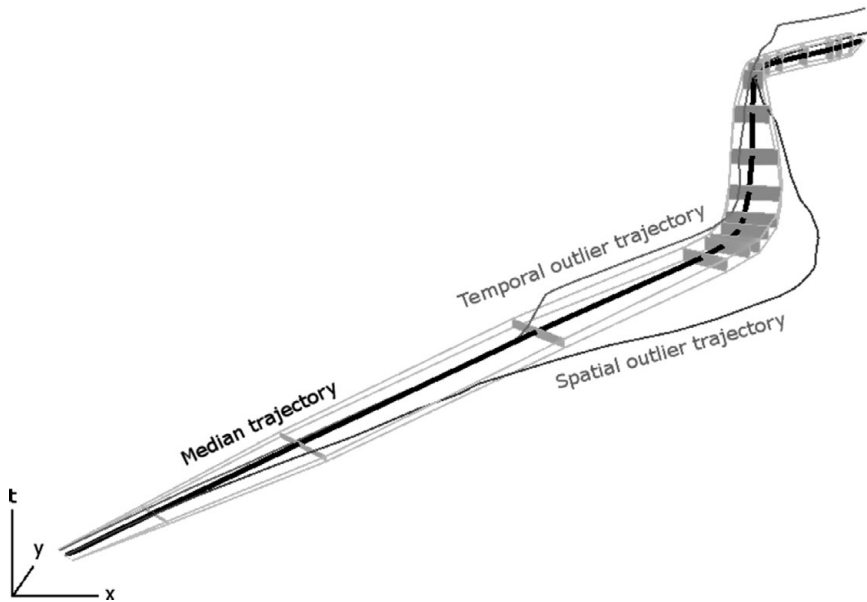


Figure 11.5 3D spatio-temporal pattern of an itinerary and outlier trajectories.

ninth decile is chosen as left limit of the channel for this $\mathcal{C}m_{p_i}$. The same process is computed to define the right limit of $\mathcal{C}m_{p_i}$. The left (right) limits are linked according to the time to define the left (right) limit of the spatial channel. Figure 11.4 presents the limits of the spatial channel in dark grey. Some positions are visually outside this channel and can be defined as outliers. In the same way, the temporal channel is defined. Positions of $\mathcal{C}m_{p_i}$ inside the spatial channel are split into two subsets, late sided and early sided, according to the difference between relative timestamps of positions and on the median matched position. The early and the late limits are computed to define the temporal channel of each $\mathcal{C}m_{p_i}$. Positions outside the spatial channel are not taken into account because these parts of trajectories including these positions could be shortcuts or detours. Spatial and temporal channels at each relative time can be combined to create the spatio-temporal channel, which is then stored in the knowledge database. Figure 11.3d illustrates the spatio-temporal channel of the HGT (Figure 11.3c) extracted from zone A to F of the zone graph (Figure 11.3b). The spatial and temporal widths change. For example, for the straight part of the pattern, the spatial width is bigger than the curved part's width.

The spatio-temporal pattern defines five different zones (usual position zone, right outlier zone, left outlier zone, late outlier zone, and early outlier zone) for each relative time. This spatio-temporal pattern (median trajectory plus spatio-temporal channel) is a 2D+t enhancement of the box plot concept. It can be illustrated in 3D using the Z axis to represent the relative time as shown in Figure 11.5. The median trajectory is plotted in black; the usual 3D zones

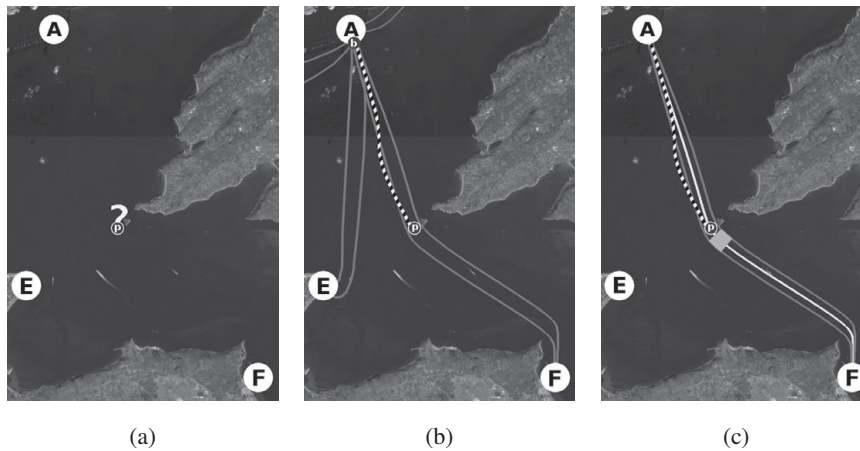


Figure 11.6 Outlier detection.

are the grey boxes defined for some key positions of median trajectory linked together. The limits of the spatio-temporal channel are outlined in light grey. Two examples of outliers' trajectories (dark grey) getting out of the spatio-temporal channel are presented in this figure. The first one presents a late temporal outlier trajectory. The second one highlights a spatial outlier trajectory (right sided). This spatio-temporal pattern must be computed for each HGT. As new positions are frequently acquired by the system, this spatio-temporal channel could be improved by updating it periodically.

Quality of the set of patterns depends on the precision of the ZOI graph and the set of mobile object types. This quality could be verified if the spatial and temporal distributions of positions of each Cm_{p_i} are unimodal. If several modes appear, a new analysis can be carried out to split the set of mobile objects according to types or to add new ZOI in the graph.

11.2.5 Outlier Detection

For each cluster, the associated spatio-temporal pattern splits the set of trajectory positions in the outlier position group and the usual position group. For a new vessel position, this knowledge could be useful to detect and to qualify this position. Therefore, this section suggests that we combine the knowledge database and the production database to obtain an inductive database and to detect the outlier positions in real time. Let's consider a new position p received. The position qualification process is decomposed into three steps (illustrated in Figure 11.6):

- Trajectory extraction from the last ZOI encountered by the mobile object to p ,

- Matching process between this trajectory and the median trajectories of a pattern, and
- Spatio-temporal comparison between p and selected pattern.

In the first step, the database is queried to select the start position from the trajectory. This position is the last one of the mobile object inside the surface of one ZOI. Positions between p and this departure position are timestamp ordered to define a trajectory path. This last one does not link two ZOIs, consequently, it is called a partial trajectory (T_p). In Figure 11.6, the last ZOI is A and the start position is (b) . The partial trajectory is the dashed polyline. The second step must match T_p with part of a median trajectory. This matching can be done according to:

- The type of the moving object,
- The geometry of the partial trajectory,
- The set of median trajectories from the departure ZOI, and
- Information about the course of the moving object to destination.

Unfortunately, information about the destination is often false or unknown, so only the type of vessel and geometry properties can be used. In order to match two linear geometries, the Fréchet discrete distance is selected as it allows partial matching processes. Fréchet distance gives the maximal distance between two lines. The Fréchet discrete distance applied to two discrete trajectories (ordered set of points) can match trajectories together, preserving order of their points. Alt and Godau (1995) demonstrate the advantage of this measure. Devogele in 2002 proposed to enhance this distance in order to compute the distance between a line and a homologous part of another line. This partial discrete Fréchet distance ($dPdF$) is very useful to match a trajectory where only the departure is known. Thanks to this $dPdF$, the distance between T_p and median trajectories from the same departure ZOI can be computed. Only the spatio-temporal patterns for the same type of this object are taken into account. T_p can be partially matched with one median trajectory (\tilde{T}) where $dPdF(T_p; \tilde{T})$ is lower than the $dPdF$ with other median trajectories plus a threshold. $dPdF(T_p; \tilde{T})$ must also be less than a maximal value. In the example, the distance between T_p and two median trajectories (from ZOI A to E and from A to F) are computed. The second distance is the lowest, so T_p is matched with median trajectories from A to F .

Finally, the position p could be qualified according to the selected pattern. The relative time of p from departure ZOI is employed to infer the spatio-temporal channel from the knowledge database. The 3D channel is cut at this timestamp and the space is split into five areas (right, left, usual, late, and early). Qualification of p is given by the area that contains p . For example, the spatial channel of the matched pattern is limited with dark grey lines and the usual area at the relative time of p is the grey area. Position p is an outlier and is located in

the late area, so this object can be spatially qualified as “inside the channel” but temporally as “running behind schedule.” Such real-time analysis methods can be used to predict the destination and time of arrival of the ship once an itinerary has been matched, and if the position is normal. The destination prediction can be higher than 90%. In the same way, the confidence interval of time of arrival could be the width of temporal channel at the arrival.

11.3 Conclusions

The maritime environment represents an increasing potential in terms of modeling, management, and understanding of mobility data. The environment is typical and recently several real-time positioning systems, such as the Automatic Identification System (AIS), have been developed for keeping track of vessel movements. This chapter outlines different aspects of maritime mobilities understanding through pattern discovery and analysis of ships’ trajectories. Underlying issues concern in particular trajectory modeling problems, trajectory querying and simplification, similarity functions, classification and clustering algorithms, and knowledge discovery (trends, unusual behaviors, and event detection).

Assuming that moving objects at sea that are following the same itinerary behave in a similar way (considered as the normality), this chapter illustrates the different steps leading to outlier detection. The suggested methodology considers several steps. First, the data flow provided by the automatic identification systems is managed in structured spatio-temporal databases. Then, data mining processes are used to extract trajectories (vessels of the same type) and spatio-temporal patterns between two zones of interest (an origin, a destination). Each pattern includes a median trajectory and a spatio-temporal channel that describes the dispersion of the set of trajectories. Such trajectory patterns are meaningful to understand maritime traffic and detect outlier positions in real time. Indeed, each new position (partial trajectory) can be spatially and temporally qualified according to spatial and temporal criteria. For end users monitoring maritime traffic, such real-time qualification of positions and trajectories is tied with triggers automatically executed when a new outlier is detected, and adapted geovisualisation process are essential for safety purposes.

While complete, the suggested methodology still leaves several additional challenges. First, cartographic information and environmental data such as currents, tides, and winds that affect ships’ movements could be taken into account for further improvements. Many other algorithmic approaches for trajectory representation and reconstruction can be considered for other knowledge discovery objectives. Interactive and adaptive geovisualisation is also of interest. Another challenge concerns new itineraries. Many factors can influence ships’ behavior, leading to the apparition of new itineraries. The proposed approach handles such

regular trajectories as outliers. An adaptive process should be therefore considered in order to detect a new pattern and possibility remove an outdated one. Finally, the approach described could be applied or extended to other kinds of moving objects evolving in open spaces, especially those having 3D trajectories (e.g., underwater vehicles or planes that behave quite similarly to ships).

11.4 Bibliographic Notes

Several maritime projects worked to enhance the tracking and monitoring of vessels. This is portrayed for example in [MarNIS \(2009\)](#). These monitoring systems use ARPA and AIS sensors as input. [Bole et al. \(2012\)](#) describe the ARPA system in detail. In a similar way, the Association of Marine Aids to Navigation and Lighthouse Authorities describes the AIS in [IALA \(2004\)](#). These new tracking and monitoring systems are parts of e-Navigation defined by the International Maritime Organization in [IMO \(2008\)](#). e-Navigation relies on Electronic Navigation Chart (ENC), defined by the International Hydrographic Organization in [IHO \(2000\)](#).

If the reader needs additional information about some special technical points of this chapter, several articles can be read. For the filtering part, [Meratnia and de By \(2004\)](#) serves as the base for the filtering process presented in this chapter. For the similarity measure between trajectories, Fréchet distance has been selected. [Alt and Godau \(1995\)](#) explains why this measure is better for this kind of data. [Devogele \(2002\)](#) describes the algorithm for discrete partial Fréchet distance. Matching process based on dynamic time warping is also possible; see [Sakoe and Chiba \(1978\)](#). Results are very similar but this later process can align only whole trajectories. Some details about our architecture are introduced in [Bertrand et al. \(2007\)](#). Finally, [Etienne et al. \(2012\)](#) details the clustering process and the spatio-temporal pattern based on box plot. This representation is defined in [Tukey \(1977\)](#).

12

Air Traffic Analysis

**Christophe Hurter, Gennady Andrienko, Natalia Andrienko,
Ralf Hartmut Güting, and Mahmoud Sakr**

12.1 Introduction

The goal of air traffic control (ATC) is to maximize both safety and capacity, so as to accept all flights without compromising the life of the passengers or creating delays. Because air traffic is expected to double by 2030, new visualizations and analysis tools have to be developed to maintain and further improve the safety level. To do so, air traffic practitioners analyze data from the ATC activity. These multidimensional data include aircraft trajectories (3D location plus time), flight routes (ordered sequences of spatio-temporal points that represent planned routes), and meteorological data. In this chapter, we detail the relevant tasks of ATC practitioners and demonstrate recent visualization and query methods to fulfill them.

The special properties of ATC data propose new challenges and, at the same time, new opportunities of data analysis. The semantics of the data are rich because they includes the third dimension (altitude), which can be used to discover salient events such as takeoffs and landings. More semantics can be added by augmenting background data such as the traffic network and the meteorological data. ATC data sets are characterized by their large sizes, adding more challenges to the analysis. Trajectory analysis is difficult due to the data set size and to the fact that it contains many errors and uncertainties. One day's traffic over France contains about 20,000 trajectories (>1 million records). Recording is done in a periodic manner (in our database: a radar plot, per aircraft, every 4 minutes), but a plot can be missed, or have erroneous data because of physical problems that occur at the time of recording.

This chapter demonstrates recent works of trajectory analysis. Three techniques are demonstrated: direct manipulation, visual analytics, and moving object database queries. Direct manipulation visually represents the raw trajectories, and allows the user to efficiently explore them and highlight interesting

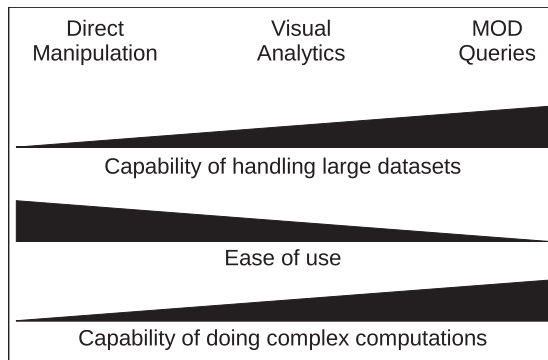


Figure 12.1 Factors of choosing among the methods of trajectory analysis.

subsets using convenient views and simple mouse interaction. Visual analytics provide a rich tool box of data transformations and visualizations that help a human analyst exploring complex movement events in the data. Moving object database (MOD) defines query operators accessible to the user through textual query languages. They are able to perform complex computations over large data sets efficiently. According to the analysis task, the experience of the human doing the analysis, and the data set size, any of these three analysis methods (or a combination of them) can be chosen. This is illustrated in Figure 12.1.

Direct manipulation is good for having a first look at the data. It is intuitive to use. Visual analytics provides more sophisticated transformations and aggregations, and thus it is able to process larger data sets, and to perform deeper analysis. Human expertise is, however, a deciding factor for good analysis results. MOD queries are mandatory for complex computations such as pattern matching. The user must however know exactly what he or she is looking for, and how to precisely describe it in terms of the MOD query language.

Throughout this chapter we will demonstrate each of these analysis methods, in the context of real tasks and using a real data set. The motivation for the analysis and the description of the data set are presented in Sections 12.2 and 12.3 respectively. Direct manipulation is demonstrated in Section 12.4. Section 12.5 demonstrates the use of visual analytics to explore movement events, such as landings and takeoffs, and to derive useful statistics from them. Finally Section 12.6 explains a MOD query operator that is able to match complex patterns in ATC data, such as missed approaches and stepwise descents.

12.2 Motivation

Aircraft trajectories are monitored and recorded by ground radar. They are displayed in real time on radar screens. This data is essential for *air traffic*

controllers, in order to maintain a safe distance between aircraft and to optimize traffic fluidity (reduce flight time, noise, and fuel consumption). Our goal in this chapter is not to provide tools for real-time usages, but rather to detail offline tools that analyze recorded trajectories in more depth. Without this real-time constraint, ATC practitioners can investigate, in more detail, recorded trajectories and therefore extract relevant information and perform three main tasks: improve safety, optimize traffic, and monitor environmental considerations.

Improving safety can be detailed as:

1. Analyzing and understanding past conflicts (when two aircraft fail to meet minimum safety distance) and then improving safety with feedback from past experience,
2. Analyzing the accuracy of data provided by ground radar with probe trajectory comparison (i.e., with GPS tracking and radar test plots), and
3. Filtering and extracting trajectories in order to reuse them for air traffic controllers' training simulations.

Traffic optimization can be detailed as:

1. Devising new air space organization and flight routes to handle traffic increase,
2. Studying profitability (i.e., number of aircraft on a specific flight route per day, number of aircraft that actually land at a specific airport, etc.),
3. Calculating the metrics from the traffic: traffic density, spacing quality (mean distance between aircraft), number of holding loops, number of rectilinear trajectories (trajectories that are close to the shortest path from departure to arrival), etc., and
4. Measuring the activity of each airport: number of takeoffs and landings per hour, etc.

Finally, environmental considerations can be detailed as:

1. Comparing trajectories with environmental considerations (fuel consumption, noise pollution, vertical profile comparison),
2. Detecting missed approach trajectories (which produce noise), lap training landings (pilots who train to take off, fly around the air field and land; lap training landings consume a lot of fuel), and
3. Counting continuous descending aircraft (since these aircraft maintain a constant descent rate, they reduce their fuel consumption).

This list is not exhaustive but it gives the main tasks that ATC practitioners perform. These tasks highlight the need for powerful tools to analyze aircraft trajectories.

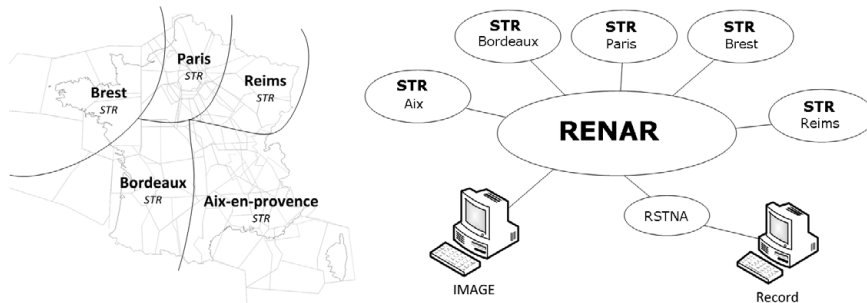


Figure 12.2 IMAGE network with STRs.

12.3 Data Set Description

In this section, we detail the different steps required to produce data sets of aircraft trajectories provided by the IMAGE system. In France, ground radars send aircraft positions through the RENAR (Réseau de la Navigation Aérienne) network. Due to network bandwidth limitations we cannot route all raw radar information toward a single network access point to record it. Therefore, we use the French IMAGE system. IMAGE is a system that aims to gather aircraft positions from all French-controlled areas. Its goal is neither to monitor aircraft activity nor to optimize traffic flow, but to give a general view of the traffic (communication purposes). The IMAGE system is connected to the five French STRs (Système de Traitement Radar), one in each en-route control center (Figure 12.2). STR systems receive aircraft information from different radar sources and calculate an estimated position for each monitored aircraft (using tracking and smoothing algorithms). The IMAGE system helps to reduce ground radar sources to only five data sources, and enables us to retrieve aircraft positions over France within the RENAR network.

Merging the five data sources raises lots of issues: unique aircraft identifiers, overlapping areas, time stamps, and sampling rates. First, each STR sends the aircraft position with an identifier from 1 to 1,023. Since more than 1,023 aircraft can fly over France at the same time, we extend this identifier to a 16-bit format and reassign a unique identifier to every trajectory. To do so, we use a spatio-temporal frame filtering to assign a new unique identifier to each trajectory: each radar plot that has the same identifier within a 600 second time frame within an area of a 200-km (100 Nm, nautical miles) radius (which corresponds to a 12-minute straight flight at high altitude) belongs to the same trajectory. At this stage, trajectories with less than three plots are removed and no trajectory has the same identifier.

Secondly, we merge all the five new, reassigned, radar records into one file. The main issue is to connect trajectories that were recorded by different STR sources. To do so, we resample all the data to ensure that every record has

the same regular time stamp. Then we set up the following merging parameters: when two trajectories overlap, they merge if the overlapping points have the same altitude (less than 600 m/2,000 ft, which corresponds to 1-minute descent), and close position (less than 9 km/5 Nm, which corresponds to the minimal safety distance).

The properties of the data set we use in this chapter are typical to any IMAGE data set. We use a data set with 17,851 flight trajectories over France during one day (Friday, February 22, 2008) consisting of 427,651 records. The trajectories, shown in Figure 12.3, include flights of passenger, cargo, and private airplanes and helicopters. The temporal resolution of the data mostly varies from 1 to 3 minutes, although larger time gaps (up to 5 minutes) also occur. 3,000 trajectories (60,000 records, 16%) fly over France per day without landing.

12.4 Direct Manipulation of Trajectories

Formulating trajectory queries is difficult for two reasons. First, they are often only specifiable with visual features (straight lines or general shapes). Second, users often explore the queries as much as they explore the data: in the course of exploration, users discover that the set of features they thought relevant has to be adapted, either because they were false, or because they cannot find how to query them efficiently. Furthermore, trajectories are numerous and tangled: one day's traffic over France, for example, represents some 20,000 trajectories. When dealing with trajectories, users must perform dynamic requests (response time < 100 ms) on a large multidimensional data set (>1 million data), which contains many errors and uncertainties. The problem we address in this section is to find a way to express these queries simply and accurately, given the constraints of size and uncertainty of the data sets. As a solution, the visualization and direct manipulation of trajectories proposes efficient interaction features. Direct manipulation was introduced by Ben Shneiderman in 1983 within the context of office applications and the virtual desktop metaphor. This term has been extended to human-computer interaction paradigms. The intention is to allow users to directly manipulate objects presented to them, using actions that correspond to the physical world (e.g., grasp, move objects, etc.).

In the following sections, we first describe direct manipulation requirements for trajectory exploration, then we detail an implementation instance, and finally we give one scenario of usage.

12.4.1 Design Requirements for Trajectory Exploration

Based on trajectory data set characteristics, we extracted the following design requirements to achieve the visual exploration of trajectories:

1. **View configuration:** The system must permit the customization of views so as to offer multiple means of understanding and visually querying the data. It should allow for a change of mapping between data and visual dimensions. The system should also provide smooth transitions between visual configurations. Hence, the user will be able to visually track patterns between different view configurations.
2. **View organization and navigation:** The system must also permit the display of multiple views. The user must be able to visually compare different visual configurations of the data set. This can be done with a matrix scatterplot or juxtaposed views.
3. **View filtering:** The system must allow the user to filter out trajectories and then reduce cluttering.
4. **Trajectory selections and Boolean operations:** The system must enable the user to select trajectories and combine them in order to perform complex queries. Some systems allow multiple selections sometimes called “layers.” Users can combine layers with Boolean operation by applying an “and” operation when they try to group differently selected trajectories.

12.4.2 Implementation Instance: FromDaDy

We have developed *FromDaDy* (Hurter et al., 2009) (which stands for “From Data to Display”), a visualization tool that tackles the challenge of representing and interacting with numerous trajectories (several million trajectories composed of up to 10 million points). *FromDaDy* employs a simple paradigm to explore multidimensional data based on scatterplots, brushing, “pick and drop,” juxtaposed views, and rapid visual configuration. Together with a finely tuned mix between design customization and simple interaction, users can filter, remove, and add trajectories in an incremental manner until they extract a set of relevant data, thus formulating complex queries.

12.4.3 Views Organization and Navigation

A *FromDaDy* session starts with a view displaying all the data in one scatterplot. The visualization employs a default visual configuration, for example, the mapping between data dimensions and visual variables. The view is inside a window, and occupies a cell in a virtual infinite grid that extends from the four sides of the cell. The user can configure the two axes of each scatterplot and use other visual variables such as color and line width to display data set dimensions. For instance, in Figure 12.3, the user attached the data set field latitude to the y axis, and the field longitude to the x axis. The user also chose to use the altitude to color trajectory sections, showing, low altitudes in green and high altitudes in blue.

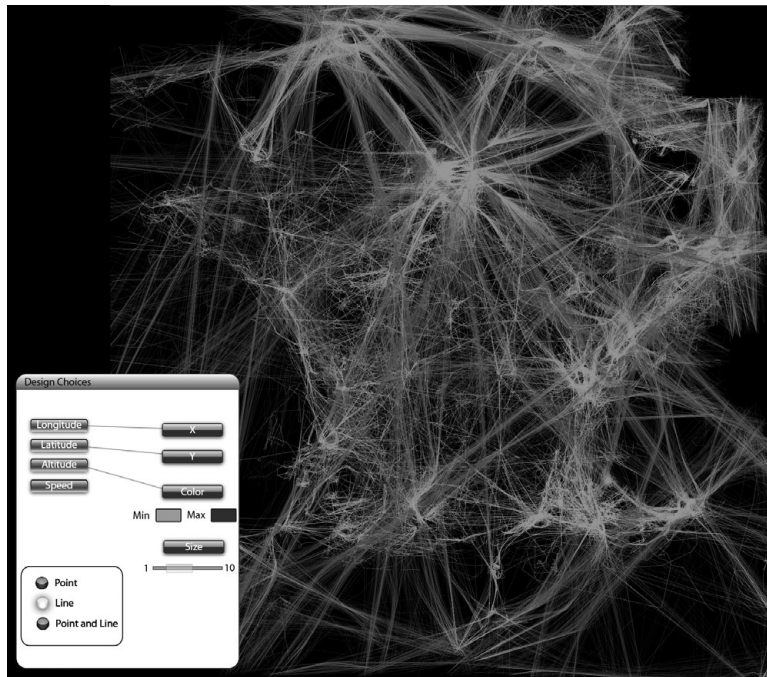


Figure 12.3 One day's record of traffic over France. The color gradient from green to blue represents the ascending altitude of aircraft (green being the lowest and blue the highest altitude). The French coastline is apparent here in terms of pleasure flights by light aircraft and the straight blue lines represent high altitude flight routes. A user interface shows the data set fields and the defined visual configuration. (See color plate.)

12.4.4 Trajectory Manipulation

We have implemented a simple and efficient direct manipulation technique: trajectory brush, pick, and drop. The user selects a subset of the data set by means of a brushing technique. Brushing is an interaction that allows the user to “brush” graphical entities, using a size-configurable or shape-configurable area controlled by the mouse pointer. Each trajectory touched by this area is selected, and becomes gray. The selection can be modified by further brush strokes, or by removing parts of it with brush strokes in the “erase” mode. The display shows a brush trail, so that the user can see and remember more easily how the selection was made. The combination of fast switching between the add/erase mode, trajectory visualization, rapid size-setting, and cursor-centered zooming allows for fast and incremental selection.

Then the user can pick brushed trajectories by hitting the space bar. The user extracts previously selected data from the current scatterplot and attaches them to the mouse pointer so they appear in a “fly-over” view (transparent background).

When the user hits the space bar for the second time, a drop occurs in the view under the cursor. If the view under the mouse pointer is empty, the software creates a new scatterplot with the selected data. If the user presses the space bar while moving over a view-containing data, FromDaDy adds the selected data to this scatterplot. Although it resembles a regular drag and drop operation, we prefer to use the term “pick and drop,” because the data are removed from the previous view and attached to the cursor even if the space bar is released. The user can also destroy a view if the brush selects all the trajectories and the user picks them.

12.4.5 Brush Pick and Drop

The fundamentally new aspect of FromDaDy, compared with existing visualization systems, is to enable users to spread data across views. Within FromDaDy, there is a single line displayed per trajectory: trajectories are not duplicated, but are spread across views. The advantage of this technique is multifold. It enables the user to remove data from a view (and drop it on to the destination view). The fly-over view enables the user to rapidly decide if the revealed data (previously hidden by the picked data) are interesting. Second, it makes it possible to build a data subset incrementally. In this case, the user can immediately assess the quality of the selection by seeing it in the “fly-over” view. Furthermore, by removing data from the first view, the user makes it less cluttered, and this makes it easier for him or her to pick and drop more trajectories.

Another advantage of the brush, pick, and drop paradigm is that this interaction helps the user to perform complex Boolean operations: “I want the trajectories that go into this area but not the ones that are too high and only those that are faster than a given minimum speed.” A seminal previous work uses containers (also called layers) to cluster trajectories and explicitly applies Boolean operations to combine them. Even with an astute interface, Boolean operations are cumbersome to produce, because results are difficult to foresee. FromDaDy overcomes this drawback, since all the operations of the interaction paradigm (brush, pick, and drop) implicitly perform Boolean operations. Removing trajectories corresponds to an XOR operation and dropping trajectories corresponds to an ADD operation. The following examples illustrate the union (AND), intersection (OR), and negation (NOT) Boolean operations. With these three basic operations the user can perform all kinds of Boolean operations: AND, OR, NOT, XOR, and so on.

In Figure 12.4, users want to select trajectories that pass through region A or through region B. They just have to brush the two desired regions and pick/drop the selected tracks into a new view. The resulting view contains their query, and the previous view contains the negation of the query.

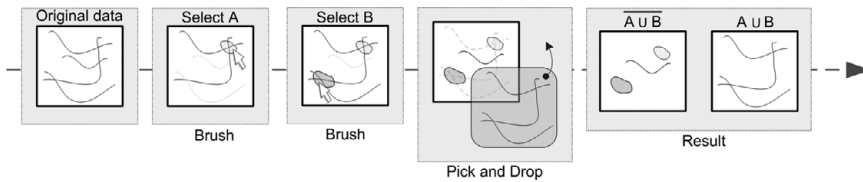


Figure 12.4 Union Boolean operation.

12.4.6 Example of Usage

In this scenario, we use one day of recordings of aircraft trajectories over France. In this data set, a unique and incremental identifier is assigned to each trajectory. The first trajectory of the day has the number 0', the next one has the number 1', and so on. Figure 12.5 shows an abstract visualization of this data set. The x screen axis shows the time of each radar plot and the y screen axis shows the aircraft's identifier. Since these identifiers are incremental over the day, the resulting visualization shows a noticeable continuous shape, in which each horizontal line represents the duration of one flight. The slope of the shape indicates the traffic increase during the day (due to the incrementally assigned identifiers). Hence, the traffic notably increases at 5 A.M. and decreases at 10 P.M., as reflected in the change of slope. The width of this shape indicates the average flight duration in the data set: it is about 2.5 hours, which represents the average time taken to cross France. But some aircraft have longer trajectory durations. The user brushes these long trails (the ones that come out of the curved shape). When visualizing them with a latitude (y screen) and longitude (x screen) visual configuration, the user discovers a figure eight-shaped trajectory. This trajectory covers 6 hours and performs 11 loops. After further investigation, it is found that it corresponds to a military supply plane.

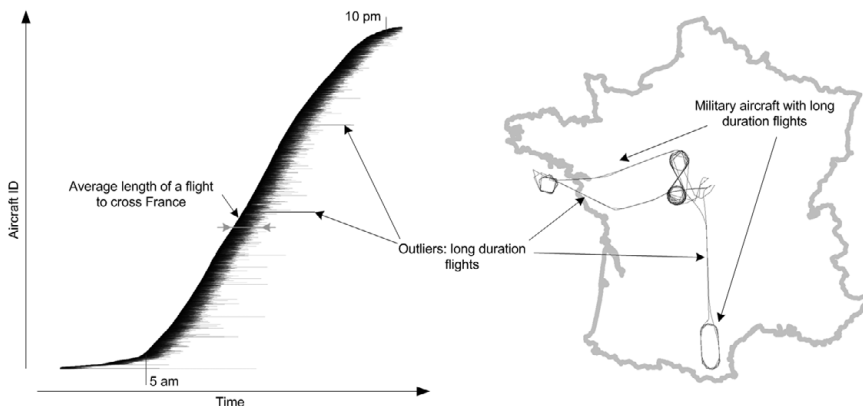


Figure 12.5 Detection of supply planes with an abstract visualization.

This data exploration has been done with a visualization tool. The user would have also been able to perform the same extraction with a textual tool, such as SQL queries. The only difference is that a textual tool would not have led the user to the idea of exploring long flight duration in order to extract military aircraft. Only with the incremental trajectory exploration can the user discover the valid requests for this data set. In a sense, the user explores the data set, and at the same time, explores the request to perform. Even if this process is efficient, the direct manipulation cannot be automatic. Analysts need tools to enhance their exploration capabilities. Therefore, extended work will be presented in the following sections.

12.5 Event Extraction

There is a class of problems where analysts need to determine places in which movement events (m-events) of a certain type repeatedly occur and then use these places in further analysis. The relevant places can only be delineated by processing movement data, that is, there is no predefined set of places (e.g., compartments of a territory division) from which the analyst can select places of interest. The relevant places may have arbitrary shapes and sizes and irregular spatial distribution. They may even overlap in space; therefore, approaches based on dividing the territory into nonoverlapping areas, as in [Andrienko and Andrienko \(2011\)](#), are not appropriate. In this section, we analyze one-day record of aircraft trajectory with a visual analytics procedure for place-centered analysis of mobility data ([Andrienko et al., 2011c](#)). The procedure consists of four steps: (1) visually supported extraction of relevant m-events, (2) finding and delineating significant places on the basis of interactive clustering of the m-events according to different attributes, (3) spatio-temporal aggregation of the m-events and movement data by the defined places or pairs of places and time intervals; (4) analysis of the aggregated data for studying the spatio-temporal patterns of event occurrences and/or connections between the places.

12.5.1 Analyzing Flight Dynamics in France

We shall apply our visual analytics procedure to ATC data with the following goals: (1) Identify the airports in use. (2) Investigate the temporal dynamics of the flights to and from the airports (i.e., landings and takeoffs). (3) Investigate the connections among the airports, the intensity of the flights between them, and their distribution over a day.

It may not be obvious to the reader why the airport areas need to be determined from the data instead of using the official airport boundaries, which should be known. The problem is the low temporal resolution of the data. For many flights, the first recorded positions lie outside the boundaries of the origin airports and/or

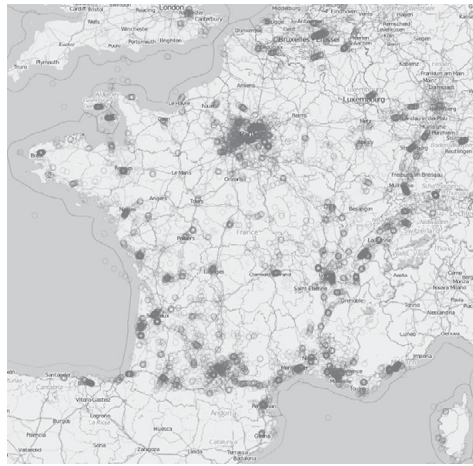
the last recorded positions are not within the boundaries of the destination airports. Therefore, to refer the flights to their origin and destination airports, it is necessary to build sufficiently large areas around the airports that would include the available first and last points. It is not known in advance how large the areas need to be and what geometrical shapes are appropriate.

Our approach to defining the areas is based on the background knowledge that airplanes typically land and take off in similar directions, which are determined by the orientation of the airport runways. We extract the available last positions of the aircraft that landed and first positions of those that took off and cluster them by spatial positions and movement directions using a density-based clustering method, Optics (Ankerst et al., 1999), with similarity measures designed for spatio-temporal events (Andrienko et al., 2011c). As a result, points lying outside or even quite far from the airports are grouped together with the points lying within the airport boundaries if they correspond to landings or takeoffs with similar directions. The airport “catchment” areas are built as buffers around these clusters. The areas can be verified using the known positions of the airports: they must be within the areas.

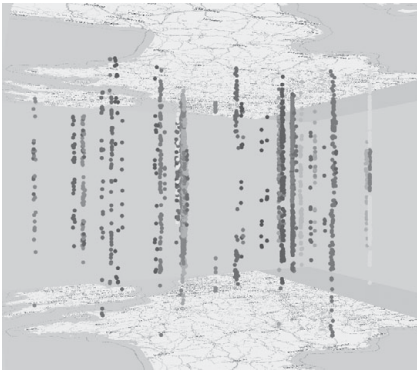
Not always do starts and ends of trajectories correspond to takeoffs and landings. The radar observation data also contain parts of transit trajectories that just pass over France as well as flights going outside France and those coming to France from abroad. Real takeoffs and landings must be distilled from the available starts and ends of the recorded tracks. To extract the landings, we use the following query condition: the altitude is less than 1 km in the last 5 minutes of the trajectory. From each trajectory that has such points, we extract the last point as an m-event representing the landing (Figure 12.6a). In the second step of the analysis, we cluster the landing events by the spatial positions and directions (SD) using the thresholds of 1 km and 30 degrees, respectively. The resulting SD-clusters are presented in the space-time cube in Figure 12.6b; the noise (events not having sufficient counts of SD-neighbors) is excluded. The colors represent different clusters. The vertical alignments of points correspond to the airports where multiple landings took place during the day.

An interesting pattern can be observed in the area of Nice in the southeast of France. There are two SD-clusters of landings, yellow and green; their points make a column on the right in the cube. The green cluster appears as an intrusion inside the yellow one. This means that the landing direction changed in this area twice during the day due to a change of wind direction (aircraft take off and land facing the wind). The map fragment in Figure 12.6c shows that the yellow cluster contains landings from the southwest and the green cluster landings from the northeast. The blue lines in Figure 12.6 show the last 10-minute fragments of the respective trajectories and reflect the mandatory landing directions.

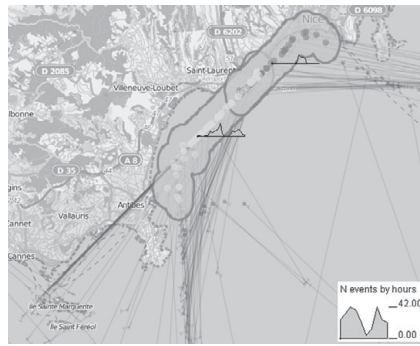
The observation of the direction changes gives us an idea that the temporal patterns of landings should be investigated not by airports only but by airports



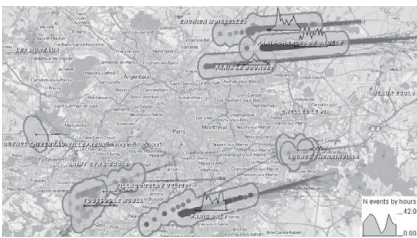
(a)



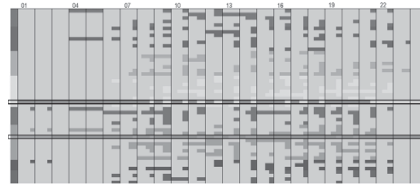
(b)



(c)



(d)



(e)

Figure 12.6 Event extraction results. (a) The positions of the landing events extracted from the flight data are drawn with 50% opacity. (b) The space-time cube shows the landing events clustered by spatial positions and directions. (c) The yellow and green dots represent two SD-clusters of landings in the airport of Nice. The time diagrams show the dynamics of the landings from two directions. (d) The time diagrams show the dynamics of landings in the airports of Paris. (e) The flight distribution between the airports by hourly intervals. Highlighted are rows for the connections Marseille–Paris (yellow) and Paris–Marseille (orange). (See color plate.)

and landing directions. Therefore, we build 500-meter spatial buffers around the SD-clusters, as shown in Figure 12.6c. For an analysis by airports, irrespective of the directions, we would do a second stage of clustering (after excluding the noise) by only the spatial positions of the events and then build buffers around the resulting spatial clusters.

In the third step of the analysis procedure, we aggregate the landing m-events in space by the buffers and in time by 1-hour intervals. In the fourth step, we visualize the resulting time series by temporal diagrams positioned on the map display; two of them can be seen in the map fragment in Figure 12.6c. They show that the aircraft landed in the airport of Nice from the southwest almost all times except for an interval in the middle of the day, when the landing direction changed to the opposite. The exact times and values are displayed when the mouse cursor points on an area.

Figure 12.6d presents the map with the temporal diagrams for the Paris region. We can see that the Orly airport and the northern runway of the Charles de Gaulle airport have clear peaks in the morning and in the evening. It is a typical pattern for airline hubs: a short period of time, during which many flights arrive and take off, maximizes the number of possible connections. The southern runway of the Charles de Gaulle airport is used with almost constant intensity during the day. The remaining airports are used much less intensively and mostly in the afternoon.

So far we have considered only the landings. To investigate the takeoffs, we repeat the procedure. To extract the takeoff events in the first step, we use the query condition that the altitude must be less than 1 km at the beginning of the trajectory. The remainder of the procedure is similar to that for the landings.

To investigate the connections among the airports, we need to define the airport areas so that they include both the takeoff and the landing events. We join the sets of the takeoff and landing events, which have been previously filtered by removing the noise after the SD-clustering. Then we apply clustering by spatial positions, to unite the clusters of takeoffs and landings in different directions occurring at the same airports. We build spatial buffers around the spatial clusters to obtain the airport areas. In the third step (spatio-temporal aggregation), we aggregate the trajectories by pairs of places (airport areas) and time intervals (1-hour length). We use only those trajectories that have both takeoff and landing events. As a result, we obtain aggregate flows (vectors) with respective hourly time series and totals of flight counts.

To investigate the aggregates (Step 4: analysis of the aggregated data), we visualize the total counts on a flow map. The aggregate flows are shown by directed arrows with the widths proportional to the flight counts. By interactive filtering, we hide minor flows (less than 5 flights) and focus on the short-distance flows (less than 100 km distance). We see that there are quite many flights connecting close airports, particularly in Paris. As explained by a domain

expert, a part of them are flights without passengers used for relocating aircraft between big airports, such as Charles de Gaulle and Orly. Short-distance flows between small airports correspond to training and leisure flights of private pilots. Focusing on the long-distance flows (100 km and more) reveals a mostly radial connectivity scheme with a center in Paris.

To investigate the temporal dynamics of the flows, we use the table display as shown in Figure 12.6e. The columns of the table correspond to the hourly time intervals and the rows to the flows. The lengths of the colored bar segments in the cells are proportional to the flight counts for the respective flows and intervals. The colors correspond to the eight compass directions. The table view is linked to the flow map. Thus, clicking on the vectors connecting Paris Orly and Marseille on the map, we get two rows highlighted. The yellow one corresponds to the northwestern direction, that is, from Marseille to Paris, and the orange one to the opposite direction, from Paris to Marseille. There are one or two flights from Marseille to Paris every hour in the intervals 07–14h and 15–18h and three flights per hour from 22h to midnight. The traffic in the opposite direction has a different profile: three flights per hour from midnight till 02h and several flights in the morning, at noon, and in the evening. The complementary link from the table view to the map can be used to locate flows with particular dynamics.

12.5.2 Validation of the Findings

First, to assess the validity of the extracted areas of takeoffs and landings, we compared them with the known positions of the airports and found that the areas include the airports. Furthermore, the areas have elongated shapes (Figure 12.6d) whose spatial orientations coincide with the orientations of the runways of the respective airports. Next, the results of data aggregation by the areas (i.e., counts of takeoffs, landings, and flights between airports) correspond very well to the common knowledge about the sizes and connectivity of the French cities and airports. The discovered patterns have been also checked and interpreted by a domain expert who confirmed their plausibility.

12.6 Complex Pattern Extraction Using a Moving Object Database System

Moving object database systems are another good candidate for air traffic analysis. This section demonstrates a concrete example of using the `SECONDO MOD` system in order to extract complex spatio-temporal patterns from the flight trajectories. The task is to extract the *missed approach* and the *stepwise descent* events that occurred in the ATC data set described in Section 12.3. The *spatio-temporal pattern (STP) algebra* in `SECONDO` brings a generic set of query operations accessible through the `SECONDO` query languages to let the user express arbitrarily

complex patterns and efficiently match them on large moving objects databases. This algebra defines the STP predicate, which is the main tool we are going to illustrate in the section. To get the most out of this section, please first read the chapter about moving object database systems (Chapter 3), especially the part explaining the `SECONDO` query languages.

12.6.1 The Spatio-Temporal Pattern Predicate

A traditional select-from-where query is formulated based on a single predicate given in the where clause. Such a query scheme is not sufficient when dealing with moving objects. A moving object has a lifetime and it fulfills several predicates during it. In many applications it is required to find the objects that fulfill a set of predicates in a certain temporal order. In ATC, for instance, it is required to detect landing procedures such as *go-around*, *missed approach*, and *touch-and-go*. Each of these procedures consists of a set of well-defined steps that have to be implemented by the pilot in a certain temporal order. Extracting these situations from the aircraft trajectories requires a query tool that accepts such descriptions and matches them against the trajectories. Here comes the *spatio-temporal pattern predicate* to extend the traditional select-from-where scheme, and let the user formulate such queries.

Essentially the STP predicate is a pair $\langle P, C \rangle$, where P is a set of predicates and C is a set of temporal order constraints on their fulfillment. Given a tuple u , for example, representing one flight trajectory, the STP predicate yields true iff u fulfills all the predicates in P in the temporal order asserted by all the constraints in C . Consider for example the *missed approach* procedure. It can be described by three predicates: aircraft comes close to destination, aircraft descends to a height of less than 1,000 m, and aircraft climbs. Temporally, the third predicate must be fulfilled after the second predicate, and both of them must be fulfilled during the fulfillment time of the first predicate. Let's have a quick illustration of how this *missed approach* query is expressed using the `SECONDO` executable language:

```
... stpattern[
  Close: distance(.Position, .Destination) < 5000.0,
  Down: ((.AltitudeDerivative < 0.0) and (.Altitude < 1000.0)),
  Up: .AltitudeDerivative > 0.0;
  stconstraint("Close", "Down", vec("abba", "a.bba", "baba")),
  stconstraint("Close", "Up", vec("abba", "aba.b", "abab")),
  stconstraint("Down", "Up", vec("aabb", "aa.bb"))] ...
```

where `stpattern` is the `SECONDO` operator denoting the STP predicate. For simplicity, we omit the query parts before and after the `stpattern` operator and denote them by three dots. The `stpattern` predicate is placed in the query

as a filter condition within the `SECONDO filter` operator. Here it receives a tuple with the schema:

```
tuple[Id: int, Position: mpoint, Altitude: mreal, Destination: point,
      AltitudeDerivative: mreal],
```

where `Position` represents the (lon, lat) of the aircraft and the `Altitude` is separately represented. This is because `SECONDO` does not contain types for 3D moving points. The `Destination` is precomputed as the final (lon, lat) of the trajectory, and `AltitudeDerivative` is precomputed as the derivative of `Altitude`. The three predicates constituting P have the aliases `Close`, `Down`, and `Up`. The `Close` predicate asserts that the aircraft is close (within 5 km) to its destination airport. Note that this is a *time-dependent predicate*, also called *lifted predicate*. That is, the result of such a predicate is a time-dependent boolean *mbool*. It is false whenever the aircraft is far from its destination, and true whenever the aircraft is close to destination. Similarly, `Down` and `Up` are time-dependent predicates. Actually, this is how the `stpattern` operator is able to check the temporal constraints on the predicate fulfillment, because an *mbool* contains information about when the predicate was fulfilled. The STP predicate expects that P be a set of time-dependent predicates, each of which is a mapping $tuple \rightarrow mbool$. The aliases of the time-dependent predicates make it possible to refer to them in the temporal constraints.

The set of temporal constraints C in this example consists of the three temporal constraints denoted as `stconstraint`. Each of them asserts a temporal relation between two predicates forming a pair in P . The temporal relation is expressed by the `vec` operator. Each of the terms inside the `vec` operator specifies a relation between two time intervals. The start and the end points of the first interval are denoted `aa`, and those of the second interval are denoted `bb`. The order of the symbols describes the temporal order of the four end points. The dot symbol denotes the equality. For example, the relation `aa.bb` between the intervals i_1, i_2 denotes the order: $((i_1.t_1 < i_1.t_2) \wedge (i_1.t_2 = i_2.t_1) \wedge (i_2.t_1 < i_2.t_2))$. The temporal relation expressed by the `vec` operator is the disjunction of its components. A temporal constraint between two predicates p_i, p_j is fulfilled iff there exists an interval on which p_i is fulfilled, and another interval on which p_j is fulfilled, and the two intervals fulfill any of the interval relations in the constraint. For the STP predicate to be fulfilled, all the temporal constraints in C must be fulfilled.

Formally, given $P = \{p_1, \dots, p_m\}$ a set of time-dependent predicates, $C = \{c_1, \dots, c_n\}$ a set of constraints, and a tuple u , let $p_i(u)$ denote the evaluation of p_i for the tuple u (i.e., $p_i(u)$ is of type *mbool*). Let $[p_i(u)]_j$ denote the j^{th} time interval on which $p_i(u)$ is true. The evaluation of the STP predicate $\langle P, C \rangle$ for the tuple u is true iff: $\exists j_1..j_m$ such that the set of time intervals $[p_1(u)]_{j_1}..[p_m(u)]_{j_m}$ fulfills all the temporal constraints $c \in C$, and we call $[p_1(u)]_{j_1}..[p_m(u)]_{j_m}$ a

supported assignment. The STP predicate yields true iff at least one supported assignment is found. This completes our description of the STP predicate.

The *STP Algebra* in `SECONDO` defines other variants of the STP predicate (e.g., `stpatternextendstream`). This operator is a triple $\langle P, C, f \rangle$ where P, C are the same as before, and f is an additional condition on the time intervals of the supported assignments. One can express, for instance, that the `Down` predicate in this query must be fulfilled for at least 2 minutes. The `stpatternextendstream` is also a stream operator, not a predicate. It extends every input tuple with attributes containing the time intervals on which the pattern occurs. Since one trajectory might contain several matches of the pattern, the `stpatternextendstream` copies the tuple, and extends every copy with one match. The following example expresses the *stepwise descent* scenario:

```

1  ...stpatternextendstream[
2      Dive1: .SecondAltitudeDerivative < 0.0,
3      Lift: .SecondAltitudeDerivative >= 0.0,
4      Dive2: .SecondAltitudeDerivative < 0.0 ;
5      stconstraint("Dive1", "Lift", vec("aa.bb")),
6      stconstraint("Lift", "Dive2", vec("aa.bb"));
7      (end("Lift") - start("Lift")) > OneMinute ]
8  filter[isdefined(.Dive1) and
9      (AverageDiveAngle(.Alt atperiods .Lift) < 30.0)]...
```

In this scenario, the aircraft alternates between dive and cruise during its final approach. It is expressed as a sequence of increasing, decreasing, then again increasing rate of descent. Line 7 asserts that the `Lift` event stays more than a minute. Line 9 invokes the `SECONDO` function object `AverageDiveAngle` to assert that the aircraft is flying almost horizontally during the `Lift` event, having a slope of less than 30° with the horizontal. The two queries in this section finish in approximately 1 minute on the given data set with 17,851 trajectories (427,651 records). The `SECONDO` relation storing these flight trajectories occupies approximately 172 MB of disk-space on a Linux 32 bit machine.

12.6.2 Exploring Patterns by Integrating MOD with Visual Analytics

So far, we have shown that the STP predicates and its variants are very flexible and can be used to express arbitrarily complex patterns. In practice, tuning the parameters of these operators is tricky. The integration with visual analytics allows for fine tuning these parameters through user interaction. `SECONDO` and `V-Analytics` realize such an integration scheme. They are integrated so that it is possible to interchange query results in both directions. Typically the user starts by loading the whole data set in the databases of the two systems. The

exploration starts in V-Analytics by removing incomplete data and artifacts, and sending the identifiers of the candidate trajectories to `SECONDO`. In `SECONDO` the user issues an STP query, and moves the result back to V-Analytics for validation. The visualization in V-Analytics helps the human analyst in refining the query parameters. It can take as many cycles as needed between `SECONDO` and V-Analytics till the results are satisfactory.

The STP query can be written in `SECONDO` so that the result contains the time intervals in which the pattern occurred. These can be interpreted as movement events (m-events) in V-Analytics, so that the analysis procedures in the previous section are applicable. For example, one is able to explore the percentage of stepwise descents during one day, the percentage of missed approaches for each airport, the temporal distribution of missed approaches for a given airport, and so on.

12.7 Conclusions

In this chapter, we gave an overview of up-to-date research techniques to explore and analyze trajectories. We detailed our motivations, gave the process we used to build trajectory data set, and explained three trajectory exploration techniques (direct manipulation, m-event, and MOD queries).

First, we introduced FromDaDy, a multidimensional visualization tool making it possible to explore large sets of aircraft trajectories with direct manipulation techniques. It uses a minimalist interface: a desktop with a matrix of cells, and a dimension-to-visual variables connection tool. Its interactions are also minimalist: brushing, picking, and dropping. Nevertheless the combination of these interactions permits numerous functions: the creation and destruction of working views, the initiation and refinement of selections, the filtering of data sets, the application of Boolean operations. The cornerstone of FromDaDy is the trajectory spreading across views with a simple brush/pick/drop paradigm. With the incremental trajectory exploration and direct manipulation, the user can discover the worthwhile requests for data sets. In a sense, the user explores the data set, and at the same time, explores the request to perform.

Second, we detailed a generic procedure for analyzing mobility data that is oriented to a class of problems where relevant places need to be determined from the mobility data in order to study place-related patterns of events and movements. The procedure includes: (1) extraction of relevant events from trajectories by queries involving diverse instant, interval, and cumulative characteristics of the movement and relations between the moving objects and elements of the spatio-temporal context; (2) density-based clustering of the events by spatial positions, temporal positions, movement directions and, possibly, other attributes, which may be done in two stages for an effective removal of noise and getting clear clusters; (3) spatio-temporal aggregation of events and trajectories

by the extracted places; and (4) analysis of the aggregated data. Visual analytics and m-events provide a rich tool box of data transformations and visualizations which help a human analyst exploring the data.

Third, MOD queries deal efficiently with vary large data sets with theoretically no limitation, and are able to express complex queries (neighborhood, patterns, aggregations, etc.). Although direct manipulation is easy to use (users are accustomed to manipulating tangible objects), it does not support automatic exploration. Furthermore, direct manipulation techniques need to be interactive, which works against the data size. For instance, FromDaDy can display up to 10 million points with an acceptable frame rate. If more data need to be displayed or manipulated, new computation techniques need to be developed.

Since our visual analytics process uses m-events (geographic and temporal events), this tool is not suitable for complex computations such as pattern extraction. MOD can easily extract patterns, but the user needs to know in advance what he or she is looking for. MOD systems are not good for data exploration. As a future work, we plan to break the direct manipulation data set limitation with new interaction paradigms (more complex Boolean operations). We also plan to combine MOD, visual analytics, and direct manipulation to explore large data sets. Visualize a small sample, roughly figure out your query parameters, issue the query in MOD, validate the results by visual analytics, refine the MOD query, and so on and so forth.

12.8 Bibliographic Notes

For further reading, we recommend the book by [Card et al. \(1999\)](#), which details the information visualization research area. We also recommend the book by [Tufte \(1990\)](#), which contains many remarkable visualization instances. Two conference proceedings contain many examples of visualizations and interaction techniques. InfoVis: The IEEE Information Visualization Conference (IEEE Transactions on Visualization and Computer Graphics) contains novel research ideas and innovative applications in all areas of information visualization. Also, VAST, the IEEE Conference on Visual Analytics Science and Technology, is the first international conference dedicated to advances in visual analytics science and technology. The scope of the conference includes fundamental research contributions within visual analytics as well as applications of visual analytics, including applications in science, engineering, medicine, health, media, business, social interaction, and security and investigative analysis.

The spatio-temporal pattern predicate was first proposed in [Sakr and Güting \(2011\)](#). It is demonstrated in [Sakr et al. \(2011\)](#). We used this demonstration as the basis of Section 12.6.

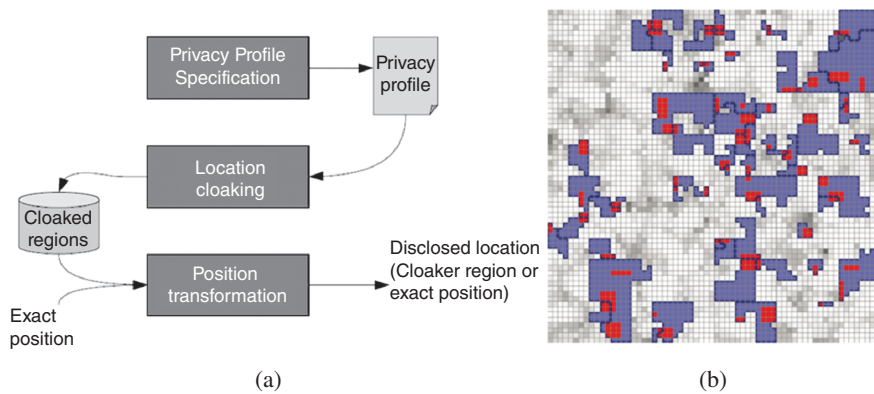


Plate 2.9 The Probe system: (a) The workflow. (b) Obfuscated map: the blue polygons represent cloaked regions, the red rectangles sensitive places, the gray background the distribution of population in space.

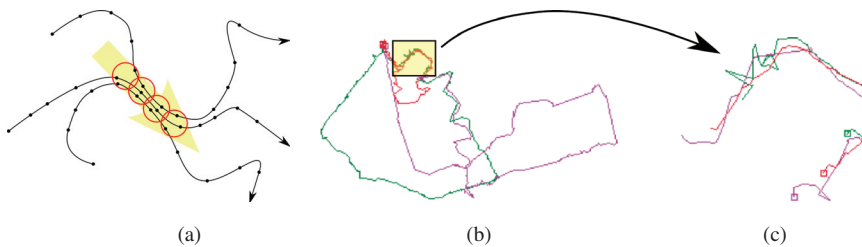


Plate 6.1 Visual representation of (a) a trajectory flock, (b) a sample result on a real data set with all trajectories involved, and (c) a zoom on the segments that form the flock.

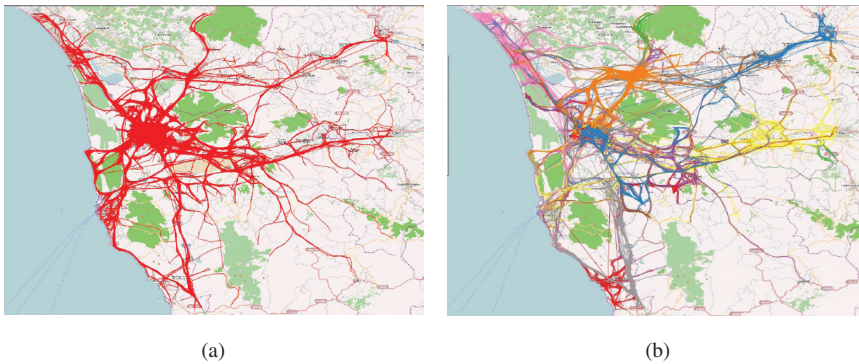
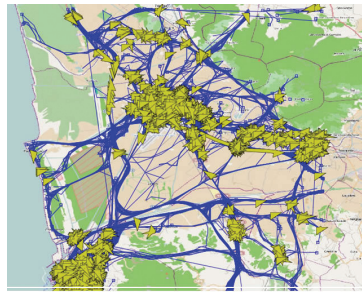
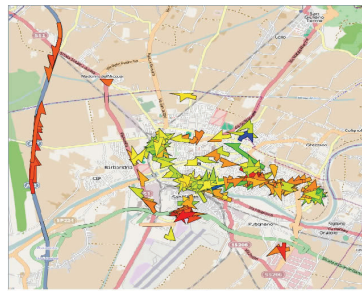


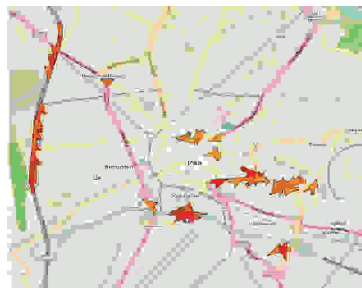
Plate 6.6 Sample trajectory clustering on a real data set of vehicles (GPS data collected by OctoTelematics S.p.A.), obtained using a density-based clustering schema and a spatial route distance function.



(a)

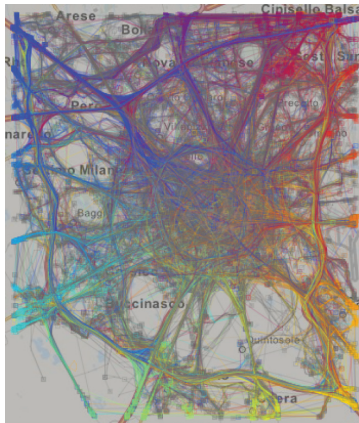


(b)

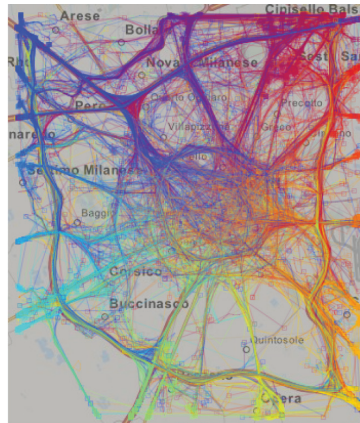


(c)

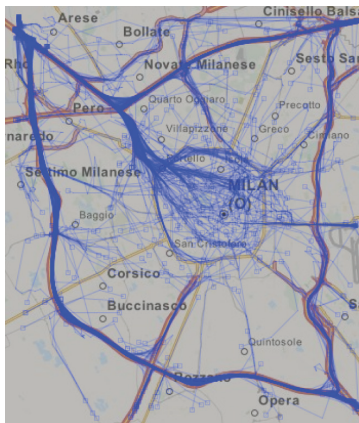
Plate 7.4 A graphical representation of the process of extracting traffic jams from the data. (a) Using the T-flock algorithm all the candidates are extracted. (b) The patterns are colored based on ratio between their speed and the free speed in the same area (Blue > 1 , Red < 1). (c) The patterns with a speed lower than $1/4$ of the free speed.



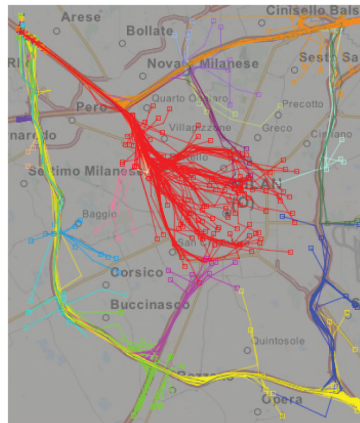
(a)



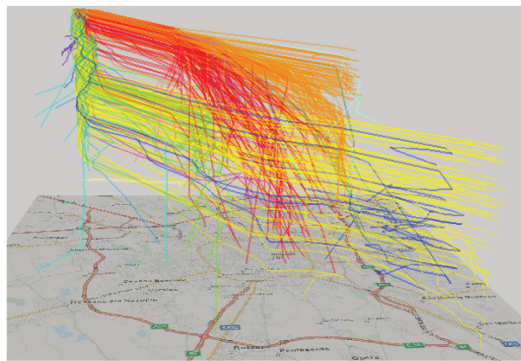
(b)



(c)

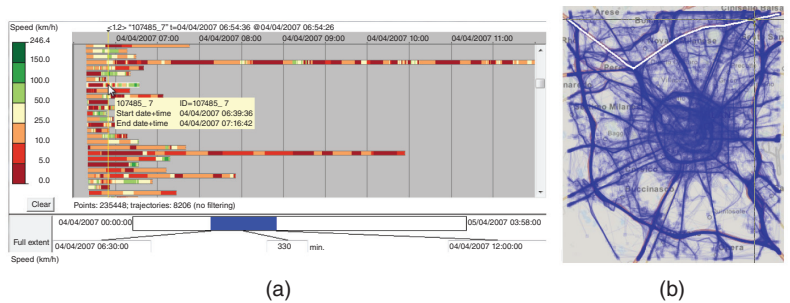


(d)



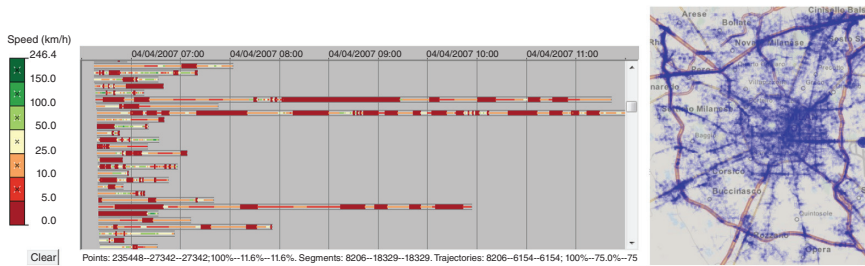
(e)

Plate 8.2 Interactive progressive clustering of trajectories. (a) The car trajectories have been clustered according to the destinations. (b) The noise is hidden. (c) One of the clusters is selected. (d) Clustering by route similarity has been applied to the selected cluster; the noise is hidden. (e) The clusters by route similarity are shown in an STC.



(a)

(b)



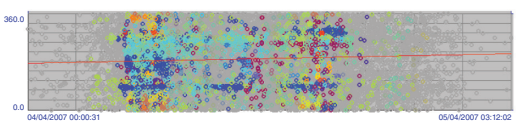
(c)

(d)



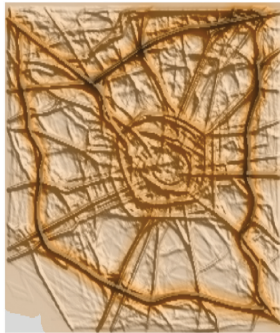
(e)

(f)

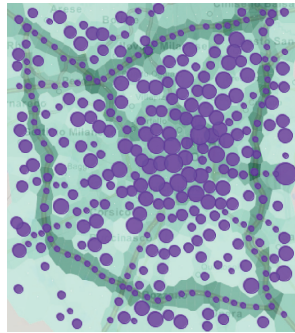


(g)

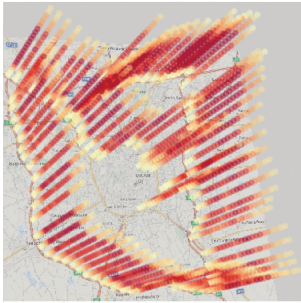
Plate 8.3 (a) A time bars display shows the speeds by color-coding. Mouse-pointing highlights the trajectory and marks the pointed position in a map (b). (c) Trajectory segments are filtered according to the speed values. (d) Only the segments satisfying the filter are visible on the map. (e) Low-speed events have been extracted from the trajectories according to the segment filter. (f) Density-based spatio-temporal clusters of the low speed events are shown in a space-time cube. (g) A scatterplot shows the times (horizontal dimension) and movement directions (vertical dimension) of the low-speed events.



(a)



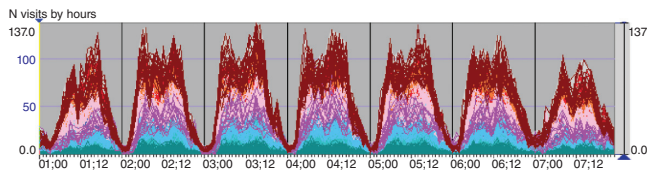
(b)



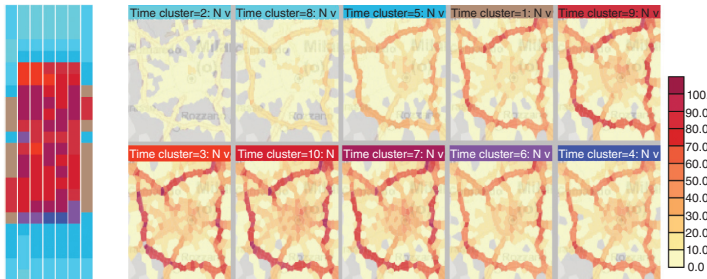
(c)



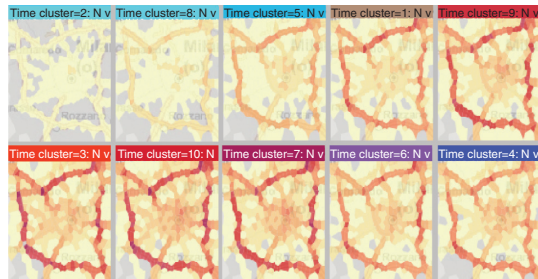
(d)



(e)



(f)



(g)

Plate 8.4 (a,b) Car tracks aggregated in a continuous density surface (a) and by discrete grid cells (b). (c) STC shows the variation of car presence over a day in the most visited cells. (d) The cells clustered by similarity of the presence time series shown on a time graph in (e). (f) Hourly time intervals clustered by similarity of the spatial distributions of car presence, which are summarized in (b).

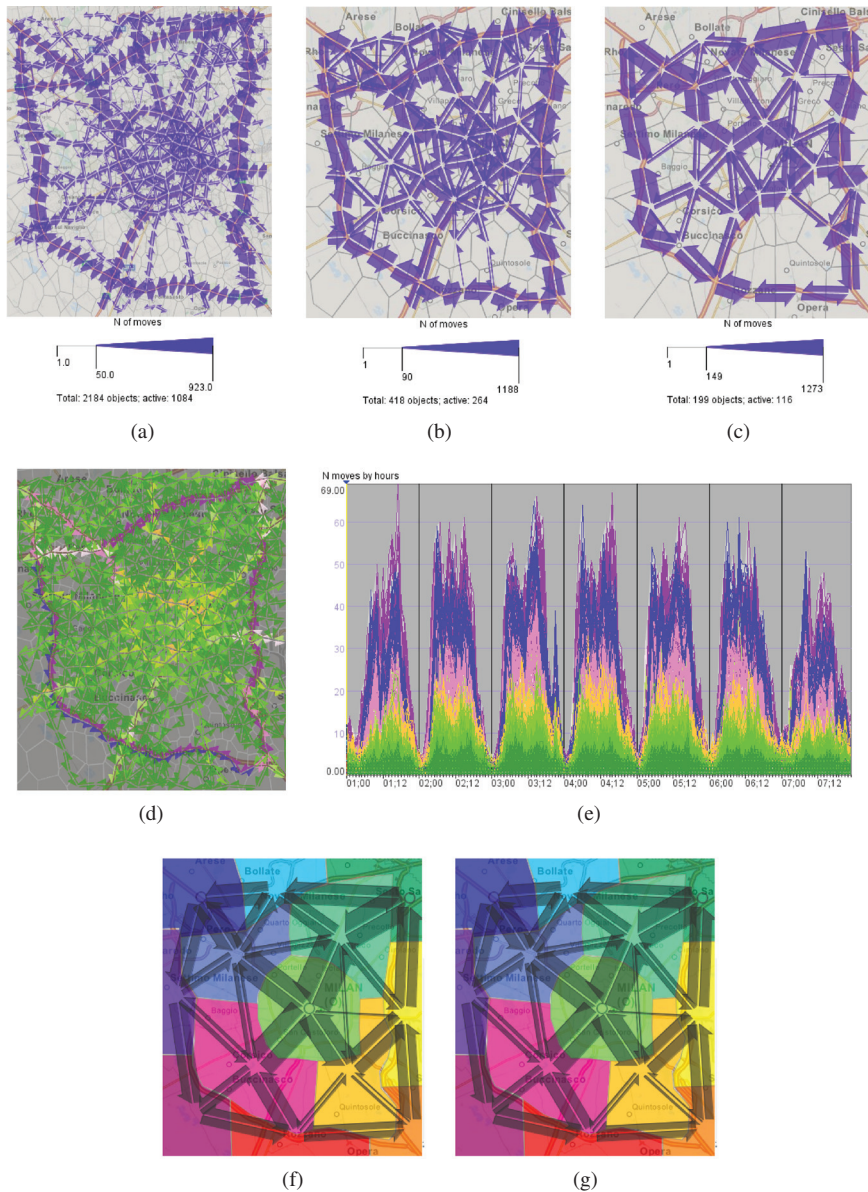
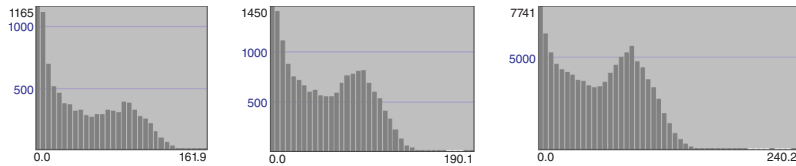
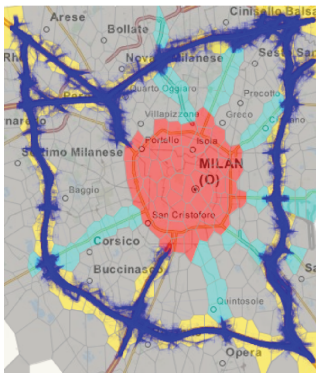


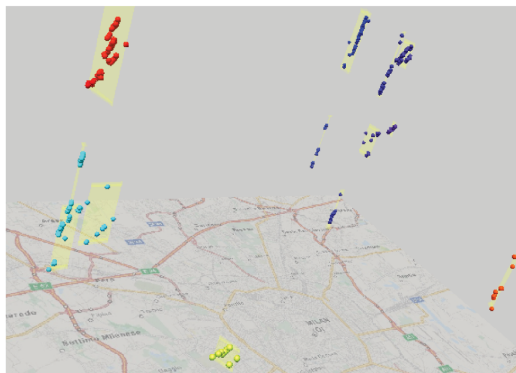
Plate 8.5 (a,b,c) Flow maps based on fine, medium, and coarse territory divisions obtained automatically. (d,e) Clustering of flows based on the time series of flow magnitudes. (f) Flows between predefined regions. (g) Investigation of movements between the regions over time adjusted to individual lifetimes of the trajectories.



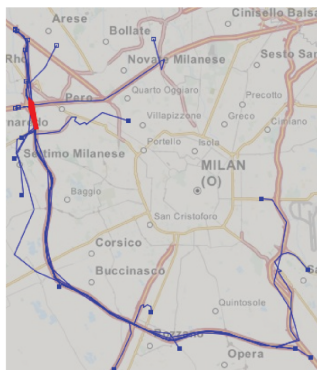
(a) (b) (c)



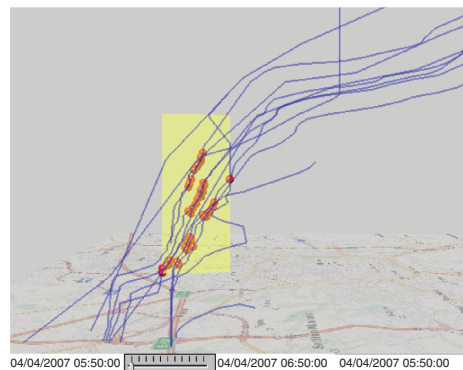
(d)



(e)



(f)

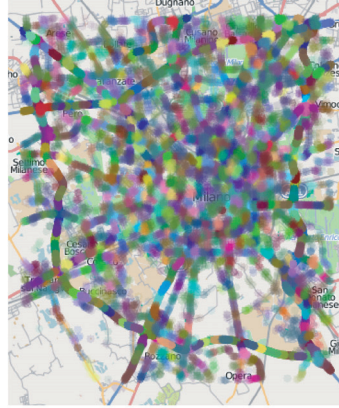


(g)

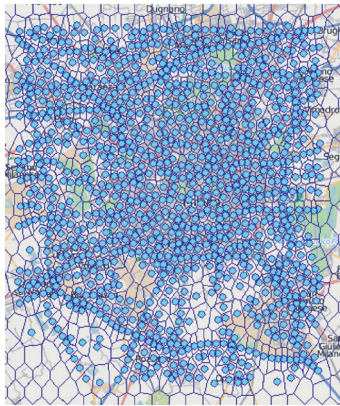
Plate 8.6 (a,b,c) Frequency distributions of car speeds on motorways in different ranges of distance to the nearest neighbor car: (a) below 20 m, (b) 20–50 m, (c) over 50 m. (d) Trajectory segments on or near motorways selected by means of segment filter. (e) Spatio-temporal clusters of low-speed events on motorways where the distance to the nearest neighbor is 10 m or less. Yellow shapes represent spatio-temporal convex hulls of the clusters. (f) Trajectories that passed through one of the convex hulls are selected by filtering. (g) The selected trajectories and respective low speed events in a STC.



(a)



(b)

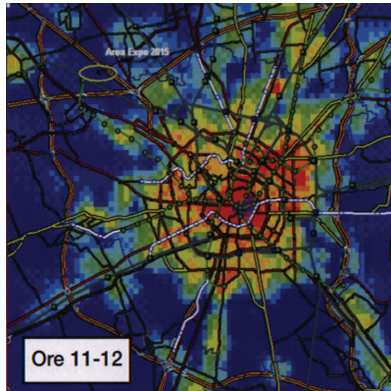


(c)

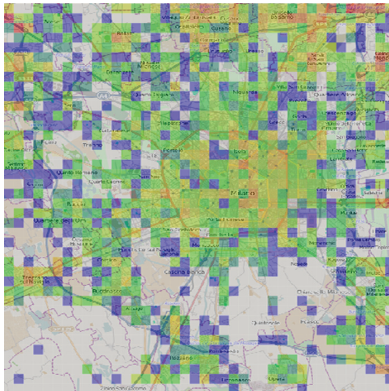


(d)

Plate 9.5 Anonymization steps. (a) Characteristic points. (b) Spatial clusters. (c) Territory tessellation. (d) Generalized trajectories.



(a)



(b)

Plate 10.2 Presence distribution between 11 A.M. and noon, (a) survey, (b) GPS data; frequent locations plotted with lighter shades.

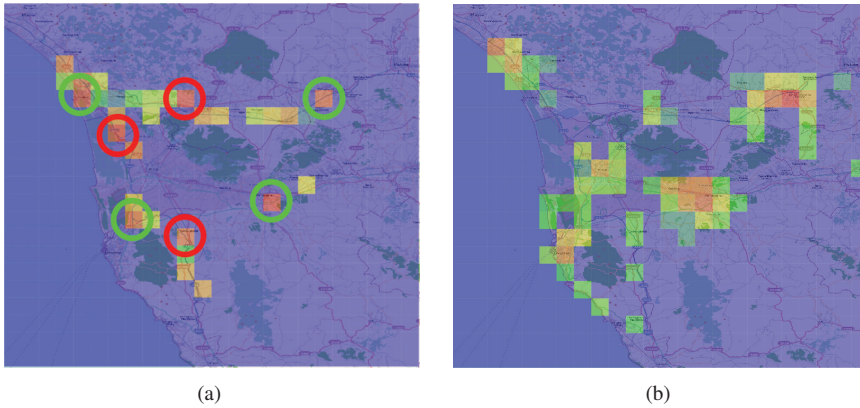


Plate 10.13 Distribution of presence: (a) with predicted trajectories, (b) with the real trajectories. As highlighted on (a), the predictor is able to correctly guess the most dense locations (green circles), though it introduces some false positives (red circles).

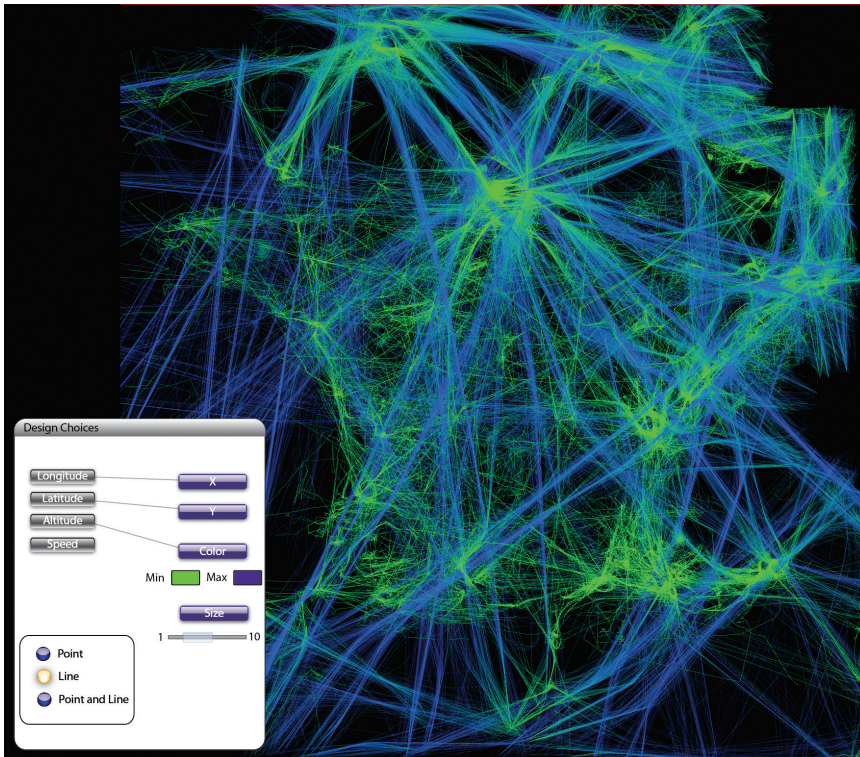
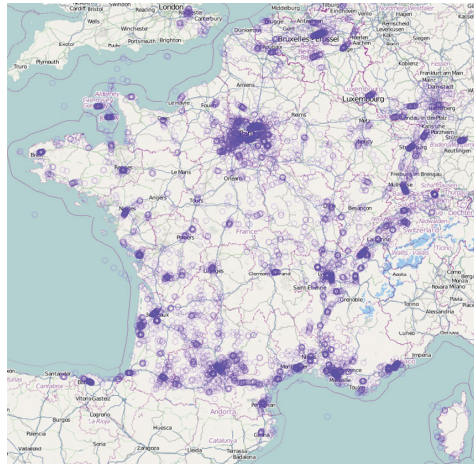
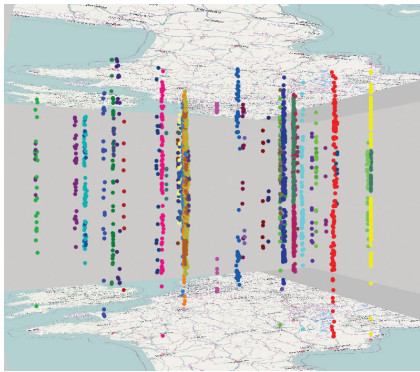


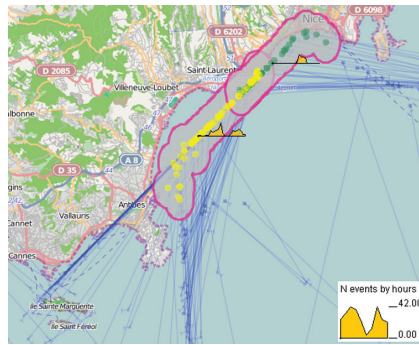
Plate 12.3 One day's record of traffic over France. The color gradient from green to blue represents the ascending altitude of aircraft (green being the lowest and blue the highest altitude). The French coastline is apparent here in terms of pleasure flights by light aircraft and the straight blue lines represent high altitude flight routes. A user interface shows the data set fields and the defined visual configuration.



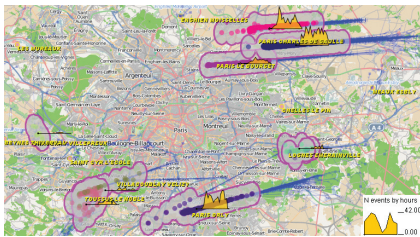
(a)



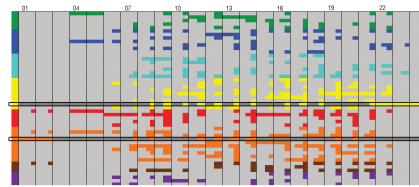
(b)



(c)



(d)



(e)

Plate 12.6 Event extraction results. (a) The positions of the landing events extracted from the flight data are drawn with 50% opacity. (b) The space-time cube shows the landing events clustered by spatial positions and directions. (c) The yellow and green dots represent two SD-clusters of landings in the airport of Nice. The time diagrams show the dynamics of the landings from two directions. (d) The time diagrams show the dynamics of landings in the airports of Paris. (e) The flight distribution between the airports by hourly intervals. Highlighted are rows for the connections Marseille–Paris (yellow) and Paris–Marseille (orange).

13

Animal Movement

Stefano Focardi and Francesca Cagnacci

13.1 Introduction

13.1.1 Historical Overview

The curiosity of humans about animal movements dates back to ancient times and probably to prehistory. As a matter of fact, Aristotle (in *The History of Animals*) described animal migrations. The capacity of animals to move with accuracy during long displacements was surprising and has been considered a mystery of nature till recent times. Much before the scientific foundation of diffusion due to the botanist Robert Brown in 1927, the roman poet Lucretius described in detail the motion of dust. For centuries, scholars hold Descartes' view that animals are thoughtless automata. Modern experimental research dates back to the end of nineteenth century, after the publication of *The Origin of Species* by Darwin in 1859. Researchers of that period adopted a subjective and anthropomorphic view of animal behavior and movements. Later, scholars started to interpret animal movement in a more objective, scientifically sound way, by investigating animal reactions to stimuli present in their environment, such as the gravitational field, the presence of light, gradient of humidity, and so forth.

The concept that individual animals restrict their movements to finite areas known as *home ranges* is perhaps as old as ecology itself. Seton in 1909 observed that "No wild animal roams at random over the country; each has a home-region, even if it has not an actual home." The definition of home range from Burt, dating back to 1943, is probably one of the most long lasting and widely used in ecology: "that area traversed by the individual in its normal activities of food gathering, mating and caring for young. Occasional sallies outside the area, perhaps exploratory in nature, should not be considered as in part of the home range." This definition does not contain a quantitative definition of home-range boundaries, but it implies that a home range is a well-identifiable area; one consequence

is that movement of animals is constrained to boundaries. Another implication of Burt's definition is that space use can arise from different behavioral activities such as finding food, shelter, partners, and where they survive, reproduce, and maximize their fitness, that is, the use of space is tightly connected to selective pressures. Indeed, those are the same forces causative of much more impressive movement bursts, that is, migrations. Forms of movement behavior intermediate between migration and residence have been described; for example, nomadism or commuting behavior. A reductionist approach was used by observing the behavior of organisms (usually invertebrates) in a simple sensorial environment in controlled experiments. This research showed how simple behavioral mechanisms were adaptive for the animals, which were thus able to avoid stress factors and exploit windows of opportunities to get significant resources. In other words, these studies introduced the idea that appropriate responses of organisms to cues present in the environment allowed them to attain simple forms of habitat selection, improving their fitness. The analysis of movement becomes hence fully embedded in the evolutionary theory.

Past studies have led to important definitions still used in animal movement studies:

- A *stimulus* represents a cue in the external environment that produces predictable physiological modifications. Stimuli can be scalar if they do not carry directional information (e.g., temperature, chemical concentration) or vectorial if they carry directional information (electromagnetic field, light beam).
- When orientation occurs on the basis of a scalar stimulus the orientation mechanism is called *kinesis*. The signal can induce a variation in the speed of movement (ortho-kinesis) or in the turning frequency or turning angles (klino-kinesis).
- When the stimulus is vectorial, the orientation mechanism is called *taxis*. According to the direction of the movement with respect to the direction of the stimulus we speak of positive or negative taxis. According to the nature of the stimulus we have photo-taxis, geo-taxis, chemio-taxis, and so on.

13.1.2 State of the Art

Animal movements can be categorized in different broad categories along a continuum of sedentarism–nomadism.

- *Home range*: Sedentary animals use a stable range. The definition excludes occasional sallies or exploratory movements outside the home range. Usually only the 95% of inner spatio-temporal positions are considered part of the home range.

- *Commuting*: Recently, an intermediate use of space between residence and migration has been described as “commuting behavior,” that is, displacement of individual animals between resources that are set apart in space, but not in time.
- *Migration*: Migratory movement defines the shift of an organism between two nonoverlapping home ranges. Typically, migration is a seasonal process, but it can span also an individual life cycle, or even several generations. Vertical migrations represent the special case where organisms shift up and down a fluid column. Altitudinal migration indicates a shift between lowland and elevations. It must be noted that migration may refer to the population or to the individuals. Partial migration indicates that only a segment of the population migrates, while facultative migration indicates that an individual may, or may not, migrate. Differential migration means that two segments (typically males and females) of a population have different migratory schedules. Proterandric (proteroginic) migration indicates that males (females) migrate before the other sex.
- *Dispersal*: At the individual level, dispersal indicates spreading with respect to a reference point or area. The dispersal from the origin of the movement is given by its mean squared displacement, $MSD = (\mathbf{x}_t - \mathbf{x}_0)^2$, where \mathbf{x}_t denotes the coordinates at time t and \mathbf{x}_0 at time 0, respectively. The more common types of dispersal are natal dispersal, when an organism leaves forever the range where it was born, and mating dispersal, when the home range is left only for breeding purposes. The adaptive consequence of this behavior is to reduce inbreeding.
- *Nomadism*: Nomadic behavior refers to an opportunistic use of space, which is continuously searched for resources from one spot to the following.

Migrations represent one of the most surprising patterns observable in nature; animals can move for thousands of kilometers and finally recover their wintering, or breeding, grounds. Indeed much effort has been dedicated to the study of long-range migrations, which are quite impressive examples of animal movement. The Artic stern, *Sterna paradisea*, for instance, migrates from the North to the South Pole, flying about 80,000 km per year. Both marine and terrestrial mammals perform long range migrations. Grey whales (*Eschrichtius spp.*) in the Pacific Ocean move from the Baja California, where they reproduce, to the Arctic Ocean to forage; wildebeests (*Connochaetes taurinus*) and other ungulates move hundreds of kilometers to attain favorable foraging habitats. Another very impressive example is the migration of the European eel (*Anguilla anguilla*) from Europe to the Sargassum sea, 5,000 km.

Migrations are outstanding movements that allow animals to exploit resources (food, breeding territories, or refuges) that are separated in space and time. By migrating, animals reach the most suitable conditions to their survival, and

reproduction, at a certain time of their seasonal activity (e.g., bird migrations), life cycle (i.e., eel, or atlantic salmon, *Salmo salar* migrations) or life history across generations (i.e., monarch butterfly, *Danaus plexippus*). When individuals have reached the final destination of their long displacements, they range over shorter distances to select and use local resources.

At the other end of the movement behavior continuum, and very commonly, animal species show a *sedentary* behavior, that is, they occupy a *home range*. Use of space, and therefore movement, is tightly linked to the use of resources: the former ecological concept cannot be understood without taking into account the latter. Many animals use stable refuge for egg laying, rearing the juveniles, accumulating reserves, overriding unfavorable climatic conditions, and so forth. These organisms alternate the use of the refuge with excursions in the external environment where they search for resources (e.g., food and mates). Clearly for these organisms, it is vital to recover quickly and safely their refuge. The homing pigeon (*Columba livia*) represents the paradigmatic example of this behavior. For homing successfully an animal needs two tools: a map to know its own position (with respect to home) and a mechanism (usually a taxis) to move in the right direction. Many orienting mechanisms based on different cues (sun, moon, stars, magnetic field, light polarization, and so forth) have been demonstrated, although the maps used in animal navigation are more elusive. The earth's magnetic field can be used as a global map. At shorter distances it is indeed possible to use olfactory maps, and in the area usually explored by animals, a memory-based landmark map can be effective. Landmarks can be of different kinds, and usually these are naturally present in the environment, but sometimes these are pheromones purposely laid by the animals themselves, such as in trail-following of ants, snails and butterflies. *Navigation*, the ability to use "compasses and maps," has been demonstrated for several species of animals in different taxonomic groups.

Another broad line of research on animal movement is represented by the use of space outside the refuge, if any, by the animal. When an organism faces contradictory requirements while exploiting an environment, a trade-off between the needs of minimizing risks and maximizing resource acquisition exist. Thus, the available space is not used at random; some areas are preferred and others avoided. The ranging movements of the animal are therefore led by the optimal use of available space and resources and are constrained by both physical (e.g., presence of natural obstacles) and biological (presence of competitors and predators) factors. The resulting area is defined as home range, or as territory, depending on whether it is defended or not against intruders.

Many methods for computing home range size have been proposed. Now there is a general agreement that kernel density distribution methods represent an appropriate approach for describing the structure of home ranges. Quite recently, an innovative approach has been proposed based on the formulation

of mechanistic models of home range. A mechanistic approach implies that the researcher is able to formulate competing models based on hypotheses about the action of causal factors on animal movement.

The tactic used by the animal to invest time in different parts of the home range is called habitat selection. Habitat selection represents a differential use of a resource with respect to its availability. It is usually evaluated as a hierarchical process at different levels. The first order selection or level is the selection of a geographical range by a species, the second level is the selection of the home range with respect to the range typical of the species, the third level is the selection of different habitats within the home range, and the fourth level is the selection of a resource item (typically food) within the habitat. More specifically, it refers to a hierarchical process of behavioral responses that may result in the disproportionate use of habitats with respect to their availability. Finally, habitat selection studies have taken advantage of the development of generalized mixed model platforms, which allow researchers obtain realistic and assumption-free models for habitat selection.

Optimization of resource acquisition is analyzed by optimal foraging theory. In the classical approach, OFT describes the optimal use of resources after they have been found by the animal (post-encounter processes). A new and interesting field of research deals with the problem of optimal search (pre-encounter processes).

The analysis of the different kinds of animal movements has received a formidable boost by technological development. Recording animal movement under natural conditions (known as *animal tracking*) is fundamental to understanding why and how animals move. Even today this task is not trivial. The first important breakthrough was represented by the development of very high frequency (VHF) telemetry. Animals are fitted with transmitting devices, the signal is recorded by a receiver, and the spatio-temporal position (which in ecological literature is referred to as “a fix”) of the animal is obtained using different methods, mainly based on triangulation. This approach, albeit valuable, present several shortcomings. In the earliest times, VHF telemetry was more amenable for terrestrial than flying animals due to the weight of transmitters, but now VHF transmitters have been fitted even on insects (e.g., large grasshoppers). The major limitation is represented by the need of operators to retrieve the signal, who may in turn “lose” animals when the animals are moving quickly. Therefore, VHF telemetry was especially useful for animals residing in a known area, while the collection of long range displacements was quite difficult, missing most of the migratory or dispersing movements. In the 1980s the development of Platform Terminal Transmitters (PPTs) to uplink data to Argos satellites (using Doppler-based positioning to compute animal locations) produced the first records of wide-range movements of marine mammals and birds (note that to contact the satellite the device must to be outside water), but the true

revolution that has enormously spurred the study of animal movement was the advent of GPS-based devices to track animals (Cagnacci et al., 2010).

In parallel with technological and experimental developments there was an improvement of the statistical methods necessary to study spatial and temporal processes. In Chapter 1, a *raw trajectory* is described by a list of tuples containing mainly the instant and point of the moving object. In this chapter we use another representation, a list of vectors, each one (in a 2D space plus time) characterized by angle and distance. However, the statistical analysis of angles is challenging because it requires a specific approach, because angles are defined in the interval $-\pi$ and $+\pi$ and appropriate distributions are obtained by wrapping conventional linear distributions (i.e. $-\pi = +\pi$, or $2\pi = 0$). A relevant improvement was represented by the use of the use of *circular statistics* in the study of animal orientation. The first compendium on the modern analysis of biological diffusion was due to Okubo, although Turchin provided a comprehensive theoretical summary. The discipline studying animal paths is referred to as *trajectometry*.

The literature about animal movement is double-faced: the newcomer to this field has to be aware that two different approaches are used. Many scholars investigate the *proximate* causes of movement, for instance, which orienting cues an animal uses to move from point A to B. On the other hand, researchers are interested in the *ultimate* causes of movement, for instance, which are the factors causing the size of one animal's home range. Indeed, there is not a clear separation between the two approaches and today the use of complex statistical modeling makes it possible to investigate both levels of causation within the same framework. Studies on animal movement have to be directly linked to evolutionary theory, species life history, and the ecological modulation of behavior.

The aim of this chapter is to give a presentation of the state of the art in the study of animal movement, which could help the student or the beginner to orient him or herself in this rather cumbersome field of research. In this chapter we try to avoid as much as possible mathematical formulations and we will use verbal models and simulations to illustrate the main concepts. Thus the reader can (1) use this chapter as an introduction to more complex and mathematically demanding papers, or (2) grasp the main concepts in order to better plan data collection and experiments and to acquire concepts and terminology useful to foster cooperation with statistical and mathematical experts.

13.2 The Study of Animal Movement

13.2.1 A Revolution: *Biologging Technology*

The simplest method to study animal behavior is to use individually recognizable tags, such as rings, collars, and ear tags. The results obtained by tags are prone to bias, due, for instance, to differences in recovery rates (in times and/or space) or

to the fact that only animals surviving the movement can usually be recovered. Despite such limitations, simple methods have allowed researchers to learn important information about the life history traits of many species.¹

In general, the use of telemetry has improved the sampling design, reduced bias, and improved reliability. In this paper, we denote by *biologger* any animal-borne device able to record position and/or environmental/physiological data. Miniaturization of GPS devices has allowed development of small and light devices that can be fitted to a large number of animal species. It was possible to shift by a collection of spatio-temporal positions (more or less statistically independent) as done using VHF telemetry to a very dense (and highly correlated) sampling of locations, which may represent an approximation of the actual path followed by the animal. GPS devices use different technologies to transmit position data. The GPS store-on-board (SOB) devices are recovered after use and data are downloaded. SOBs are usually cheap but require that one is able to recapture the animal or recover the device; SOBs can be used for nesting birds or other animals likely to be easily recaptured or harvested. Drop-off mechanisms that should cause the detachment of the SOB from the animal do not always perform well. GPS-GSM use a GSM (Global Service for Mobile Communication) public network to exchange data between the biologger and the user, often using small message services (SMS). Clearly this method is useful only in those countries where there are dense GSM networks. In other situations there are several systems to remotely download GPS collars and retrieve the data. Probably the cheapest method is to use a VHF beacon. The receiver can approach the tagged animal on the terrain or using an airplane. This system is useful in wild areas where the amount of data to recover is limited. The alternative is transmitting data to a satellite constellation. There are several possibilities: to exploit the Argos DCLS transmission, which allows only one-way transmission from the animal to the user, or using satellite mobile phone systems (namely Iridium and Globalstar services).

13.2.2 Interpretation of Animal Movement

As happens in human mobility (see Chapter 1), wildlife telemetry has quickly changed in the last years; “a brave new world” (Tomkiewicz et al., 2010) arose and now researchers have an array of technologies able to record the trajectories of many species of wild animals with high accuracy and in many different ecological conditions, from the desert to the deep ocean, worldwide. It is fundamental to be able to exploit efficiently the information contained (some might say “hidden”) in movement data for a large array of scientific and management purposes. An important consequence of the use of biologging in ecology is the

¹ See, e.g., <http://www.phidot.org/software/mark/docs/book/>

availability of large data sets on animal behavior. Spatial databases represent a new challenge and opportunity for scientists. On one hand, suitable analytical methods are needed, on the other hand this requires the application of appropriate data management tools. The amount of movement data recorded on hundreds of species and many thousands of individuals has accumulated over the last 20 years. This represents a “treasure” and allows researchers to study species at their distribution ranges, and to address “general questions.” Examples are MoveBank,² TOPP,³ and EURODEER.⁴

13.2.3 Emerging Theories

New paradigms and technical challenges have recently pervaded the analysis of animal movement. On the empirical side, the extensive use of biologgers grants for the first time systematic access to animal-borne information, such as positional, behavioral, physiological, and environmental parameters. At the theoretical level, the large amount of quantitative data thus obtained naturally encourages the development of new analytical concepts and computational tools to analyze and understand movement and its associated behaviors/parameters. Large amounts of data can be first screened with *data mining* automatic procedures, which can search for inner consistent structure of data. These procedures can prove very useful, but they have to be supervised by sound ecological interpretations. On a theoretical level, a unifying framework and an integrative paradigm for animal movement has been recently proposed, referred to as *movement ecology*:

the proposed framework integrates eclectic research on movement into a structured paradigm and aims at providing a basis for hypothesis generation and a vehicle facilitating the understanding of the causes, mechanisms, and spatio-temporal patterns of movement and their role in various ecological and evolutionary processes. (Nathan et al., 2008: 19052)

Movement ecology aims to become an hypothesis-based (sensu Karl Popper) discipline where theory dictates experiments and observations. Ecological studies (1) have a strong interest for the orienting mechanisms underneath animal movement (e.g., kinesis, navigation), (2) stress the importance of adaptive value of these movements (e.g., risk avoidance, resource gathering), and (3) model the consequences of individual movements at the level of social group and population (e.g., flocking behavior, diffusion). The ecologist is not especially interested in knowing what a given whale or deer is doing, but is interested in deducing general features (the tactics) of one species movement from a sample of individual

² <http://www.movebank.org>

³ <http://www.topp.org>

⁴ <http://www.eurodeer.org>

Table 13.1 *Types of Tags, Sampling Schedules Often Used, and Type of Analysis Performed. Geolocators Use Light Pattern to Compute Length of the Day (from which One Derives Latitude) and the Time of Solar Noon (Used to Compute Longitude).*

Type of Mark	Sampling Density	Sampling Schedule	Type of Movement Analyzed
Tag	Low (usually 2 are available)	On occasion	Migration Dispersal
VHF	Intermediate (e.g., 1 spatio-temporal position per some days)	Systematic (only daytime with airplanes)	Home range Habitat selection
Geocator	1 per day	Systematic	Migration
GPS	1 each few hours	Systematic	Home range Habitat selection Migration Dispersal
GPS	1 each few minutes	Systematic	Search behavior

trajectories. Movement tactics typically vary as a function of environmental conditions. In practice, good precision in reconstructing individual animal trajectories is necessary to increase the power of statistical tests. Ecologists are interested in describing patterns, but especially in understanding the processes below spatio-temporal patterns, so that statistical inference is the fundamental tool to be used. Models allow us to deal with ecological complexity at different scales. Two axes are relevant: the temporal scale of the explanation (proximate versus ultimate) and the sampling unit that goes from the individual to the population through kin and social groups. Orientation mechanisms, dispersal, and foraging are typically individual-specific but causes operate at different temporal scales (say, minutes, years, and generations). Population diffusion, spatial distribution, and interactions determine ecosystem complexity and biodiversity, that is, their long term properties. There are also intermediate processes: a typical home range lasts for the life span of each individual and is determined by its energetic needs but also by local interactions with neighbors. Movement analysis has the potential to unify these different aspects.

13.2.4 Data Sampling

In ecology an appropriate *sampling design* is the foundation of good science but its importance is often overlooked. Table 13.1 summarizes the different options, because technology, sampling design, and aims of the research are interlaced. There is a trade-off between the cost of the tracking device and sample size. For instance, tags are very cheap but to obtain reliable results one needs many hundreds or thousands of marked animals. The use of a systematic sampling schedule

allows one to reduce the sample size, which, however, should be representative of the variability within the population of interest. In case of GPS devices, there exists a trade-off (given a certain load allowed by organism's size) between the number of spatio-temporal positions and the duration of the survey. A compromise is often used, adopting a standard low frequency of spatio-temporal positions (typically one per 4 hours) but programming a denser collection of spatio-temporal positions during a specific period of interest (breeding time, natal dispersal, etc.).

There is a basic difference in sampling with VHF and GPS. In the former case one tends to use independent spatio-temporal positions, while in the second case there is a specific aim to exploit the autocorrelation among positions to deduce the movement tactics used by the organism. In the case of VHF telemetry the interval among spatio-temporal positions is large (e.g., at least 24 hours) to guarantee a certain degree of statistical independence, and the time of location shifts from an occasion to the next one (usually one or two hours) to cover the whole 24 hours (often sampling is stratified by month or season). However, this sampling approach makes difficult the analysis of trajectories. On the contrary, the usual sampling with GPS is to compute a location at fixed times of the day (e.g., midnight, 2, 4, . . . , 22) with an interval between spatio-temporal positions usually ranging between 2 and 6 hours. In general it is preferable to sample at fixed time so as to yield a good estimate of animal's speed. Modern collars, which are going to be on sale soon, have internal capabilities for analysis of geographical data. This would allow for adaptive sampling schedules, the merits and potentialities of which are not yet well understood.

Many methods are available for uncertainty reduction and to get a precise *trajectory reconstruction*. The causes of imprecision are reviewed in Chapter 5. In animal movement studies, fuzzy sets are not used for correcting spatial uncertainty, and state-space Bayesian models are becoming popular in the ecological literature. Independently of the method used, a filtering of the trajectory is usually necessary. It is useful to think to a hierarchical analysis: first-order analyses are intraindividual and so include spatio-temporal position correction, path interpolation, computation of angular resultants, mean speed of displacement, and so forth. The second-order analysis refers to the sample and in this case statistics are relative to the population on which to make inference, whether individuals are sampled independently.

13.2.5 Analysis of Animal Trajectories

Turchin (1998) represents the reference text in this field. Let us suppose we are recording the true trajectory or path of an animal. We may represent the path as a set of vectors connecting spatio-temporal positions. In the first case (Figure 13.1a) our sample overlaps the walked path at a large extent, while in the second case (Figure 13.1b) our representation is much coarser and important

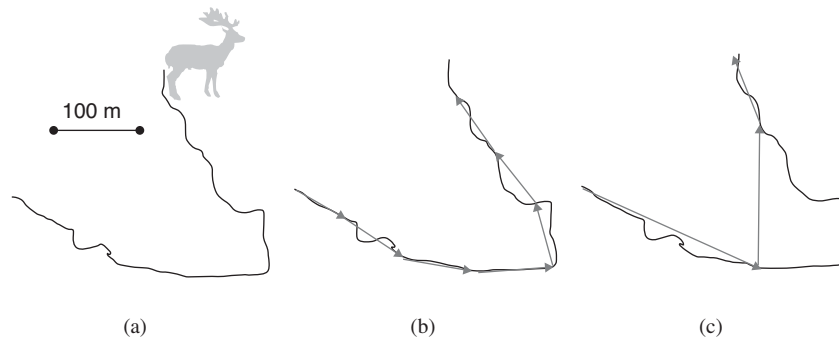


Figure 13.1 Discretization of animal trajectories. (a) The path of an animal (in the example a foraging deer) is perfectly recorded. (b) Path sampling at high resolution, and (c) sampling at low resolution. Sampling is done at fixed time intervals.

biological details might be lost. Clearly the appropriate sampling depends on the process of interest and there are not general guidelines. Thus, in order to plan the sampling design, researchers need some preliminary understanding of the dynamics of the process of interest. In the example of Figure 13.1, sampling (b) cannot be sufficient to investigating the selection of food items, while sampling (c) can be appropriate to investigate the annual home range.

The parameterization of a discretized path is represented in Figure 13.2a. Till now we have considered a fixed time sampling. The distance among spatio-temporal positions is referred as a “step.” The step itself is arbitrary and does not represent a significant behavioral feature of the studied organism. In the example path, during the first four steps the organism may have maintained the same motivation and behavioral tactics so that the small observed differences in speed and direction of steps may represent environmental or sampling nuisances (such as precision of a GPS collar, irregularities of the terrain or presence of obstacles for terrestrial organisms, or drifts due to wind or streams in flying or swimming animals). On the contrary, the spatio-temporal position at $t + 1$ – where a sharp directional shift occurs – may represent a changing of motivation. A move represents the distance between biologically meaningful variations in the path. It is thus appropriate to use a different discretization of the trajectory with variable temporal intervals (Figure 13.2b). This approach is more natural than the use of steps but it is difficult to identify turning points where meaningful behavioral changes occur. The basic idea to identify the moves is that within each move the α_i are strongly correlated (i.e., are very close to 0) while the β_i are uncorrelated.

13.2.6 An Example: Foraging and Social Behavior of the Fallow Deer

To exemplify methods used to investigate animal movement we review some studies about foraging and social behavior of fallow deer (*Dama dama*). In

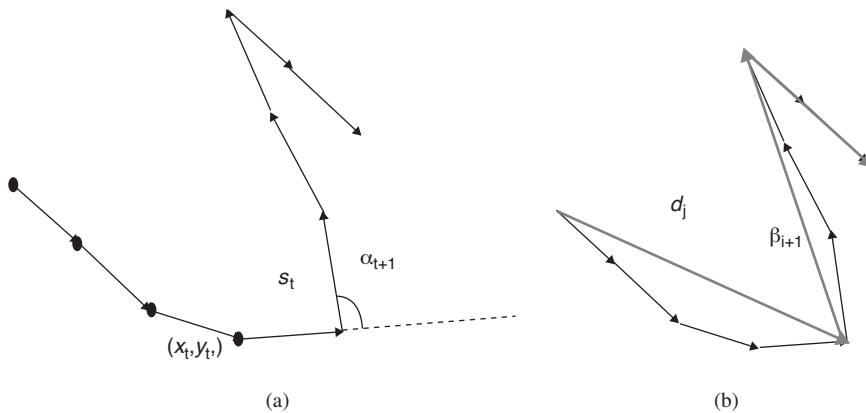


Figure 13.2 Computation of turning angles and move lengths in a trajectory. (a) A path sampled at fixed time intervals denoted by index t . Black dots represent spatio-temporal positions ($t = 1, 2, \dots, T$). Red arrows joining consecutive spatio-temporal positions are the steps of discretization of module s_t and angle α_t , (s_t is defined in $[1, T - 1]$ and α in $[2, T]$). (b) Sampling at unequal time intervals denoted by index i . Each grey arrow represents a move of length d_i . Note that α indicates turning angles between steps, while β represents turning angles between moves.

Figure 13.3a one can observe that fallow deer do not use pastures at random. Some locations are much more exploited than others. Even within the same habitat, animal movements appear quite variable. In Figure 13.3b we present some examples of paths of fallow deer recorded at twilight in an large open habitat surrounded by a dense forest where deer remain during the daytime in order to minimize disturbance and risk for neonates; at night they move to open fields to forage on good-quality food available in meadows, especially during spring. Spatio-temporal positions were collected by observers hidden in high seats scattered in the study areas to understand short-scale habitat selection and dynamics of social organization.

The aim of the study was to identify general mechanisms able to reproduce animal movements. The basic model proposed by Okubo (1980) is the Brownian diffusion. In this specific model, the distribution of turning angles, α_t , is uniform, that is, there is not directional persistence, while the distribution of distances d_t can assume different forms provided the originating distribution is characterized by a finite variance. Let us suppose we are studying the movement of a “jumping frog” in a one-dimensional universe (Figure 13.4). The rules of movement for a solitary frog are:

1. The frog jumps from the starting coordinate $d_0 = 0$ (at time $t = 0$). It may move leftward or rightward with probability 0.5.
2. Each jump, or move, is of constant length δ for a constant time duration τ .
3. Each move is independent from any previous move.

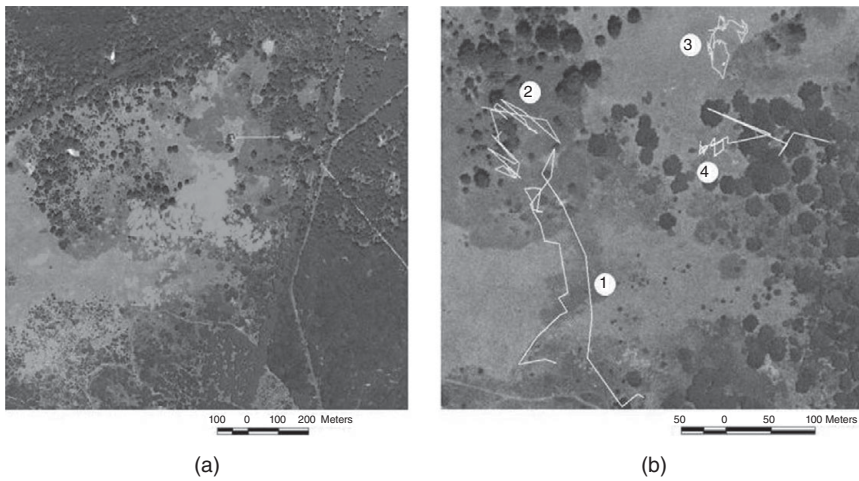


Figure 13.3 (a) Distribution of foraging station in the study area of Castelporziano, Rome, Italy. (b) Examples of foraging paths followed by individual fallow deer. Paths 2 and 3 are characterized by constant sinuosity, while Paths 1 and 4 are characterized by area-restricted search.

The question is to compute the probability, $p(d, t)$, that the individual is at a distance d from the starting point at a given time t . Clearly the displacement on the n th move is $d = \sum_1^n \delta_i$, where δ_i may be positive or negative according to Rule 1. For instance, with $n = 5$ one may have, among the others, the following series: $w = \{-\delta, +\delta, -\delta, -\delta, +\delta\}$, or $w = \{-\delta, -\delta, -\delta, -\delta, -\delta\}$. Each realization, w_n , of this stochastic process is called a *random walk*.

From a behavioral point of view the probability to arrive at coordinate three (on five moves) is to go rightward three times and leftward twice. In other words, it is the probability to have three “successes” (and hence two failures) on five “trials,” probability which is given by the binomial distribution. Of course, the probability of having five moves leftward (or rightward) is much smaller than having, for instance, three moves rightward and two moves leftward; this is because one has many more realizations in this case, as the order of moves is

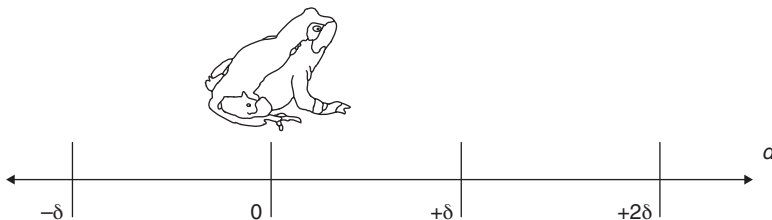


Figure 13.4 A jumping frog in the one-dimensional universe.

not relevant to compute the displacement: $\{+\delta, +\delta, -\delta, -\delta, -\delta\}$ is equivalent to $\{-\delta, -\delta, -\delta, +\delta, +\delta\}$ or $\{-\delta, +\delta, -\delta, +\delta, -\delta\}$, etc.

To be more realistic, let us consider a bidimensional space. In many cases it appears that animal movements are characterized by a directional persistence, in the sense that animals tend to persist along their previous direction. This is evident in Path 1 (Figure 13.3a) displaying the observed movement of a female fallow deer. The *correlated random walk* (CRW) is useful to represent directional persistence. CRW is similar to URW except that directions are correlated. This means that next steps are more or less oriented toward the same direction (e.g., to the north) so that turning angles are close to zero. The CRW represents a standard model to describe animals' movement (Turchin 1998). For instance, Paths 2 and 3 in Figure 13.3 are more sinuous than Paths 1 and 4.

In some respects CRW appears a more realistic model for the movement of actual organisms than URW, however, there are some shortcomings in this approach. First, movement appears more or less sinuous but the amount of turning is similar (apart stochastic fluctuations) along the path. In the simple case where the step length d is constant, the sinuosity $S = \frac{\sigma}{\sqrt{d}}$, where σ is the standard deviation of the angular distribution. However, many animals exhibit areas where sinuosity is high intermingled with areas where the path is straighter. This behavior is called *area-restricted search* (ARS).

However, animal tactics can be more complex to increase search efficiency, that is, the amount of resources encountered per unit time. The walker intensifies its search (increases path sinuosity) in areas where the density of targets is likely to be higher than on average (e.g., in a food patch) and perform more linear paths while moving among patches. This is represented by Path 4 (Figure 13.3). In fallow deer it is possible to show the presence of area-restricted search by computing the autocorrelation function of move length or the cross-correlation between angles and distance. Fallow deer present a positive cross-correlated function so that large displacements are correlated to turning angles. These mechanisms allow these animals to remain within a food patch and provide behavioral mechanisms for ARS.

Semantic trajectories can be used to study the ecology of the species of interest. In this study on fallow deer we recorded the foraging stations used by the animals and later we determine the amount of vegetal biomass of each station. According to optimal foraging theory the animal should leave in each station a prescribed amount of vegetal biomass. This was indeed observed.

Several models may explain the presence of ARS in one animal's path. Here we consider two basic, and hence potentially general, approaches to this problem. The composite CRW (CCRW) derives directly from CRW theory by assuming that an animal is able to vary its movement parameters (α_i and d_i) as a function of some specific spatial parameter. The CCRW is also called adaptive CRW.

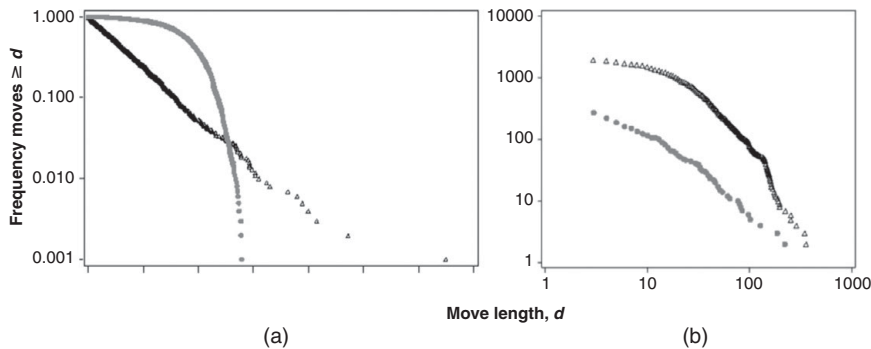


Figure 13.5 Rank-frequency plots using \log_{10} axes. (a) Models' prediction for a Brownian walk (grey dots) and a Lévy walk (triangles). (b) The observed behavior of animals in group (black) and solitary (grey dots). Given a sample of move length d_1, d_2, \dots, d_n sort these value in increasing order and rank them from 1 to n . For each d_i compute the number of distances $\geq d_i(s_i)$. Finally, s_i is plotted as a function of d_i in double-logarithmic plot.

As noted above, we have assumed till now that the probability density function (PDF) of move length, $p(d)$, exhibits finite variances. Recent literature, however, has been shifting toward distributions that have a long-fat tail (see Chapter 15). Some authors have conjectured that organismal movement is so generally heavy-tailed that the moments of the PDF are no longer finite. Lévy distributions have figured prominently in such treatments. Lévy walks are random movements where the probability of a displacement d is $p(d) = cd^{-\mu}$ for $d > d_{min}$ where $c = (\mu - 1)d_{min}^{\mu-1}$. Lévy behavior applies only to the tail of the distribution, and $P(d)$ is valid only beyond some minimal value of d ; the investigator must select an appropriate value of d_{min} for a particular data set. The scaling parameter μ has the remarkable property of being independent of the measurement units, so direct comparison can be made across studies. Application of the central limit theorem shows that for $1 < \mu \leq 3$, a sum of Lévy distributed moves is also Lévy distributed. Conversely, for $\mu > 3$ the distribution of the sum of such moves converges to a Gaussian distribution, recovering Brownian motion. Obviously, sample variances are always finite and some authors have invoked the use of truncated Lévy distribution as more realistic for actual animals.

The basic differences between a Brownian and a Lévy walk is presented in Figure 13.5a, using rank-frequency plots. The rank-frequency plot is recommended to discriminate between Brownian and Lévy walks. The plot demonstrates that in a Brownian walk the fraction of very long moves falls rapidly to 0 while in a Lévy walk such a decrease is much slower and follows a linear pattern, indicating that a Lévy distribution is characterized by a “fat” tail.

The presence of CCRW and LW in fallow deer has been studied by Focardi et al. (2009) and it was shown that solitary fallow deer adopted a LW tactic while

animals in groups performed a Brownian motion, indicating that living in group reduces foraging efficiency (Figure 13.5b). This effect has indeed demonstrated that foraging rates decrease with group size. The mechanisms seem linked to a variation in the foraging efficiency of animals moving inside the group, which forage less than deer moving on the borders of the group. Because with large groups there is a larger proportion of animals inside the group, the average foraging efficiency is lower in large groups. This is the price to pay to enjoy the better protection afforded by large groups. Besides that both LW and CCRW may simulate the process of area-restricted search, the two models are basically different: Lévy walks are scale-free, while the two-level scale-specific CCRWs are a mixture of two movements characterized by specific scale (typically, inter- and intrapatch movement).

13.3 Conclusions

The last years have seen impressive development in the study of animal movement. Important breakthroughs originated from the technological development of biologging devices, which have allowed researchers to collect huge amounts of movement data from a large number of animals. All possible scales of analysis (from migration to food search) were investigated. In parallel with technological development we have witnessed substantial improvements in data storage and data mining and statistical models became more and more flexible. An application of multiscale movement analyses to the understanding of animal ecology was provided in relation to post-reintroduction displacement of elk (*Cervus elaphus*). This analysis documents behavioral shifts at different spatial and temporal scales that permitted to these animals to survive in a difficult environment.

Despite these improvements, the analysis of animal movements remains challenging and requires important progress.

The first, apparently banal, observation is that biological samples are often biased: to fit animals with a biologger, the animals have to be captured, thus incurring behavioral biases, for example, higher ability to escape or a higher attitude to use baited traps than the average individual. Further, the device itself may modify the behavior of an animal because of its weight and shape. Moreover, the sample size might be inadequate to express the variability of the population. Last but not least, the devices determining animal positions are subject to errors as are all instruments. Finally, a movement is a continuous process but we are forced to sample it in a discrete manner. Careful experimental design is strongly advertised.

As also noted in Chapters 1 and 2, it is quite important to use semantically rich trajectories. Of course the use of biologically meaningful attributes can enhance our understanding of animal behavior and data enrichment is an important future task to be developed with the use of innovative technology. In the

example of fallow deer, the possibility of observing deer behavior while recording spatio-temporal positions allowed us to understand whether or not foraging behavior was optimal and to evaluate trade-offs between foraging efficiency and protection, a factor that dominates the life of these animals.

The analyses of animal movement can be done through two different approaches: mechanistic and statistical. A mechanistic model needs data collected at a spatial and temporal scale compatible with the (behavioral) scale at which the organism takes its own decisions or interacts with its environment, for example, selection of food items or habitat, predator avoidance, interaction with conspecifics, and so forth. The chance of collecting such data depends on the context (e.g., field or experimental settings, studied species) and on the development of appropriate technologies. For instance, the recent release of video camera collars will allow researchers to collect data on foraging with an unprecedented resolution on large mammals. However, the step of analysis allowed by actual technology is often coarser than the typical behavioral scale and the use of mechanical statistical models is appropriate to perform an analysis of movement patterns, which can give us relevant insights in the processes of spatial distribution of animal population, such as dispersal and habitat exploitation. On the other hand, the choice of appropriate models to analyze movement data is dependent on the sampling design used by the researcher, which, in its turn, depends on trade-offs among costs of tags, costs of capture, weight of devices, and so forth.

We expect that important developments may arise by the application of methods from statistical mechanics to animal movement as suggested in the recent book by Viswanathan et al. (2011). There are many instances of transfer of methodologies from statistical mechanics to ecology; since Okubo's book, several types of random walks have been first developed in physics and then applied to organismal movement. Even results relative to diffusion are shared in the two disciplines. This has not been a swift process, as methods and ideas that are well established in statistical mechanics have demonstrated to be unsuitable or problematic for applications in ecology. The study of animal movement is a challenging task: animal behavior is dictated by drives that have evolved over a long time. On the other side, the natural environment, influencing animal movements, is highly heterogeneous in time and space. Interdisciplinary research between engineers, physicists, ecologists, and ethologists is more than a rhetoric plea: it is the key to relevant breakthroughs in the future.

13.4 Bibliographic Notes

The literature on animal movement is immense. We have kept citations to a bare minimum. An useful review of old literature can be found in [Fraenkel and Gunn \(1961\)](#). [Okubo \(1980\)](#) represents the passage of a number of models and concepts

from physics to biology in order to develop a more quantitative understanding of animal movement. [Okubo \(1980\)](#) mainly uses an Eulerian approach by studying the “average” movement. For a discussion of the differences between Eulerian and Lagrangian approaches, see [Smouse et al. \(2010\)](#). In [Alt and Hoffman \(1990\)](#), one can find a very useful glossary, many examples of studies on solitary and social organisms, descriptions of simulation methods, and a first mention of Lévy walk in relation to animal movement. The book by [Turchin \(1998\)](#) is the best reference text on the subject. It deals with both data analysis and modeling and presents a wide review of the literature. The beginner is suggested to start with this book. [Gould and Gould \(2012\)](#) are a useful reference for animal navigation. [Nathan et al. \(2008\)](#) introduced “movement ecology” theory. Hierarchical resource selection was originally developed by [Johnson \(1980\)](#). The presence of Lévy walks in nature and the methods to discriminate Lévy walks and Brownian motion have determined much controversy. A short presentation of this debate can be found in [Smouse et al. \(2010\)](#), and a more detailed discussion is given by [Viswanathan et al. \(2011\)](#). Finally, a comprehensive description of recent technology, data management, and analysis issues in the study of animal movement, mainly driven by the use of GPS-based devices, is offered in the Thematic issue of [Cagnacci et al. \(2010\)](#). The study on elks is from [Fryxell et al. \(2008\)](#). The experimental studies on fallow deer can be found in [Focardi et al. \(2009\)](#), and information about biologgers in relation to animal movement studies is described by [Tomkiewicz et al. \(2010\)](#). [Urbano et al. \(2010\)](#) is a useful reference for animal movement databases.

14

Person Monitoring with Bluetooth Tracking

Mathias Versichele, Tijs Neutens, and Nico Van de Weghe

14.1 The Difficult Nature of Measuring Human Mobility

Human mobility on different spatial and temporal scales affects many processes taking place in our world. Although few will disagree that the large increase in human mobility during the twenty-first century has improved our general quality of life, it is increasingly confronting us with some of its more negative implications as well: congestion in and around densely populated areas by daily commuter traffic and the resulting strain on our environment, safety issues arising from the gathering of large crowds in relatively small areas, sudden risks of global pandemics and the difficulty of containing them, and so on. As such, an increase in human mobility should be accompanied by a deeper understanding of the processes governing these movements in order to better mitigate their negative implications.

A starting point in learning more about these movements is adequately measuring them. Until recently, this has been quite problematic. Qualitative methods such as shadowing and the collection of travel diaries are known to be error prone and labor intensive. An alternative method of tracking people in smaller-scale settings is through video surveillance systems. Despite technological advancements in the last decade, using cameras to reconstruct the movements of a large number of people in a realistic environment remains very difficult. Correctly identifying trajectories of individuals in one camera view is already nontrivial due to interactions between moving objects, changing illumination in outdoor environments, and so on. Reconstructing trajectories over multiple camera views is even more challenging and to date remains somewhat of a scientific fiction.

A third way of measuring human movement is through the use of proxies: objects whose movements are in some way linked to the movement of humans and can thereby serve as indicators of these movements. A rather unusual example of this is the tracking of one-dollar bills throughout the United States (as

already mentioned in Chapter 15), which could potentially offer insights in how people move from one state to another over time. In the end, however, it is the rapid development of positioning technologies such as GPS, and the growing penetration of these technologies in mobile devices, such as car kits and mobile phones, which acts as the main catalyst in a new and burgeoning research area. After all, these devices can be regarded as very good proxies for capturing human mobility.

As a result, there has been a rapid increase in the amount of mobility data sets. As these data sets tend to be large, the sheer volume of data confronts researchers with difficulties in extracting interesting and relevant knowledge. While the importance of this issue – often used in paradigms such as “the data avalanche” – is undeniable, it should also be stressed that human mobility is not always as easily measurable as might be conceived at first sight. First and foremost, persons can move around in a variety of ways. As more and more vehicles are equipped with GPS navigation kits, the movement data from these vehicles are already used for purposes such as the real-time monitoring and prediction of traffic jams. Capturing the movements of cyclists and pedestrians, however, is already significantly more difficult. Mobile phones usually remain very close to their owners at all times, so they are the most obvious candidate proxies. Because mobile operators keep records of telephone calls making use of their cell towers, it is possible to reconstruct movements of phones by mining their call logs. This methodology – usually called “mobile positioning” – delivers very large mobility data sets that have already been used to study regional movements. GPS loggers carried around by a test audience form an alternative way of measuring the movements of people. Because the resulting trajectories are usually very accurate and participants can also be surveyed before or after their cooperation, this method is becoming increasingly popular among scientists.

Both methodologies, however, have their deficiencies. First, the cooperation with mobile operators for mobile positioning data sets has proven to be difficult. More importantly, the spatial accuracy of this methodology (at best a few hundred meters in urban settings) is insufficient for studying human mobility on smaller scales. Alternatively, the distribution and recollection of GPS loggers among a test audience is labor intensive and possibly expensive, which will automatically result in a smaller sample size. Additionally, research projects making use of this technology will essentially be limited to outdoor environments where shadowing due to dense urban environments can potentially lower data quality as well.

The difficult nature of capturing human mobility on smaller (in this context subregional) scales shows that, despite the undeniable data avalanche confronting researchers, there remain challenges in capturing movement data besides processing them. In short, there is a need for a methodology that can measure human movement on a small scale in a cost- and labor-effective way, in



Figure 14.1 Components of a Bluetooth scanner for tracking purposes: computational unit (1), power source (2), USB cable (3), class 2 Bluetooth sensor (4), class 1 Bluetooth sensor (5), and different types of external antennas (6, 7).

a wide variety of environments and for a sufficiently large sample size in order to make representative statements for the entire population.

14.2 How Bluetooth Offers an Alternative Solution

In response to these issues regarding data collection and given the ubiquity of Bluetooth-enabled devices such as mobile phones and personal digital assistants (PDAs) carried around by their owners, Bluetooth technology has increasingly been suggested as a simple and low-cost alternative for the reconstruction of spatio-temporal behavior. Section 14.5 outlines some of the research that has already used Bluetooth as a tracking technology. “Discoverable” devices – and by extension their owners – can be traced by means of a unique media access control (MAC) address that is broadcasted in the Bluetooth discovery process. Because this MAC address cannot be directly linked to any personal (or other sensitive) information, individuals remain anonymous, avoiding potential privacy infringements.

14.2.1 Bluetooth Tracking Methodology

Bluetooth *scanners* – depicted in Figure 14.1 – can sense the presence of discoverable Bluetooth devices in their vicinity by continuously inquiring for nearby devices with a Bluetooth *sensor* and logging the broadcast messages sent by

responding mobile devices within the scanners communication range. Every time a device is detected, its MAC address, COD (Class of Device) code, and the timestamp of the detection are registered. Additionally, the received signal strength intensity (RSSI) of the inquiry response is logged. This intensity value is inferred from the received power level with which the response packet was detected by the scanner and is theoretically negatively correlated with the distance between the scanner and the detected device. Because some users include personal information in the friendly name of the detected device (name, phone number, etc.), it is not registered to safeguard privacy. The inquiry phase does not require an active connection between the scanner and the mobile device, so the methodology does not necessitate any cooperation of the tracked individual.

By placing Bluetooth scanners at different strategic locations, meaningful trajectories generated by mobile devices (and correspondingly by their owners) can be reconstructed. Because of the complex environmental setting and the resulting unpredictability of the propagation of Bluetooth signals, positioning is currently done through the *proximity* principle, where the position of a detected mobile device is approximated to the point position of the scanner by which it is detected. The strategic locations of the scanners are used to semantically enrich the resulting trajectories, which then become geo-localized semantic trajectories. As with any other form of sparsely sampled (sometimes also called *episodic*) movement data, the locations of mobile devices that are not within range of any scanner are unknown.

The spatial granularity of the resulting trajectories ultimately depends on the detection range of the Bluetooth scanners, and on the number and coverage of Bluetooth scanners within the study area. In theory, the detection range depends on the power class of the Bluetooth device (Class 1: 100 m, Class 2: 10 m, Class 3: 1 m). In practice, however, this range is variable due to environmental factors influencing (blocking, reflecting, etc.) radio signals leading to a detection region with a fuzzy border. The temporal granularity cannot be predicted either because the Bluetooth scanners register detections whenever they arrive instead of using a fixed sampling interval. Devices within a direct line of sight with a sensor will usually lead to new detections every few seconds.

14.2.2 Preprocessing and Software

The raw tracking data consist of log files – named after the combination of the scanner and the MAC address of the sensor – containing log lines with the following format: *timestamp of detection, MAC address of the detected device, COD code of the detected device, RSSI of detection*. In order to obtain a compressed data set, the scanners are programmed to create a second set

```

voyage103_01:A3:B5:0A:4B:42_rssi.log
20100720-175338-CEST,20:21:A5:45:40:40,5898756,-81
20100720-175340-CEST,20:21:A5:45:40:40,5898756,-80
20100720-175341-CEST,20:21:A5:45:40:40,5898756,-72
20100720-175353-CEST,20:21:A5:45:40:40,5898756,-78
20100720-175355-CEST,20:21:A5:45:40:40,5898756,-82
↓
voyage103_01:A3:B5:0A:4B:42_scan.log
20100720-175338-CEST,20:21:A5:45:40:40,5898756,in
20100720-175341-CEST,20:21:A5:45:40:40,5898756,out
20100720-175353-CEST,20:21:A5:45:40:40,5898756,in
20100720-175355-CEST,20:21:A5:45:40:40,5898756,out

```

Figure 14.2 Extract of logged data showing the raw time-point detection data (top) and the compressed time-interval data (bottom), depicting the compression of solitary detections into intervals leading to an abstract and structured geo-localized trajectory. This example shows one Bluetooth device (MAC address 20:21:A5:45:40:40) being detected five times. The buffer time of 10 seconds causes the raw data to be split into two separate detection time intervals (in → out). The COD code of the device (5898756) shows that this was a cell phone.

of log files during the scanning process in the following compressed format: *timestamp of detection, MAC address of the detected device, COD code of the detected device, in/out/pass*. A buffer time of 10 seconds is used to create detection time intervals from the detection time points. *In* is written when a device enters the detection range of the sensor, and *out* is written when the device leaves the range. *Pass* is used for solitary detections with no prior or later detections within 10 seconds. The principle of this logging system is depicted in Figure 14.2. In correct terminology, this compression actually transforms a geo-localized semantic trajectory into an abstract and structured semantic trajectory where individual detections are compressed into detection intervals representing the presence of a mobile device within a scanner's range during a certain time interval.

This compressed interval-based representation adhering to the proximity principle is then imported into our processing environment for further analysis. Figure 14.3 shows a screenshot of this environment, dubbed a Geographical Information System for Moving Objects (GisMo). It was developed in Java as a desktop client.

14.3 Case Studies

To give a general overview of the merits of the Bluetooth tracking methodology, we will show three case studies that have been carried out in three different application contexts: crowd management and safety at a mass event, and marketing insights in two retail environments: a professional fair and a shopping mall.

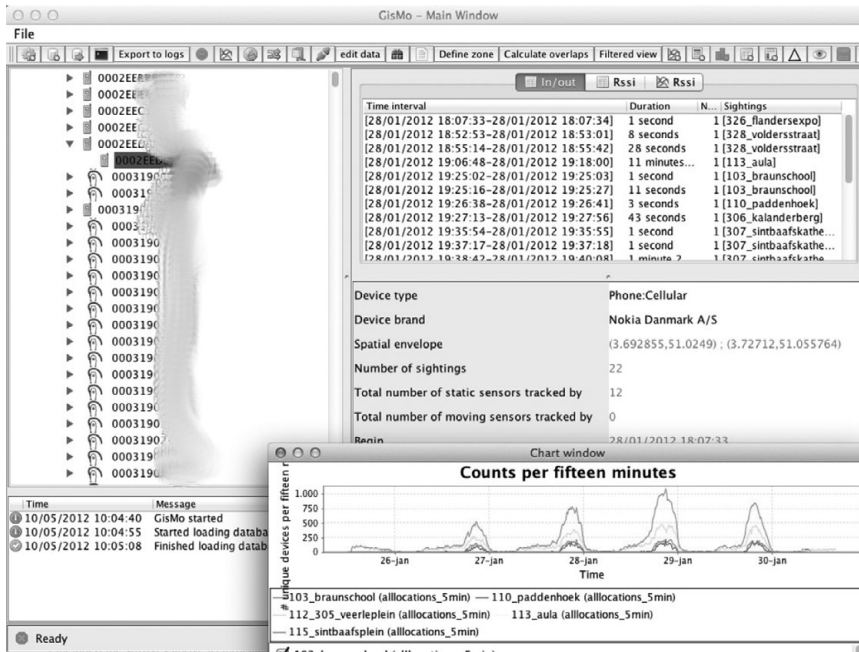


Figure 14.3 Screenshot of the GisMo analysis environment. MAC addresses have been partly smudged for privacy reasons.

14.3.1 Crowd Management and Safety at a Mass Event: Ghent Festivities 2010 and 2011

Because Bluetooth allows for nonparticipatory, unannounced, and simultaneous tracking of a large number of individuals, it is particularly useful for monitoring visitor flows at mass events. However, and despite this potential, only a few studies using Bluetooth tracking at mass events have been reported in the academic literature (some are described in Section 14.5). Hence, the methodology was tested at the Ghent Festivities, one of the largest outdoor cultural events in Europe, which lasts for 10 days in July and attracts around 1.5 million visitors annually. This setting offers a challenging test bed in terms of crowd size, duration of the event, and spatial extent of the study area (the historic city center of Ghent comprises around 4.5 km²). Because of the size and the open nature of the event – most activities in the festival are free, and there are no explicit entrance or exit points – collecting objective numerical data on visitors is challenging. The resulting lack of quantitative data acts as a bottleneck for research into the spatio-temporal dynamics of visitor movements. Exemplary to this is the issue of calculating the total number of visitors that attend the festival, which has traditionally been estimated by using proxy variables such as the daily amount of waste collected and the number of tram or bus tickets sold. As such, estimations

vary but the general consensus is that approximately 1.5 million (nonunique) visitors attended the festival in 2010. Other than this rough figure and the use of video technology by the police department to give a qualitative indication of crowdedness or other safety issues, little is known about the general movement patterns of these visitors within and around the festival site: how long they stay at the festival, the number of days they visit the festival, how they reach the event, and so on.

Given the limited range of Bluetooth scanners and the size of the event, a full coverage of the entire study area was impossible from a practical point of view. Instead, a careful selection of strategic coverage sites was made after consultation with local policy makers and urban experts, with the purpose of collecting as many significant individual movements as possible. In 2010, 22 locations were covered, including the large public squares in the city center, a selection of points of access into the event zone, two train stations, and a tram station located next to a park and ride facility. In 2011, we were able to capture visitor movements in the center in a more finely grained way by employing 43 scanners exclusively in and around the center of the city.

As overcrowding is usually regarded as the main danger at mass events, we started by using Bluetooth tracking as a counting methodology instead of a tracking methodology as such. In order to extrapolate from counts of detected devices to real numbers of people within the detection range of a scanner, we need to know the fraction of visitors that are detected by our system (corresponding to individuals carrying devices that have a discoverable Bluetooth interface). To this end, we compared visual head counts with the number of unique Bluetooth devices in a number of narrow passageways during a certain amount of time (usually 15 minutes), and divided the latter by the former. This penetration rate – also referred to as *detection ratio* – usually varies slightly from event to event, but in 2010 it amounted to $11.0 \pm 1.8\%$. Using this figure, we could extrapolate and roughly estimate crowdedness levels. As an example of this use as a counting methodology, the daily and hourly variations in crowdedness of the event zone are illustrated in Figure 14.4.

The hourly variation is characterized by a very smooth curve with sharp troughs in the morning (usually around 7 A.M.). The peaks are also usually sharp and situated around 11 P.M. except for on days 2, 5, and 9, where a broader peak in the late afternoon is observed. These correspond to two Sundays and the national day of Belgium (July 21st), and these days are known to attract more daytime visitors (such as working couples with children). As a result, the sharp peaks around midnight do not appear because of the relatively greater crowdedness earlier in the afternoon. The three busiest days are immediately visible, with the fourth day being the most crowded with almost 10,000 detected phones or around 90,000 unique visitors in the festivities zone between 11 P.M. and 12 A.M. To aggregate over daily periods, we had to carefully consider how

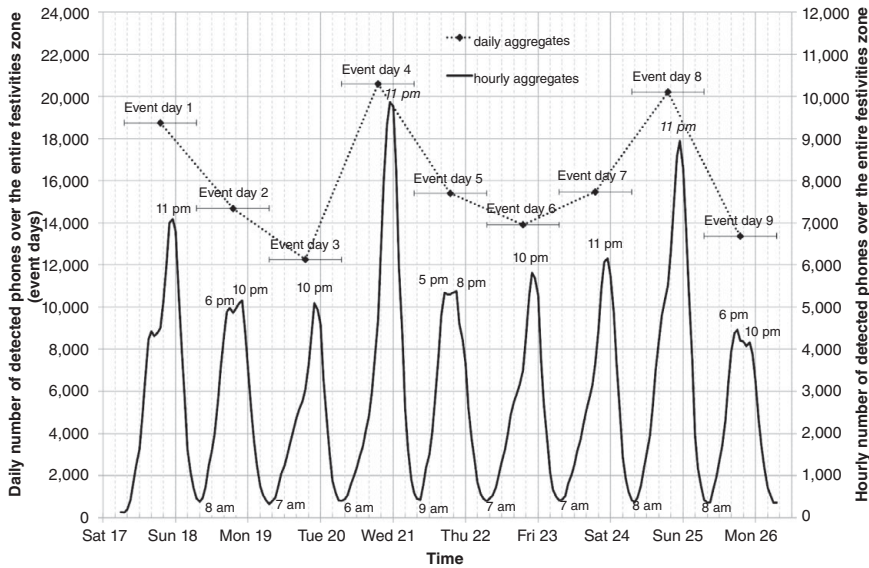


Figure 14.4 Daily (dashed line, event days starting and ending at 7 A.M.) and hourly (solid line) number of detected phones over the entire Ghent Festivities 2010 event zone as an indicator of crowdedness. Solid vertical gridlines point to midnights, dashed vertical gridlines are plotted every 4 hours.

to define a day. Looking at the hourly crowdedness, it is clear that it does not make much sense to define days starting and ending at midnight because that is generally the most crowded period of the day. Doing so would cause the Bluetooth observations to be segmented by unnatural breaks. Consequently, we have considered the starting point of an “event day” to coincide with the on average least crowded moment of a day, that is, 7 A.M. The daily aggregates again show the three busiest days, with day 4 peaking at almost 20,500 detected phones or 190,000 visitors.

Although the number of visitors present at a certain location and time is already a good indicator for the likelihood of safety issues, the movement of visitors from one location to another offers even more insight into the spatio-temporal dynamics of a crowd. Although only flows of visitors carrying discoverable Bluetooth devices can be reconstructed, the discovered patterns and trends can aid stakeholders in making well-informed decisions regarding crowd management and security in general. By making a time series of these flow diagrams, it is possible to investigate the time dependency of certain visitor flows.

Figure 14.5 shows a visualization of such dynamic visitor flows in Google Earth, comparable to the figures presented in Chapter 8. The KML file was generated in the GisMo environment and can be animated in time. Four snapshots

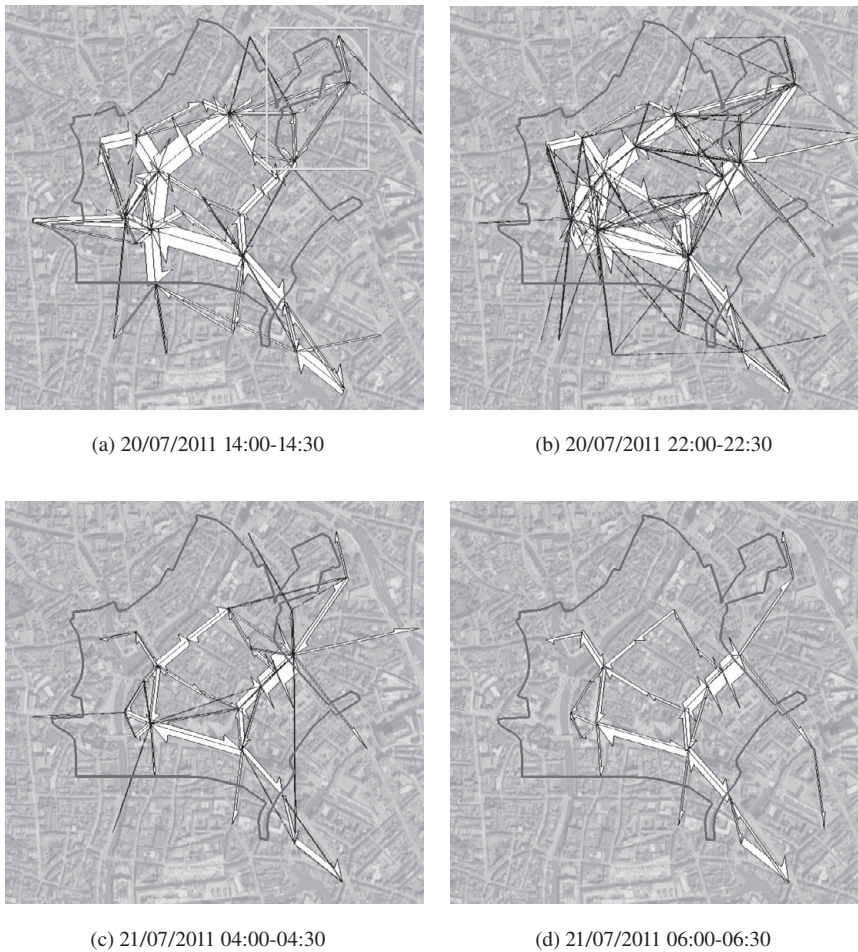


Figure 14.5 Spatio-temporal variation of visitor flows during the Ghent Festivities 2011. Four snapshots show the cumulative flows during four time windows of 30 minutes. The outer border delineates the official event zone where specific regulations are in order to make the event as safe as possible. The direction of an arrow indicates the direction of the flow; the width of the arrow indicates its size. The widths of the arrows are normalized to the size of the largest flow during each time period separately, so flows from other time windows cannot be directly compared based on this visualization.

are depicted – each depicting the cumulative flows over 30 minutes. In the afternoon (Figure 14.5a), visitor flows are quite evenly spread over the event zone, except for the area in the northeast depicted by the rectangle. Most of the large flows are balanced in their directionality, but the flows in the periphery are mainly inward oriented. Visitors regularly venture further from the center across the Leie and Lieve rivers (depicted by the oval in the west of the event zone). In the evening (Figure 14.5b), the region in the Northeast has clearly

sprung to life. Throughout the rest of the event zone, there is also an abundance of visitor flows. There is still a net inflow of visitors visible in the periphery. Later in the morning (Figure 14.5c), we see some important differences from the previous view. First of all, most flows within the event zone seem to show a net migration to the northeast, where there is a lot of activity. This is caused by nighttime visitors walking to this area after all music performances have ceased in the rest of the event zone. Additionally, flows surrounding the event zone now show a net efflux (most apparent in the southeast). Visitors generally stay closer to the center as well. Later, around dawn (Figure 14.5d), the largest flows are situated in the Northeast whereas the areas that attracted large crowds during the day are rather desolate in comparison. More importantly, most flows now point away from the northeast. This represents the ongoing egress of visitors returning home.

14.3.2 Marketing Insights in Retail Environments

As discussed above, Bluetooth tracking can be considered a helpful tool in aiding crowd management during mass events. However, the gathering of large crowds does not only cause negative consequences such as higher risks of safety issues. It also creates opportunities because large crowds represent large volumes of potential consumers when these people walk around in a retail environment. As such, marketing can be regarded as an application context for our methodology that is just as relevant as crowd safety. This is not much of a surprise as the place constitutes an essential component of the classic marketing mix, next to price, product and promotion. Traditionally, place in a marketing context can be interpreted in several ways ranging from the physical location where a product is purchased to the distribution chain linked to a product. The (changing) location of a client browsing or purchasing in a retail environment is equally relevant, however. The opportunity to measure these movements in a (semi)automatic way with modern tracking technologies has even been hailed as a “third wave of marketing intelligence.”

Visitor Movements at a Professional Fair

Fairs might not represent a retail environment *sensu strictu*, as the major aim is to showcase products or services instead of selling them, but visitor movements in these contexts are highly valuable nonetheless. Organizers of fairs often need to distribute a limited showcasing area to a large number of companies. These individual companies want to maximize their exposure, while the fair organizers want to optimize the general quality of both the visitors’ experience in general as well as the return on investment envisioned by the companies having booths at these events. Additionally, rental prices of areas occupied by exhibition stands

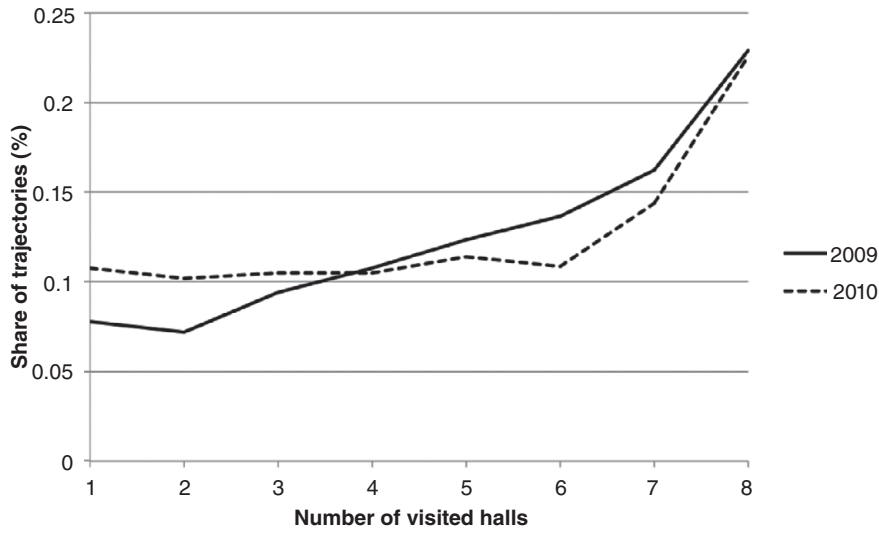
not only depend on their size but also on their location. Since certain locations are already known to attract larger portions of the crowd (“hotspots”), these will be more expensive for companies wanting to place a booth there. In the end, however, fair organizers need detailed movement data in order to give more accurate estimates of these rental prices and possibly adapt the distribution of exhibition stands based on findings extracted from these data.

In light of this application context, a cooperation was set up with a well-known fair organizer owning a large exposition venue composed of eight halls and covering an area of over 50,000 m². During two editions of a large professional catering fair (2009 and 2010), the Bluetooth tracking methodology was tested in this indoor environment. Some basic results are shown in this section.

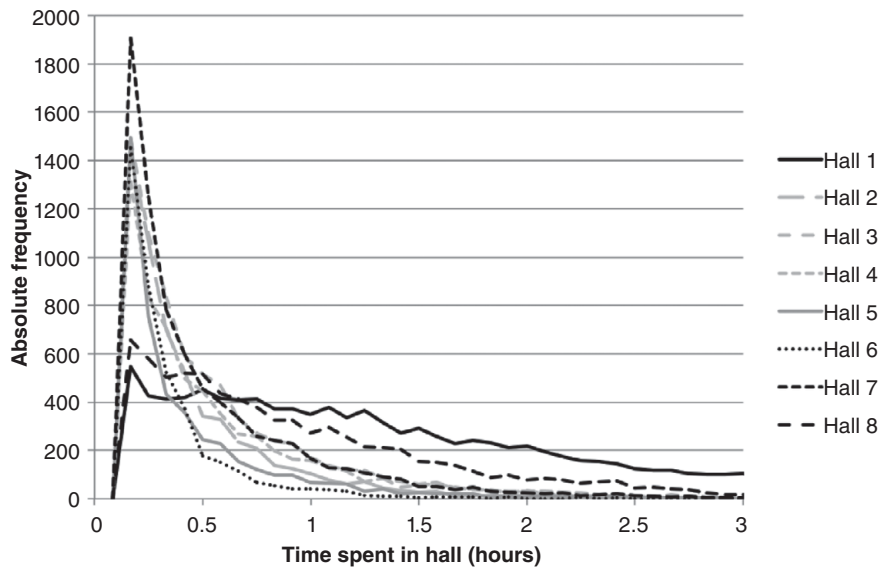
Tests showed that 35% of the visitor population was tracked, which is significantly higher than the detection ratio of around 11% during the Ghent Festivities. The most important factor contributing to this higher figure is most likely the increased penetration of Bluetooth-enabled devices in the population of catering professionals. Figure 14.6a shows the distribution of the number of halls visited per detected individual for the fairs in 2009 and 2010. The curve for the 2009 fair clearly shows a smaller share of individuals visiting four halls or less, and a higher share visiting five halls or more. In short, visitors tended to visit more halls on average in 2009 than in 2010. The histograms in Figure 14.6b show the distribution of time durations spent across the different halls. Durations of less than 5 minutes were filtered out for visualization purposes (these represent people traversing a hall instead of “visiting” it anyway). There is a clear difference in average times spent in each hall. Visitors seem to spend most time in hall 1 (which is the main and also largest hall), followed by hall 8 (which is the second largest hall). The difference between the remainder of the halls (which are all equal in size and smaller than halls 8 and 1) is smaller. Visitors spend roughly equal amounts of time in halls 7, 4, and 3, followed by halls 2 and 5. Hall 6 is on average visited for the shortest amount of time.

Customer Movements in a Shopping Mall

The value of modern tracking technologies in generating valuable marketing intelligence has already been touched upon. In order to examine the specific merit that Bluetooth tracking could hold in this context, the technology was also tested in a retail environment *sensu strictu*: a shopping mall that consists of thirty-nine stores of varying size distributed over three floors. The movement of customers from one store to another was registered during a one-month period leading up to Christmas. Scanners were also placed at the entrances and the subterranean parking lot in order to analyze visitor flows in and out of the venue. Table 14.1 shows the number of visitors that were detected in each of the stores inside the shopping hall, sorted from the most popular clothes store to



(a)



(b)

Figure 14.6 Insights delivered by Bluetooth tracking in an indoor fair environment: (a) difference in the number of visited halls between the 2009 and 2010 edition; (b) distributions of time spent in the different exhibition halls (class width of 5 minutes, durations less than 5 minutes were filtered out for visualization purposes) during the 2009 fair.

Table 14.1 *Number of Detected Visitors at Each of the Venues in the Shopping Mall During the One-Month Tracking Period, Ranked from High to Low. Venue Names have been Anonymized According to the Type of Products/Services They Offer (M: male, F: female)*

Venue	Detected Visitors	Venue	Detected Visitors
clothes_MF	8064	clothes_F.5	378
supermarket	2694	clothes_F.2	376
household_3	1964	clothes_M.1	354
household_1	1526	snacks_sweet	260
clothes_knitting	1461	lingerie_2	247
books_etc.1	1171	bistro_1	231
clothes_F.4	972	clothes_F.1	226
mobilephones_etc.1	889	bistro_2	199
cosmetics_1	810	clothes_M.2	160
shoes	799	interim_office	121
hobby	776	optician	101
snacks	717	mobilephones_etc.2	93
clothes_F.3	704	jewelry	92
home_entertainment	673	flowers	75
household_2	667	hair_salon	52
lingerie_1	588	leatherware	51
cosmetics_2	575	photo_services	41
books_etc.2	511		

a photo services store that attracted the smallest share of visitors. As is the case in most shopping malls, one can see that there are a number of dominant anchor stores accompanied by smaller stores.

As an example on how these tracking data can be mined for interesting knowledge or patterns, we will focus on *association rules* between the different shops customers visit in the same shopping trip. As such, the sequence in which shops were visited is of no importance in this analysis. Additionally, note that we cannot distinguish between customers who made a purchase in a store and customers who did not. More formally, the problem can be defined as follows. Let $I = i_1, i_2, \dots, i_n$ be a set of binary attributes called *items*. In this specific case, these items represent a customer's presence in each store. Each customer's visiting pattern constitutes a *transaction*, which contains a subset of the items in I . An association rule can then be defined as $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The itemsets X and Y are called *antecedent* and *consequent* respectively. Different measures can be used to select interesting rules from the set of all possible rules. The *support* of an itemset is defined as the proportion of transactions in the data set that contain the itemset, and the support of an association rule is defined as the support of its antecedent. The

confidence of an association rule is defined as $\frac{\text{support}(XUY)}{\text{support}(X)}$, and measures the confidence with which an antecedent can accurately predict the consequent. The *lift* of an association rule combines the previous two measures and is defined as $\frac{\text{confidence}}{\text{support}(Y)}$. As such, the lift takes both the confidence and the representativeness (support) of an association rule into account.

We used the popular WEKA data-mining platform (version 3.6.5) for a very succinct mining exercise. A preliminary run of the Apriori algorithm for the 10 rules with the highest confidence (minimum support of 0.01) shows the following output:

1. clothes_F_3=true clothes_F_4=true 238 → clothes_MF=true 201
conf:(0.84)
2. clothes_F_4=true clothes_knitting=true 267 → clothes_MF=true 217
conf:(0.81)
3. clothes_F_4=true clothes_F_5=true 220 → clothes_MF=true 174
conf:(0.79)
4. household_3=true clothes_F_4=true 258 → clothes_MF=true 199
conf:(0.77)
5. shoes=true clothes_knitting=true 221 → clothes_MF=true 169
conf:(0.76)
6. clothes_F_1=true 241 → clothes_MF=true 180 conf:(0.75)
7. clothes_M_1=true 385 → clothes_MF=true 281 conf:(0.73)
8. clothes_F_4=true 1089 → clothes_MF=true 777 conf:(0.71)
9. household_1=true shoes=true 236 → clothes_MF=true 168 conf:(0.71)
10. clothes_F_5=true 414 → clothes_MF=true 293 conf:(0.71)

The first important point to notice is that all rules contain clothes_FM as an item in their consequent. In fact, 54 out of the 64 rules found in total (minimum support of 0.01, minimum confidence of 0.3) contain this item. As this anchor store in the shopping mall attracts the majority of visitors (see Table 14.1), it appears in a large number of rules with high levels of confidence and hence also pollutes the view with rather obvious rules. Accordingly, we removed this store from the data set and reran the algorithm (minimum support of 0.005, sort by lift with a minimum lift of 1.1) in order to mine for less obvious (and hence more interesting) rules. The algorithm finds 266 rules, out of which the 20 top rules are shown below:

1. clothes_F_3=true clothes_F_4=true 238 → clothes_F_5=true 88
conf:(0.37) <lift:(14.9)> lev:(0) [82] conv:(1.54)
2. clothes_F_5=true 414 → clothes_F_3=true clothes_F_4=true 88
conf:(0.21) <lift:(14.9)> lev:(0) [82] conv:(1.25)
3. clothes_F_4=true 1089 → clothes_F_3=true clothes_F_5=true 88
conf:(0.08) <lift:(10.62)> lev:(0) [79] conv:(1.08)

4. clothes_F_3=true clothes_F_5=true 127 → clothes_F_4=true 88
 conf:(0.69) <lift:(10.62)> lev:(0) [79] conv:(2.97)
5. clothes_F_2=true 414 → clothes_F_5=true 101 conf:(0.24)
 <lift:(9.83)> lev:(0.01) [90] conv:(1.29)
- ...
14. clothes_F_4=true 1089 → clothes_F_1=true 91 conf:(0.08)
 <lift:(5.79)> lev:(0) [75] conv:(1.07)
15. lingerie_2=true 267 → clothes_F_4=true 99 conf:(0.37) <lift:(5.68)>
 lev:(0) [81] conv:(1.48)
16. clothes_F_4=true 1089 → lingerie_2=true 99 conf:(0.09) <lift:(5.68)>
 lev:(0) [81] conv:(1.08)
17. household_1=true clothes_F_3=true 162 → clothes_knitting=true 83
 conf:(0.51) <lift:(5.2)> lev:(0) [67] conv:(1.83)
18. clothes_knitting=true 1645 → household_1=true clothes_F_3=true 83
 conf:(0.05) <lift:(5.2)> lev:(0) [67] conv:(1.04)
19. household_1=true clothes_F_4=true 209 → clothes_knitting=true 103
 conf:(0.49) <lift:(5)> lev:(0) [82] conv:(1.76)
20. clothes_knitting=true 1645 → household_1=true clothes_F_4=true
 103 conf:(0.06) <lift:(5)> lev:(0) [82] conv:(1.05)

Again, clothes stores are abundant in the rules. The top 14 rules even exclusively contain clothes stores selling women's fashion. The rest of the top 20 is completed with rules that also link with a household store, a lingerie store, and a clothes store that also sells knitting accessories. Clearly, this shows that strong associations exist between stores that are focused on a more female-oriented public. It might be interesting to focus on clothes stores that sell men's fashion (clothes_M) in order to zoom in on a male audience. When we filter out the rules that do contain such a store in their itemset, we end up with the following 4 rules:

49. clothes_M_1=true 385 → snacks=true 83 conf:(0.22) <lift:(4.12)>
 lev:(0) [62] conv:(1.2)
50. snacks=true 874 → clothes_M_1=true 83 conf:(0.09) <lift:(4.12)>
 lev:(0) [62] conv:(1.08)
197. clothes_M_1=true 385 → household_3=true 111 conf:(0.29)
 <lift:(2.08)> lev:(0) [57] conv:(1.21)
198. household_3=true 2318 → clothes_M_1=true 111 conf:(0.05)
 <lift:(2.08)> lev:(0) [57] conv:(1.03)

We find associations between one men's clothing store and a snacks store and household store respectively. Although these rules are clearly less strong (low confidences), it is noteworthy that other and less trivial associations are found

in comparison with female oriented rules. The smaller number and attraction of male-oriented stores will certainly be one of the main reasons as to why we see this female bias. Clearly, more research is needed in order to mine for interesting patterns that instead of stating the obvious should provide interesting and new knowledge.

14.4 Conclusions

In this chapter, we have demonstrated the merits of Bluetooth tracking as an innovative, inexpensive, unobtrusive, and flexible methodology for measuring human mobility in a variety of contexts and environments. At mass events it can aid crowd managers by delivering quantitative data on crowd sizes and flows, and in retail environments it can extract marketing intelligence or other organizational intelligence through methods ranging from visual data exploration to data mining techniques such as association rule learning.

However, the unobtrusive nature of the tracking process resulting in large sample sizes automatically also constitutes a methodological issue: the possibility of biased results by oversampling certain segments of the total population of individuals. Adolescents with a higher education might indeed carry more Bluetooth-enabled devices than elderly people, and young children will probably never be detected. The potential difference in Bluetooth usage among different audiences might significantly influence generated insights. Accordingly, more research is needed into the use of discoverable Bluetooth-enabled devices by different population segments in order for Bluetooth tracking to evolve into a technology delivering accurate and reliable information to policy makers, crowd managers, and marketing researchers. The penetration rates we found in our experiments ranged from around 11% for a general audience to 35% for a professional fair visitor profile. In the end, a more systematic way of calculating the percentage of the population being tracked will be necessary for more reliable extrapolations in the future. Additionally, the possible influence of time and space on the detection ratio needs to be investigated.

The tentative association rule analysis with the shopping mall data only shows a very small selection of data mining possibilities with Bluetooth tracking data. Specifically for association rules, it soon became clear that there is a need for methods that can filter out more interesting rules from a larger set of less interesting rules. Intelligent visualization and/or pruning of association rules instead of solely listing them will certainly aid in this process. Besides association rule discovery, other data mining methods such as those described earlier in Chapter 6 can also generate valuable knowledge from this type of sparse movement data. They might need further modifications, however, to handle the spatio-temporal complexity of Bluetooth tracking data.

14.5 Bibliographic Notes

A review of the scientific progress in (automated) real-world surveillance by camera systems and which challenges remain to be addressed is given by [Dee and Velastin \(2007\)](#), and [Moore et al. \(2011\)](#) give a more specific example of the use of video technology for the analysis of crowd flows at mass events. The “data avalanche” paradigm was used by [Miller \(2010\)](#). The mobile positioning methodology, its accuracy, and the complex cooperation with mobile operators is discussed in [Ahas et al. \(2008\)](#). [Van der Spek et al. \(2009\)](#) discusses the use and added value of GPS tracking in different projects. More details of the Bluetooth protocol are given in [Peterson et al. \(2006\)](#). Although Bluetooth tracking is still somewhat of a peculiarity in the tracking field, there have been some reported uses in different contexts. One strain of literature focuses on mass events and how to analyze visitor flows. [Versichele et al. \(2012\)](#) used the methodology during the Ghent Festivities (Ghent, Belgium) as a counting methodology but also performed analyses on the flows, duration of stay, public transport usage, and so on. A smaller-scale feasibility test was performed during the Donauinselfest (Vienna, Austria) by [Leitinger et al. \(2010\)](#). [Stange et al. \(2011\)](#) measured the mobility of spectators of a Formula 1 race, and also focused on the spatio-temporal analysis of crowdedness and flows. Other applications that we can mention are the collection of vehicle travel time data on a highway segment by [Haghani et al. \(2010\)](#) and the deployment of mobile Bluetooth sensors in order to study complex social systems by [Eagle and Pentland \(2005\)](#). The visualization of visitor flows in Figure 14.5 is originally inspired by the work of [Tobler \(1987\)](#). The third wave of marketing intelligence is discussed by [Burke \(2005\)](#). The concept of association rule discovery was introduced by [Agrawal and Srikant \(2002\)](#), and [Bruzzese and Davino \(2003\)](#) discuss different ways of visualizing association rules as a means to extract the more interesting rules.

PART IV

**FUTURE CHALLENGES AND
CONCLUSIONS**

A Complexity Science Perspective on Human Mobility

Fosca Giannotti, Luca Pappalardo, Dino Pedreschi, and Dashun Wang

Fueled by big data collected by a wide range of high-throughput tools and technologies, a new wave of data-driven, interdisciplinary science has rapidly proliferated during the past decade, impacting a wide array of disciplines, from physics and computer science to cell biology and economics. In particular, the ICTs are inundating us with huge amounts of information about human activities, offering access to observing and measuring human behavior at an unprecedented level of detail. These large-scale data sets, offering objective description of human activity patterns, have started to reshape, and are expected to fundamentally alter, our discussions on quantifying and understanding human behavior. An impressive shift has been witnessed in statistical physics and complex system theory since the beginning of the new millennium, when the possibility of analyzing large data sets of human activities and social interactions boosted a renewed interest in the study of human mobility on one side, and of social networks on the other side.

The understanding of how objects move, and humans in particular, is a longstanding challenge in the natural sciences, since the seminal observations by Robert Brown in the nineteenth century, but it has attracted particular interest in recent years, due to the data availability and to the relevance of the topic in various domains, from urban planning and virus spreading to emergency response. A first contribution of this chapter is to provide a brief account of this body of research, with a focus on the recent results on the empirical laws that govern the individual mobility patterns: we discuss how the key variables of people's travels (such as length, duration, and radius of gyration) follow universal laws, validated against different data sets of real observations. We also discuss how predictable people's movements are, illustrating recent findings indicating that the high degree of predictability of human motion is a universal characteristic of every individual, despite the wide variety of individual whereabouts.

Next, we move from individuals to interactions – links – among individuals, and enter the domain of social network analysis. An extraordinary effort has been devoted to understanding the interconnectedness of individuals, that is, the structure of the social networks we inhabit, and how this structure influences social phenomena, such as the importance of certain individuals or groups, the diffusion of information, or the formation of communities. The second contribution of this chapter is to provide a brief account of the key findings of network science so far (what the distinctive features of real social networks compared to random networks are, how the community structure of real networks models the fabric of society, what the mechanistic processes that generate realistic networks are), to the purpose of discussing the recent results on how human mobility shapes and impacts social relations, and the other way around. Again, empirical laws were found that offer quantitative accounts of the intuition that people from the same social circles tend to co-locate in space and time more than people who are far apart in the social network. Building on this relation among social and mobility variables, it is possible to shed more light on how social networks (and mobile behavior) evolve over time.

We believe that the results surveyed in this chapter, about individual mobility laws and the relations between social ties and mobility, should become basic tools for research in various disciplines, and we envisage that the convergence of data mining research and network science research, already apparent in some of the works discussed here, will represent a strong trend in the near future aimed at combining the analytical power of statistical physics and knowledge discovery.

15.1 Models of Human Mobility

We live in an era in which understanding individual mobility patterns is of fundamental importance for epidemic preventions and urban and transportation planning. Yet human movements are inherently massive, dynamical, and complex. Indeed, on one hand, aided by modern transportation technologies, we can now travel to any place on the globe in just a day or two. On the other hand, while the mobility of our fellow species is mainly governed by mating needs and food resources, human mobility is fundamentally driven by ourselves, from job-imposed restrictions and family-related programs to involvement in routine and social activities. Therefore, quantifying the regularities and singularities behind human movements has remained an often elusive goal. Thanks to the availability of large-scale data sets generated by various domains of modern technologies, ranging from registration of dollar bills to mobile phone services and GPS devices to location-based websites, we have witnessed a proliferation of studies on human mobility.

In this section, we will start from the most fundamental models for motions, dating back to the nineteenth century. We will then describe several empirical

observations of human mobility and the new generation of mobility models, presenting to what extent real human mobility patterns deviate from those expected from simple diffusion processes.

15.1.1 Motion Models: Brownian Motion and Lévy Flights

In 1827, while he was studying sexual relations of plants, botanist Robert Brown noticed that granules contained in grains of pollen were in constant motion, and that this motion was not caused by currents in the fluid or evaporation. He thought at first that they were jiggling around because they were alive or because of the organic nature of the matter. So, he did the same experiment with dead organic and inorganic matter, finding there was just as much jiggling. The movement evidently had nothing to do with the substance ever being alive or dead, and this left him and his contemporaries with a puzzling question: What is this mysterious perpetual motion that keeps the pollen moving?

A possible explanation for the so-called *Brownian motion*¹ is that all the molecules in the fluid are in vigorous motion, and these tiny granules are moved around by this constant battering from all sides as the fluid molecules bounce off. Imagine we are in the middle of a crowd and there is a big balloon. As the individuals move around, they push the balloon from all directions: sometimes the balloon will move to the left, occasionally to the right, overall displaying a random, jittery motion like the paths in Figure 15.1. A particle of pollen behaves like a really huge balloon in the midst of a dense crowd.

Such an atomic-molecular thesis was described by Einstein, who in 1905 published a theoretical analysis of Brownian motion and showed that the mean distance reached by particles from the first collision point must grow with the square root of time. It means, for example, that after 4 seconds, the distance is only twice ($\sqrt{4} = 2$) the one found after a second, and not four times as insight would suggest. Einstein's calculations were confirmed experimentally in 1908 by physicist Jean Baptiste Perrin, who convinced even the most skeptical about the validity of the atomic-molecular hypothesis.

Before Einstein, Louis Bachelier derived independently several mathematical properties of Brownian motion, including the equation for the probability $P(x, t)$ for the position x of a Brownian random walker at time t , when the walker starts as the origin at time $t = 0$. The equation for $P(x, t)$ in one dimension is given by the *diffusion equation*, with a Gaussian solution. Therefore, a Brownian motion is basically a random walk with a normal distribution for the position of the random walker after a time t , with the variance proportional to t . It means that

¹ The first observation of Brownian motion was reported in 1785 by the Dutch physician Jan Ingenhousz. However, Brown was the first to discover the ubiquity of the phenomenon.

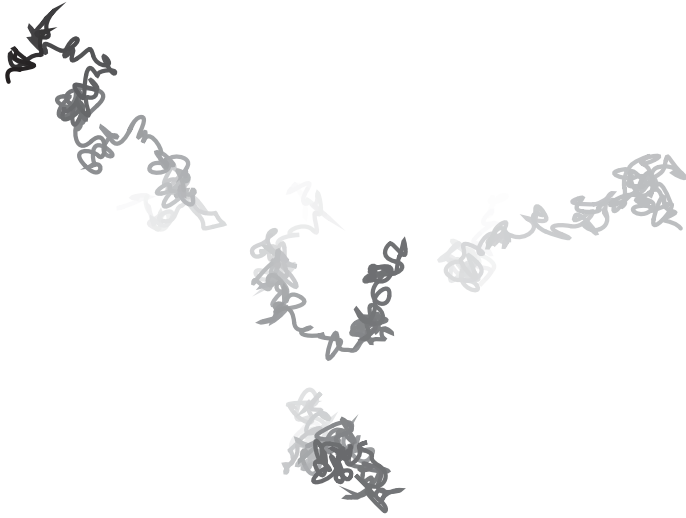


Figure 15.1 Some examples of Brownian motions.

random walkers tend to travel roughly the same distance between sightings. However, there are situations in which equations for Brownian motion are no longer applicable. An example occurs if the jumps are of very large distances: this is the case for some animal movements. Measurements on albatrosses, monkeys, and marine predators suggested that animal trajectories are different from the Brownian motion, and they are better approximated by the so-called *Lévy flight*. The French mathematician Paul Lévy investigated in the 1930s the mathematics of random walks with infinite moments. A random walk of N steps is a sum of N independent and identically distributed random variables with mean $\mu = 0$ and variance σ^2 , that is, $S_N = X_1 + X_2 + \dots + X_N$. Lévy posed the following question: when does the probability distribution $P_N(x)$ of the sum of N steps have a similar form as the probability distribution of a single step $p(x)$? For walks with finite jump variances, the central limit theorem implies that the overall probability $P_N(x)$ is Gaussian. For infinite variance random walks, the Fourier transform of $p(x)$ has the form $\bar{p}(k) = e^{-|k|^\beta}$ with $\beta < 2$. The Gaussian distribution (Brownian motion case) corresponds to $\beta = 2$, and the Cauchy distribution corresponds to $\beta = 1$. Therefore, Lévy flights are a generalization of Brownian motions (Figure 15.2).

When the absolute value of x is large, $p(x)$ is approximately $|x|^{-(1-\beta)}$, which implies that the second moment of $p(x)$ is infinite when $\beta < 2$. This means that there is no characteristic size for the random walk jumps, except in the Gaussian case of $\beta = 2$. It is just this absence of a characteristic size that makes Lévy random walks scale-invariant fractals.

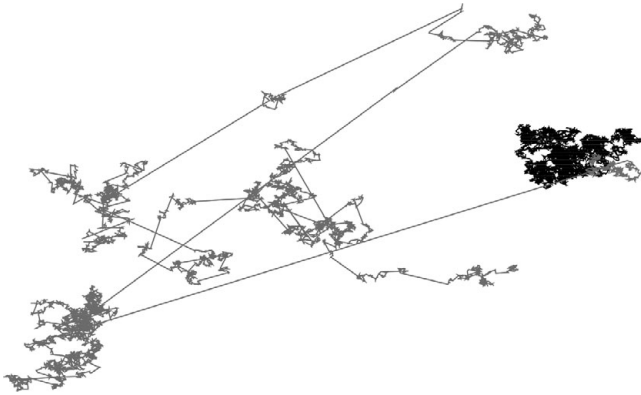


Figure 15.2 Brownian motion (darker curve on the right) is described as a random walk in which all the steps give the same contribution. A Lévy flight occurs when the trip is dominated by a few very large steps.

15.1.2 Human Mobility Patterns

Are human movements similar to those of grains of pollen, following a Brownian motion, or are they governed by Lévy flight, like the movements marine predators and monkeys? Or do they follow their own laws? To answer above questions, we need to observe humans under a microscope, like Perrin observed atoms and was able to experimentally confirm Einstein's theory. The technological era, at last, allows us to track human mobility and to test models, thanks to the exploding prevalence of mobile phones, GPS, and other handheld devices. Such devices are our social microscopes. In 2006, Dirk Brockmann and his colleagues proposed using the geographic circulation of bank notes in the United States as proxy for human traffic, based on the idea that individuals transport money as they travel. They analyzed data collected at the largest online bill-tracking Web site, www.wheresgeorge.com, and found that most bills remain in the vicinity of their initial entry, yet a small but a significant number have traversed distances of the order of the size of the United States (Figure 15.3), consistent with the intuitive notion that short trips occur more frequently than long ones. Brockmann's team calculated that the probability $P(r)$ of a bank note traversing a distance r follows a power law:

$$P(r) \sim r^{-(1+\beta)}$$

with an exponent $\beta \approx 0.6$. Moreover, they found that the typical distance $X(t)$ from the initial starting point as a function of time is a power law:

$$X(t) \propto t^{1/\beta}.$$

As we know, for Brownian motion the distance $X(t)$ scales according to the square-root law. For a power law the variance diverges for exponents $\beta < 2$



Figure 15.3 Short time trajectories of dollar bills in the United States. Lines connect origin and destination locations of bank notes that traveled for less than a week. Figure from Brockmann et al. (2006).

and it implies that bank note dispersal lacks a typical length scale resembling Lévy flights. Lévy flights are superdiffusive; they disperse faster than ordinary random walks. This discovery was a major breakthrough in understanding human mobility on global scales. In light of this discovery, in dispersal humans are similar to animals.

However, our intuition suggests that we do not move completely at random. There are regularities in our lives: most of us have a home, a workplace, a hobby. These activities necessarily shape our trajectories. Instead, if we do follow a pure Lévy flight we rarely find our way back home, but our position increasingly moves away from the initial one.

To further investigate human mobility patterns, in 2008 Barabási and his team analyzed the trajectories of 100,000 anonymized mobile phone users whose positions were tracked for a six-month period. Contrary to bills, mobile phones are carried by the same individual during his or her daily routine, offering the best proxy to capture individual human trajectories. An immediate result of the research was that the distribution of displacements Δr between a user's positions at consecutive calls is well approximated by a truncated power law:

$$P(\Delta r) = (\Delta r + \Delta r_0)^{-\beta} \exp(-\Delta r/\kappa)$$

with exponent $\beta = 1.75 \pm 0.15$, $\Delta r_0 = 1.5$ km, and some cutoff values κ . Such equation suggests that human motion follows a truncated Lévy flight, apparently confirming in a certain way observations on bank notes. However, differences

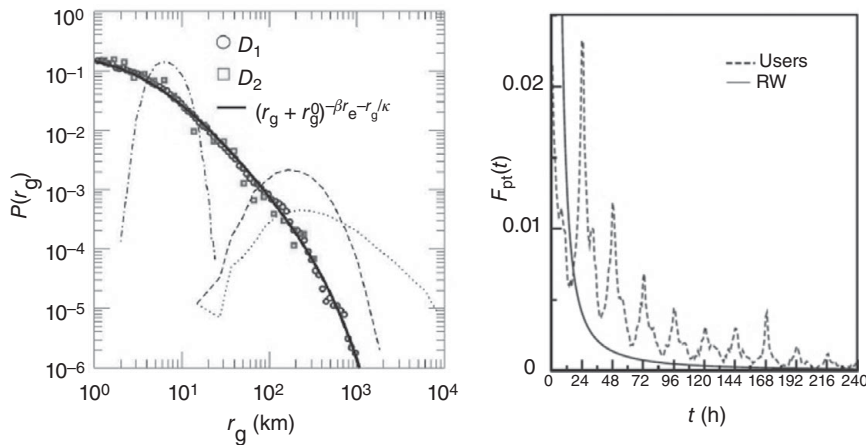


Figure 15.4 The distribution $P(r_g)$ of the radius of gyration measured for the users. The solid line represents a similar truncated power-law fit. The dotted, dashed, and dot-dashed curves show $P(r_g)$ obtained from random walk, pure, and truncated Lévy flight models. The picture on the right shows that the prominent peaks capture the tendency of humans to return regularly to the locations they visited before, in contrast with the smooth asymptotic behavior (solid line) predicted for random walks. Figure from [González et al. \(2008\)](#).

from randomness emerge from other measures. The distribution $P(r_g)$ of radius of gyration r_g , the characteristic distance traveled by a user when observed up to time t , also follows a power law, in contrast with random walks (Figure 15.4, left). So, most people usually travel in close vicinity to their home locations, while a few frequently make long journeys. Furthermore, the probability $F_{pt}(t)$ that a user returns to the position where he or she was first observed after t hours shows several peaks at 24 hours, 48 hours, and 72 hours (Figure 15.4, right), capturing the recurrence and temporal periodicity inherent to human mobility.

The most important result was the finding that, after appropriate rescaling aiming to remove the anisotropy and the r_g dependence, all individuals seem to follow the same universal probability distribution $\tilde{\Phi}(\tilde{x}, \tilde{y})$ that an individual is in a given position (x, y) (Figure 15.5b). Individuals display significant regularity, returning to a few highly frequented locations, such as home or work. This regularity does not apply to the bank notes: a bill always follows the trajectory of its current owner; that is, dollar bills diffuse, but humans do not. Song et al. 2010 extended the experiment to a larger data set and measured the distribution of the visiting time (the interval Δt a user spends at one location). The resulting curve is well approximated by a truncated power law with an exponent $\beta = 0.8 \pm 0.1$ and a cutoff of $\Delta t = 17$ hours, which the authors connected with the typical awake period of humans. The number of distinct locations $S(t)$ visited by humans is sublinear in time, well approximated by $S(t) \sim t^\mu$ with $\mu = 0.6 \pm 0.02$, that indicates a decreasing tendency of people to visit previously unvisited locations.

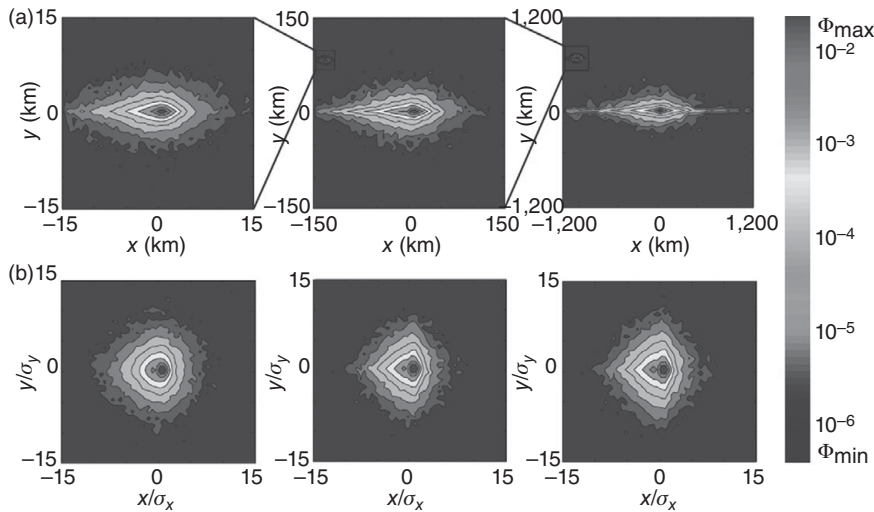


Figure 15.5 (a) The probability density function $\Phi(x, y)$ of finding a mobile phone user in a location (x, y) in the user's intrinsic reference frame. The three plots, from left to right, were generated for 10,000 users with: $r_g \leq 3$, $20 \leq r_g \leq 30$, and $r_g > 100$ km. The trajectories become more anisotropic as r_g increases. (b) After scaling each position, the resulting probability distribution has approximately the same shape for each group. Figure from Song et al. (2009).

Moreover, the visitation frequency, that is, the probability f of a user to visit a given location, is rather uneven, resulting in a Zipf-like visitation frequency distribution $P(f) \sim f^{-(1+1/\zeta)}$.

15.1.3 Predictability of Human Mobility

What is the role of randomness in human behavior and to what degree is human behavior predictable? This question is crucial, because the quantification of the interplay between the predictable and the unforeseeable is very important in a range of applications. From predicting the spread of human and electronic viruses to city planning and resource management in mobile communications, our ability to foresee the whereabouts and mobility of individuals can help us to improve or save human lives. In 2009, Song et al. 2009 provided a quantitative evaluation of the limits in predictability for human walks, using a 3-month-long mobile phone data set of about 50,000 individuals. The authors defined three entropy measures: the random entropy S_i^{rand} in the case of location visited with equal probability; the entropy S_i^{unc} that depends only on frequencies of visits; and the real entropy S_i that considers the probability of finding particular time-ordered subsequences in the trajectory. To characterize the predictability across the user population, they determined these three entropies per each user i , and

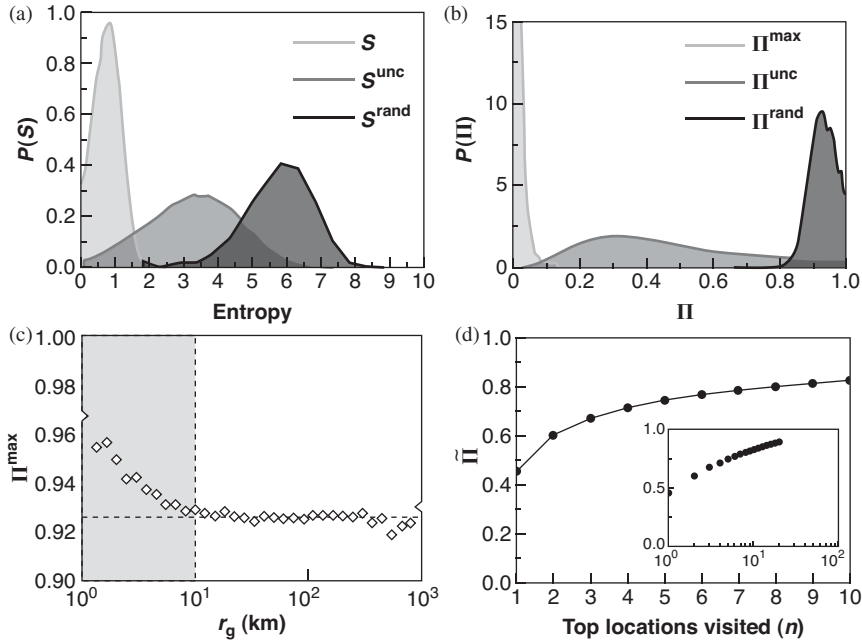


Figure 15.6 (a) The distribution of the entropies S , S^{rand} , and S^{unc} across 45,000 users. (b) The distribution of Π^{max} , Π^{rand} , and Π^{unc} across all users. (c) The dependence of Π^{max} on the user's radius of gyration r_g . For $r_g > 10$ km, Π^{max} is largely independent of r_g . (d) The fraction of time a user spends in the top n most visited locations, the resulting measure $\bar{\Pi}$ representing an upper bound of predictability Π^{max} . Figure from Song et al. (2009).

calculated the distributions $P(S_i^{rand})$, $P(S_i^{unc})$ and $P(S_i)$, that is, the frequency of entropy values. As shown in Figure 15.6a, $P(S_i)$ has a peak in $S = 0.8$, indicating that the real uncertainty in a typical user's whereabouts is $2^{0.8} \approx 1.74$. It means that a user who chooses randomly his or her next location could be found on average in two locations. A big difference emerges in respect to the random entropy, for which the peak at $S = 6$ implies $2^6 \approx 64$ locations.

To represent the fundamental limit for each individual's predictability, Song et al. 2009 defined the probability Π that an appropriate algorithm can predict correctly the user's future whereabouts. If a user with entropy S moves between N locations, then his or her predictability is bounded by the maximal predictability $\Pi^{max}(S, N)$. For a user with $\Pi^{max} = 0.2$, this means that, no matter how good the predictive algorithm is, only in 20% of the time can we hope to predict his whereabouts. They determined Π^{max} separately for each user and found that the distribution $P(\Pi^{max})$ is peaked around $\Pi^{max} \approx 0.93$. Figure 15.6b highlights that Π^{rand} and Π^{unc} are instead ineffective predictive tools.

Despite the apparent randomness of the individual's trajectories, in a historical record of the daily mobility pattern of the users there is a potential 93%

average predictability in user mobility, an exceptionally high value rooted in the inherent regularity of human behavior. The most surprising is the lack of variability in predictability across the population, obtained by explored impact of home, language groups, population density, and rural versus urban environment. Although the population has an inherent heterogeneity, the maximal predictability Π^{max} varies very little; there are no users whose predictability would be under 80%.

Knowing the history of a person's movements, the advanced pattern mining techniques described in Chapters 6 and 7 can be used to find patterns and regularities in human mobility, and to foresee his or her current location with extremely high success probability.

15.2 Social Networks and Human Mobility

In the previous section we presented the evolution of the study on human mobility, describing the main patterns and models that characterize the mobility behavior of individuals. Here, we take a step further in our journey of understanding human behavior by focusing on the interplay between human mobility and social networks, with the purpose of highlighting to what extent human movements affect social dynamics, and how social interactions influence the way people move.

We will first present a brief overview of network science and its growth in the last decade, and then we will focus on recent developments and discoveries regarding the interplay between the social world and the mobility of people.

15.2.1 Introduction to Network Science

Network science is a truly interdisciplinary field that examines the interconnections among diverse physical, engineered, information, biological, cognitive, semantic, and social systems. In mathematical terms, a network is represented by a graph $G = \{V, E\}$, where V is a set of n nodes and E is a set of edges that connect V . According to the definition, any system of interacting elements can be represented as a network. The mode of thinking of complex networks was traditionally dominated by random graph theory, first proposed by Erdős and Rényi in the 1950s. The random graph model presented a simple realization of a network: we start with N disconnected nodes, and randomly connect every pair of nodes with probability p , yielding a graph with $pN(N - 1)/2$ edges. As data regarding wiring diagrams of real systems started being collected by computer programs in late 1990s, topological information about real networks became increasingly available, prompting many scientists to ask a fundamental question: are real networks, from cell to Internet, truly random? Over the past decade, we have witnessed dramatic advances along this direction, leading to

the discovery that despite the intrinsic distinctions in the nature and functionality of the nodes and their interactions, many real-world networks follow highly reproducible patterns. There are three most studied properties that characterize a real network:

Average path length measures the average steps it takes for one node to reach another node in the network, also commonly referred as diameter of a network. Although real networks often consist of a large number of nodes, they have a very small diameter, which is most known as the “small world” property or “six degrees of separation.” That is, individuals on the planet are separated by six degrees of social contacts. Despite its simplicity, the random graph model well captures this property, predicting the average path length $d \sim \ln N$, where N is the size of the network.

Clustering represents densely connected cliques in a network, and was formally quantified by Watts and Strogatz (1998). They introduced clustering coefficient C_i for node i , which measures the fraction of neighbors of i are also connected to each other. In the random graph model, as links are distributed randomly among the nodes, it predicts $C_i = p$. Yet in almost all real networks, the clustering coefficients are significantly higher than the random graph model prediction. To capture the pervasive clustering phenomena, Watts and Strogatz introduced the small-world model, also known as the WS model: start from a regular network, for instance a ring, in which each node is connected to its k nearest neighbors. Let us redirect links with probability p , moving one end of an edge to a new location chosen uniformly at random from the lattice. When $p = 0$, the network is a regular lattice, thus characterized by a very high clustering coefficient but a large average path length. On the other end, when $p = 1$, the network is equivalent to a random graph. As we start to increase p from 0 to 1, the diameter of the network quickly shrinks, while the clustering coefficients remain roughly the same. Therefore, for a wide range of p , the WS model gives rise to networks with both high clustering coefficients and small diameter.

Degree distribution, $P(k)$, measures the probability that a randomly selected node has k edges. The random graph model predicts $P(k)$ follows a Poisson distribution corresponding to a homogeneous network, where every node has roughly the same degree around $\langle k \rangle$. However, a variety of real networks, spanning from the Internet and WWW to scientific citations and actor collaborations, exhibit the “scale-free” property, a highly reproducible pattern not accounted for by either random graph model or WS model. That is, $P(k)$ follows a power law $P(k) \sim k^{-\gamma}$. This result indicates that real networks are rather heterogeneous: most nodes in the network have very low degree, although there are a notable number of nodes with a large number of connections. Think about Yahoo! for the Web, ATP protein for metabolic networks, and Heathrow for air traffic network. To explain the possible origin of the observed scale-free property, Barabási and

Albert (1999) introduced the scale-free model (or BA model) by viewing the network as a dynamical object that evolves with addition of nodes and links to the system, in strong contrast to the static models that dominated the literature before. Imagine an initial network of a small number of nodes m_0 . At each time step we add a new node with m edges that links the node to m different vertices already present in the network. The probability that a new node will be connected to node i depends on the connectivity k_i of that node. After t time steps the model leads to a network with $t + m_0$ nodes and mt edges. This network evolves into a scale-invariant state with the probability that a node has k edges following a power law with exponent $\gamma = 3$.

In addition to the measures listed above, the concept of *tie strength* has attracted particular attention in the study of social networks. It was introduced by sociologist Mark Granovetter in 1973 as a “combination of the amount of time, the emotional intensity, the intimacy (mutual confiding) and the reciprocal service which characterize the tie.” He proposed a model of society consisting of small and fully connected circles of friends, linked by strong ties. Weak ties connect the members of these intimate circles to their acquaintances, who have strong ties to their own friends. Since weak ties act as bridges between separate “social micro-worlds,” they play a crucial role in any number of social activities, such as the spreading of information, ideas, and diseases, or in finding a job. Conversely, strong ties link persons in intimate and tight communities, affecting emotional and economic support.

The existence of a local coupling between tie strengths and network topology is confirmed by recent research, which exploits the huge quantity of human interactions recorded by modern tools and technologies. A study conducted by Onnela et al. analyzed a huge data set that stores the mobile phone interaction of millions of individuals in a time period of 18 weeks. The researchers inferred a social network from data connecting two users with a link if there had been at least one reciprocated pair of phone calls between them, and defining the strength of a tie as the aggregated duration of calls. Consistent with Granovetter’s hypothesis, the majority of the strong ties were found within highly connected communities, indicating that users tend to talk for most of their time with the members of their immediate circle of friends. In contrast, most links connecting different communities were weaker than the links within the communities. Moreover, as a consequence of the topological structure of the network, removing the weakest links leads to a rapid network’s sudden disintegration, while removing first the strongest ties shrinks the network but will not precipitously break it apart.

The interesting findings discovered by the Onnela et al. study, together with those of more recent works, confirm the importance of tie strength in study of networks, suggesting that weak and strong ties play a different but crucial role in the understanding of many dynamic processes regarding our society.

15.2.2 Interplay between Human Mobility and Social Networks

Recent advances on human mobility and social networks have turned the interplay between these two aspects into a crucial missing chapter in our understanding of human behavior. To make progress in this direction requires large-scale data that simultaneously capture dynamic information on individual movements and social interactions. Thanks to the increasing availability of mobile phone data sets and location-based online social networks (LBSN, see also Chapter 16), scientists have started to look into the questions of to what extent human mobility patterns shape and impact our social ties, and how our social surroundings affect where we go. The central hypothesis here is that social interactions increase with physical proximity. Indeed, social links are often driven by spatial proximity, from job- and family-imposed shared programs to joint involvement in various social activities. These shared social foci and face-to-face interactions, represented as overlap in individuals' trajectories, are expected to have significant impact on the structure of social networks. There are three lines of inquiry in current literature: (1) geographic propinquity yields higher probability of forming a tie; (2) overlap in trajectories predicts tie formation; (3) social environment affects individual mobility.

Geographic Propinquity

The considerable influence of geographic distance on the formation, the evolution, and the strength of friendships is probably rooted in the very nature of our social brain. According to the anthropologist Robin Dunbar, there is a physical cognitive limit in the number of strong ties the human brain is able to manage, partly because it must be powered by a form of social grooming, a time-consuming activity mainly based on geographical proximity and face-to-face contact.

Recent analysis on Facebook and email data confirmed Dunbar's intuition, showing that the volume of communications is inversely proportional to geographic distance and that the probability $P(d)$ of having a friend at a certain distance decreases following a sort of "gravitational law." Although in the last decades technology has contributed to reducing distances, proximity is still important for the establishment of relevant relationships, breaking down the illusion of living in "a global village": a small world in which physical and cultural distances vanish and where lifestyle become homogeneous.

In studying the social versus geography problem, data from LBSNs proved to be very useful. Scellato et al. used information from both the social and location components of several LBSNs to identify the relation between friendship and geographic distance. They noticed a weak positive correlation between the number of friends and their average distance, and observed that the socio-spatial structure of the users cannot be explained by taking into account separately

geographic factors and social mechanisms. Cranshaw et al. (2010) studied the entropy related to LBSNs locations in order to understand how it affect the underlying social network. They found that co-locations at high entropy locations are much more likely to be random occurrences than co-locations at low entropy locations. So, if two users are only observed together at locations of high entropy (for example, a shopping mall or a university), they are less likely to actually have a link in the underlying social network than if they are observed in a place of low entropy. Moreover, users who visit locations of higher entropy tend to be more social, having more ties in the social network than users who visit less diverse locations.

Trajectory Overlap

Given that two persons have been on multiple occasions in the same geographic place at the same time, how likely are they to know each other? This is another interesting and open problem about the interplay between sociality and mobility, regarding to which extent social ties between people can be inferred from co-occurrence in time and space.

Crandall et al. (2010) studied this problem by analyzing a huge data set from the popular photo sharing site Flickr, reaching interesting and striking conclusions. They inferred a spatio-temporal co-occurrence between two Flickr users if they both took photos at approximately the same place and at approximately the same time. Rather surprisingly, they found that even a very small number of co-occurrences can lead to orders-of-magnitude greater probabilities of a social tie. Indeed, two users have nearly 5,000 times the baseline probability of having a social tie on Flickr when they have just five co-occurrences in a day in an 80-km range of distance. With the aim of a deeper understanding of the underlying phenomenon, they developed a mathematical model in which the probabilities of friendship as a function of co-occurrence qualitatively approximate the distributions they observed in the Flickr data.

Wang et al. (2011) presented a data-mining approach to the question of to what extent individual mobility patterns shape and impact the social network. Following the trajectories and communication patterns of approximately 6 million mobile phone users over 3 months, they defined three groups of similarity measures: mobile-homophily (similarity in trajectories), network proximity (distance in the call graph), and tie strength (number of calls between two users). Exploring the correlation between these measures, researchers discovered that they strongly correlate with each other. The more similar two users' mobility patterns are, the higher the chance that they have close proximity in the social network, as well as the higher the intensity of their interactions. Starting from these results, they designed a link prediction experiment, constructing the entire repertoire of both supervised and unsupervised classifiers, based either on network and/or mobility quantities. Results showed that mobility on its own

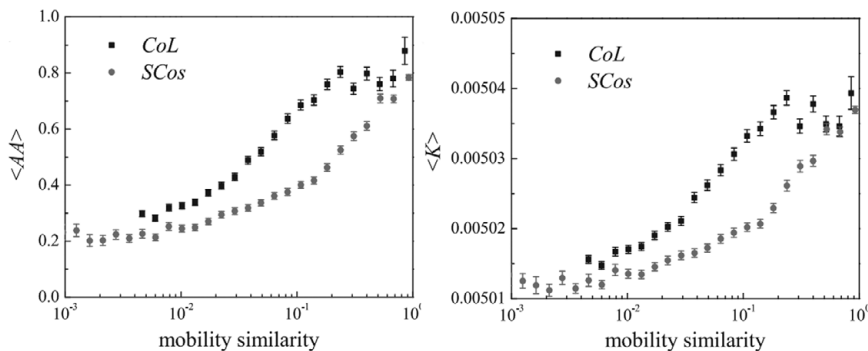


Figure 15.7 Correlations between mobility measures and Adamic-Adar coefficient (left), tie strength (right). The proximity measures used are the spatial co-location (*CoL*) and the spatio-temporal co-location (*SCos*) inferred from the trajectories of the users. Figure from Wang et al. (2011).

carries high predictive power, comparable to that of network proximity measures. By combining both mobility and network measures, in the supervised case authors obtained that only approximately one-fourth of the predicted new links were false positives, and only one-third of the actual links were missed by the predictor.

The results of the study by Wang et al. suggest that Granovetter’s theory should be integrated with a “mobility” dimension: as we can notice in Figure 15.7, the strength of a tie is correlated not only to social proximity (the extent to which people share the same community) but also to their mobility behavior (the overlapping of their spatio-temporal trajectories).

Social Environment Affects Individual Mobility

Cho et al. investigated the interaction of a person’s social network structure and his or her mobility using data sets that capture human movements from Gowalla, Brightkite, and phone location trace data. Because they uncovered a surprising increase of the effect of distant friends on an individual’s mobility, they tried to understand if friendships influence where people travel, or if it is more traveling that influences and shapes social networks. In order to measure the degree of causality in each direction, they downloaded the Gowalla social network at two different time points, t_1 and t_2 , three months apart. Considering friendships at time t_1 , they calculated a set of check-ins C_a that occurred after time t_1 and quantified the influence of sociality on future movements by measuring what fraction of them occurred within the vicinity of friends’ homes. Similarly, researchers examined the influence of mobility on creating new social ties by examining a set of check-ins C_b before time t_1 and counted the fractions of check-ins that led to creation of new friendships. They found that whereas there is, on average, a 61% probability that a user will visit a home of an existing friend,

the probability that a check-in will lead to a new friendship is only 24%. Such results were confirmed in phone call data, with the influence of friendship on an individual's mobility about 2.5 times greater than the influence of mobility on creating friendships. Moreover, data also display a strong dependency between probability of friendship and trajectory similarity, suggesting that there is a strong presence of social and geographical homophily.

The most interesting aspect of such main findings in the interplay between sociality and mobility is that they can be used to develop a model of human mobility dynamics combining periodic daily movement patterns with the social movement effects coming from the friendship network.

15.3 Conclusions

We have discussed in this chapter how the tools of statistical physics and complexity science have been applied to the study of human mobility, both focusing on individual movements and considering also the social relations among individuals. We have observed how, in both cases, general laws can be devised and empirically validated based on the newly available mobility data, shedding a new light on the underlying mechanisms behind phenomena that, at first sight, seem to be governed by chaos.

We conclude with an observation that spontaneously emerges from the current trend of research, as presented here: there is an evident push toward the convergence of network/complexity science and data mining research, a progressive merge of the two scientific communities that is only beginning today, but is steadily increasing due to the advantages of combining the complementary strengths and weaknesses of the two approaches. Why is this merge convenient?

We learned in this chapter that statistical physics and network science are aimed at discovering the global models of complex social phenomena, by means of statistical macro-laws governing basic quantities; the ubiquitous presence of power laws and other long-tailed distributions allows us to witness the behavioral diversity in society at large, such as the huge variability and individual differences of human movements. On the other hand, data mining is aimed at discovering local patterns of complex social phenomena, by means of micro-laws governing behavioral similarity or regularities in subpopulations, such as the mobility patterns and clusters discussed in Chapters 6 and 7 of this book. This dualistic approach is illustrated in Figure 15.8. In the overall set of individual trajectories across a large city we observe a huge diversity: while most travels are short, a small but significant fragment of travels are extraordinarily long; therefore, we observe a long-tailed, scale-free distribution of quantities such as the travel length and the users' radius of gyration. Despite this complexity represented in the data, mobility data mining can automatically discover travel patterns corresponding to a set of travelers with similar mobility: in such

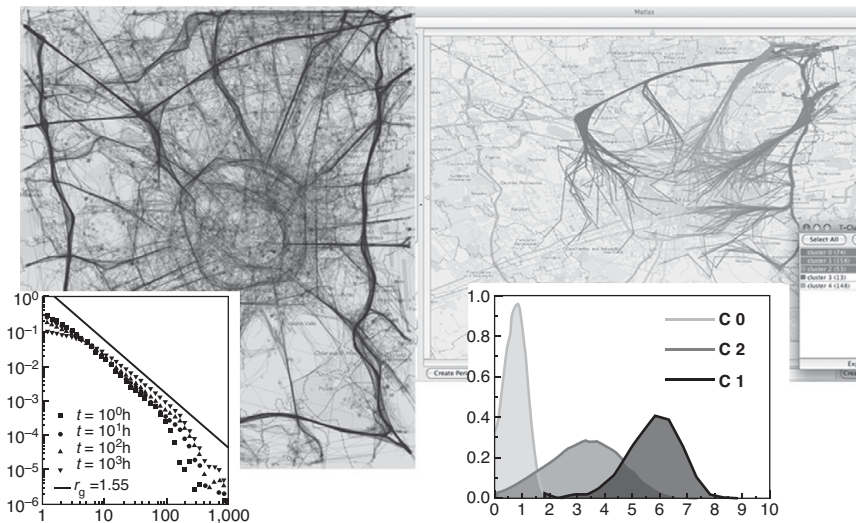


Figure 15.8 The GPS trajectories of tens of thousands of cars observed for one week in the city of Milan, Italy, and the power-law distribution of users' radius of gyration and travel length (left); the work-home commuting patterns mined from the previous data set by trajectory clustering and the normal distribution of travel length within each discovered pattern (right).

subpopulations the global diversity vanishes and similar behavior emerges. The dual scenario of global diversity (whose manifestation is the emergence of scale-free distributions) and local regularity (within clusters, or behavioral profiles) is perceived today as the signature of social phenomena, and seems to represent a foundational tenet of computational social sciences. Although network science and data mining emerged from different scientific communities using largely different tools, we need to reconcile the *macro/global* approach of the first with the *micro/local* approach of the second within a unifying theoretical framework, because each can benefit from the other and together they have the potential to support realistic and accurate models for simulation and what-if reasoning of social phenomena. This vision of convergence among computer science, complexity science, and the social sciences is shared today by large research initiatives, such as the FuturICT program.²

15.4 Bibliographic Notes

[Erdős and Rényi \(1959\)](#) is the seminal paper that introduced random graphs. The famous small-world model was presented in [Watts and Strogatz \(1998\)](#), and the first argumentations on the small-world phenomenon and the cliquishness

² <http://www.futurict.eu>

nature of society can be found respectively in [Milgram \(1967\)](#) and [Granovetter \(1973\)](#). The scale-free model was introduced first in [Barabási and Albert \(1999\)](#).

The analysis of human mobility based on dollar movements can be found in [Brockmann et al. \(2006\)](#). In [González et al. \(2008\)](#) are described the mobility patterns discovered by analyzing a rich mobile phone data set, a work later extended in [Song et al. \(2010\)](#). Limits on predictability of human mobility are presented in [Song et al. \(2009\)](#), while [Karamshuk et al. \(2011\)](#) classifies mobility patterns in temporal, social, and spatial dimensions. [Cranshaw et al. \(2010\)](#) studies the entropy related to LBSN locations in order to understand how it affect the underlying social network. [Crandall et al. \(2010\)](#) analyzed a data set from Flickr and discovered that even a small number of co-occurrences leads to high probability of a social tie. [Wang et al. \(2011\)](#) presents a data mining approach to the question of to what extent individual mobility patterns shape and impact the social network. In [Cho et al. \(2011\)](#), authors investigate the interactions between social network and mobility by analyzing data sets from location-based social networks and a mobile phone network.

Mobility and Geo-Social Networks

Laura Spinsanti, Michele Berlingiero, and Luca Pappalardo

16.1 Introduction

The social web is changing the way people create and use information. Every day millions of pieces of information are shared through the medium of several online social networks and online services with a social layer such as Facebook, Google+, Twitter, Foursquare, and so on. People have discovered a new way to exploit their sociality: from work to entertainment, from new participatory journalism to religion, from global to local government, from disaster management to market advertisement, from personal status update to milestone family events, the trend is to be social. Information or content is shared by users through the web by posting images or videos, blogging or micro-blogging, surveying and updating geographic information, or playing geographic-based games. Considering the increase in mobile Internet access through smartphones and the number of available (geo-) social media platforms, we can expect the amount of information to continuously grow in the near future. To understand the potential of this change it is worth noticing the amount of “geo-social information” produced during recent years to be a daily occurrence. The following are just few examples. In August 2006, Flickr introduced the geo-tagging feature; by 2007, more than 20 million geo-tagged photos were uploaded to Flickr. In August 2011, Flickr announced its 6 billionth photo, with an increase of 20% year-on-year over the last 5 years.¹ Similarly, Twitter was born in 2006. The most impressive performance indicator is the increasing rate of messages. In 2010, the average number of Tweets sent per day was 50 million² while in March 2012 it has increased to 340 million.³ In 2010, the geo-tagging feature was added to Twitter. Even considering that the amount of geo-enabled messages is only around

¹ Source: <http://blog.flickr.net/en/2011/08/04/6000000000/>

² Source: <http://blog.twitter.com/2011/03/numbers.html>

³ Source: <http://blog.twitter.com/2012/03/twitter-turns-six.html>

1%, this still means millions of geo-tagged messages per day. People can now be considered as sensors, producing signals on events they are directly involved in or they have witnessed. Finding, visualizing, and making sense of vast amounts of geo-referenced information will lead to a multi-resolution, multi-dimensional representation of the planet known as *Digital Earth*.

Such multi-modality and heterogeneity of online geo-referenced multimedia has encompassed challenges not seen in traditional geographic data analysis and mining and has attracted the attention of researchers from various communities of knowledge discovery in databases, multimedia, digital libraries and computer vision. However, there are clearly several challenges associated with such information: the frequent changes in the data structure, the unstructured nature of contents, the limited quality control of information, varying uncertainty of geographic information, and the semantic aspect on the content published, to mention a few issues. In the era of Web 2.0, the various geo-referenced media are mostly socially generated, collaboratively authored and community contributed. The temporal and geographical references, together with textual metadata, reflect where and when the media were collected or authored, or the locations and time described by the media content. The enriched online multimedia resources open up a new world of opportunities to discover knowledge and information related to location and our human society.

Social networks that also use and create geo-social information have grown in importance and popularity, adopting names such as location-based mobile social networks, or geographic social networks, or simply social networks with geographic features. In general, there exist several types of media with temporal and geographical references on the Internet: (1) geo-tagged photos on photo-sharing websites like Flickr, (2) geo-referenced videos on websites like Youtube, (3) geo-referenced web documents, such as articles in Wikipedia and blogs in MySpace, (4) geo-referenced microblogging websites such as Twitter, and (5) “check-in” services (users can post their location at a venue and connect with friends) such as Foursquare. Most of these services publish unsupervised (geo-spatial) content. Their importance has grown in such a way that several terms are currently circulating: *crowd sourcing*, which considers users as sensors for gathering data; *distributed intelligence*, where users are basic interpreters or preprocessors in transmitting information; *participatory science*, when citizens participate in problem definition, data collection, and data interpretation; *volunteered geographic information (VGI)*, when the contributive aspect is crucial; *contributed geographic information (CGI)*, when the geographic features are activated by the user; or just *user generated geographic content (UGGC)*, when there is a geographic reference, such as a place name, but the user-active contribution is unpredictable. Some ambiguity in the use of different terms exists, such as crowd-sourced data being synonymous with volunteered geographic

information (VGI) without distinguish different levels of participation (or “voluntariness”) when providing information. However, the term CGI could act as a broader term in this context and, therefore, it will be used in the rest of the chapter.

The voluminous geo-referenced contents on the Internet are a result of collective *geo-tagging* by the web community. Geo-tagging refers to the process of adding geographical identification metadata to media resources, such as photographs, video, articles, and web sites. The metadata usually consist of latitude and longitude coordinates and, sometimes, altitude, camera heading direction, IP address, and place name. In general, the means of geo-tagging can be classified into two types: integrated hardware (automatic), and purely software solutions (manual). GPS and other geolocation acquisition hardware provide an automatic solution for geo-tagging contents. However, till now, only a small portion of geo-referenced information is geo-tagged via these means and any geographic information mostly depends on the nature of the content. For example, most geo-referenced photos on the Internet are tagged by web users manually via a geo-tagging software platform. To facilitate easy geo-tagging, commercial media sharing services have adopted map-based tagging tools. In general, these geo-tagging tools allow a user to drag and drop photos to a location on the map. The intuitive map and user-friendly interface render the geo-tagging a simple and straightforward process. However, the major limitation of such geo-tagging processes is that, currently, no industry standards exist on tagging and storing the geo-tags of media. Most commercial media repositories store geo-tags in tag-based systems, similar to how text tags are stored. The most important consequence is that several facets of uncertainty are related to the location that can be retrieved as we describe later in the chapter.

The rest of the chapter gives an overview of existing and foreseen applications that use this CGI data with a particular focus on mobility. It then describes the problem to reconstruct trajectories from the Semantic Web and the research issues related to geographic and semantic uncertainty of this data. Several open issues still remain due to the novelty of this research area and they are described at the end of the chapter.

16.2 Geo-Social Data and Mobility

The use of geo-social data covers a wide range of possible applications, essentially all the contexts in which location (and time) plays an important role, such as health, entertainment, work, personal life, and tourism. Although we want to focus on the mobility aspects of geo-social data, we have to say that this topic is really a forefront research of latest years. The studies conducted so far have started based on several works produced on mobile phone data. Despite the

similarities between geo-social data and mobility phone data, as explained in Section 16.4.2, the conceptual framework and the characteristics of geo-social data lead to a real new branch of research. The research about this new domain is far from being exhaustive. As described in Section 16.3, trajectories resulting from geo-social data are built from a collection of sparse data points. This ends up in different groups of applications, as described here.

We can distinguish a first group of applications that use only the location from geo-social data, generally to filter the contents (message, photo, video, news, tweets, and so on) from a zone they want to analyze or about which they want to receive alerts (newsfeed mechanism). Some examples from the natural disaster field include wild fires in the United States and France, hurricanes in the United States, the 2010 earthquake in Haiti, and floods in the United Kingdom, while an example from social-political field is the Arabic revolutions started in late 2010. In all these cases, messages were filtered using the related location such as coordinates, user location settings, or place names in text or tags. The impact of (geo)social media during crisis events has been shown to have high value for relief workers or coordinators and the affected population.

Another group of applications uses the set of places to discover patterns. An example is the tourism knowledge scenario. In Web 2.0 communities, people share their traveling experience in blogs and forums. These articles, named travelogues, contain various tourism-related information, including text depiction of landmarks, photos of attractions, and so on. Travelogue provides an abundant data source to extract tourism-related knowledge. Travelogues can be exploited to generate location overviews in the form of both visual and textual descriptions. The method consists first in mining a set of location-representative keywords from travelogues, and then in retrieving web images using the learned keywords. The model learns the word-topic (local and global tourism topic, such as an attraction sight) distribution of travelogue documents and identifies representative keywords within a given location. Complementing travelogues, geo-referenced photos also tell a great deal about tourism knowledge. The photos, together with their time- and geo-references, implicitly document the photographer's spatio/temporal movement paths. The tourist-visited points can be grouped, mined to distinguish patterns, and used to rank places of interest and generate recommendations. In most of these cases, applications use location extracted from human trajectories in the real world, but they are not really using the trajectories itself.

A third group of applications also considers the users' interactions and relationships. In fact, geo-social networks provide not only the location, but also the explicit social links, and in some cases explicit declaration of kinships and partnerships, giving the possibility to overcome the shortcomings of techniques to infer tie strength. They also give high-resolution location data, as one

can distinguish between a check-in to a different floor of the same building. To give an example, Yahoo! research labs published a study on the attempt to extract aggregate knowledge on certain locations from large scale geo-referenced photos at Flickr. The knowledge here refers to the word or concept that can best describe and represent a geographical region. The challenge is to extract structured knowledge from the unstructured set of tags. The premise of the proposed solution is based on the human attention and behavior embedded in the photos and tags. Namely, if tags concentrate in a geographical area but do not occur often outside that area, then these tags are more representative to the area than those spread over large spatial region. This example shows also that there is a need to model human behavior and this aspect constitutes an interesting research topic by itself. Of course, models and hypotheses are geographically dependent as Western people often act differently from Eastern people in a social context. However, online social networks' check-ins are usually more sporadic than phone calls, providing less temporal resolution than mobile data.

Some references for further information are provided in Section 16.7.

Theoretical Application Scenarios

In this section we describe some possible scenarios where the analysis of virtual movements in geo-social networks can be useful, but has not yet been investigated by researchers. For instance, in the emerging field of human dynamics, a central point is the understanding of the interplay between human mobility and social networks. How do the mobility patterns and parameters depend on social network characteristics? The study of such interaction requires massive society-wide data sets that simultaneously capture dynamical information on individual movements and social relationships. Traditionally, this problem is addressed by using mobile phone networks, because they provide at the same time temporal information and social contacts. However, there are at least two problems with this kind of mobile phone data. First, friendships are not explicit but are inferred by creating a who-called-whom graph, with the possibility of inaccurate information about tie strengths. For example, a person does not often call people who live with him or her. The low number of calls between them is interpreted as a weak tie, leading to a bad representation of reality. This aspect is overcome by social networks where strong ties generally generate more direct messages/interactions. Secondly, we know users' positions only when they perform a call, and merely know the position of the tower managing the area the user is within, and not the actual geographical location of the user. In the geo-social network application the user's location can be retrieved when he or she publishes content and can also be derived from the user's, friends' contents if he or she is moving with them. In the last case some level of uncertainty is introduced (see Section 16.4.4).

The spreading of biological and mobile phone viruses is another context in which geo-social data could be useful, because epidemics are also determined by the structure of social and contact networks within the population, and human mobility patterns. The mathematical modeling of infectious diseases must take into account travel patterns within a city or the entire world, and accurately shape the underlying contact network depending on the nature and the infectiousness of the pathogen. For example, with highly contagious diseases (e.g., transmission based on coughs and sneezes) the contact network will include any pair of people who sat together in the same place. For a disease requiring close contact (e.g., sexually transmitted disease), the contact network will be much sparser. Similar distinctions arise in the computer virus context, where malware infecting computers across the Internet will have a much broader contact network than one that spreads by short-range wireless communication between nearby mobile devices. Depending on the case, a contact network based on co-location in a place or the explicit social network could be inferred by using geo-social data from geo-social networks, geo-tagged photo websites, or geo-referenced microblogging websites. Some research in this direction has been conducted, but it is far away from being exhaustive.

Mobility patterns of a population can be extracted by using check-in trajectories of users, in order to define the epidemic model or to perform a simulation scenario. A very fascinating application is the development of mobility models and routing algorithms for the so-called opportunistic networks. They are a new paradigm of computation in which there is no fixed infrastructure, and mobility is exploited as an opportunity to deliver data among disconnected parts of a network. When a node has data to transfer to another node, and no network path exists between the sender and the receiver, any possible encountered mobile device represents an opportunity to forward and carry them until encountering another node deemed more suitable to bring the message to the final destination. Both in the design of routing algorithms and in the evaluation of them, a promising approach is that of incorporating the spatial dimension into a model based on time-varying social graphs. Geo-social data are clearly the most appropriate and useful tool in this context because they provide at the same time all three dimensions of human movements: spatial, temporal, and social dimensions. In addition to this, explicit social relationships from online social networks can be incorporated to better design protocols that are able to learn the social network of users, for example in order to exploit the role of hubs (users with the highest number of contacts) in the dissemination process, or to predict new friendships and contact opportunities.

These examples are, of course, not exhaustive. They just give a hint of the possible uses of CGI data in several different context scenarios. The technology and the application are moving and changing so fast that some very unexpected and innovative applications can be developed even in the next few months.

16.3 Trajectory from Geo-Social Web

Users in the social web leave footprints of their movements: they visit real and virtual places and their movements can be recorded and analyzed. Following the previous scenarios, we want now answer the following question: What kind of trajectories can we reconstruct from the geo-social web data?

Data we can commonly retrieve and access from geo-social networks is punctual and discontinuous. The only exception so far is the GeoLife project, an experiment carried out by Microsoft research in which 165 users tracked their GPS trajectories on a social platform. The main reason of discontinuity is not only related to the localization systems (GPS use low mobile battery duration), but also to the users' communication behavior on social networks. Generally a user posts a content when it is important for him or her to share it with others users or friends. This means that he or she is not interested in communicating in a continuous way. Moreover, some media are more used in specific circumstances (i.e., photo sharing/repository during holidays), while others in daily routine (i.e., check-ins or status updates). Following a single user on a single social network generates a finite list of spatio-temporal positions, which can be used to implement a discrete trajectory (see Chapter 1). This use of discrete position is in a very early stage and still has several limits. One example is the increasing popular service called Google Latitude, which allows users to share their location with friends and add it to their status message in other Google applications. The *history option* (in a beta release at the moment of writing) stores the user's past locations. The user can access a restricted area where he or she can visualize the trajectory on Google Maps/Earth and a dashboard showing information such as trips, frequently visited locations, distance traveled, and time spent in different places. This application uses raw data to reconstruct the user trajectories and enrich them with semantic information, as described in Chapter 1 for semantic trajectories and behaviors. In Figure 16.1 it is possible to see one of the authors' Google Latitude trajectory from one month of data. As it is possible to see, the trajectory reconstruction in the social network has some challenging issues such as, for example, the data acquisition, given that the data can be discontinuous in time. This is manifest in the figure where long straight lines connect far points on the map. There is no attempt to connect to road map layer or transportation means.

Following a single user in his or her daily social networks activity on different social platforms could help in creating different trajectories or in filling some gaps with respect to using only one social network source. A very nice example of a *segmented trajectory* (see Chapter 1) is shown in Figure 16.2, extracted from an advertising of a train WiFi connection. The option to share information between different social networks, publishing content from one platform to another platform, is a very recent trend and it has not yet been studied by the scientific community.



Figure 16.1 One-month trajectory from one author's Google Latitude logs in the northeast part of Italy.

16.4 Geographic Information in Geo-Social Web

In the following sections we focus on the geographic aspects related to information it is possible to retrieve from the social web. We then answer the following questions:

- How does location information relate to generated information on the web? (Section 16.4.1)
- Which are the characteristics of these data? (Section 16.4.2)



Figure 16.2 A social network use segmented trajectory.

- How can trajectories' footprints in the web be retrieved? (Section 16.4.3)
- What are the possible sources for uncertainty (with respect to the location information)? (Section 16.4.4)

16.4.1 Location: From Real World to Geo-Social World

We refer to “content” as any piece of information (such as text, image, audio, or video, in any possible format) that it is possible to publish on the web as a resource. Content is generated by a person (that represents him/herself or a broader entity, such as an enterprise or an agency) using a device. The content describes a real/abstract object/event. Based on the Oxford English Dictionary definitions of “real” and “event,” we refer to a real object/event as “actually existing as a thing or occurring in fact; not imagined or supposed.” A real object is a perdurant entity in the world such as a mountain or a building and a real event is “a thing that happens or takes place, especially one of importance” in a specific place in a limited amount of time, such as a forest fire or a football match. We refer to abstract object/event for every other type of information, including mood and feeling description, such as messages like “I really feel good, today.” Even if abstract object/event can have associated geographic coordinates, we limit the discussion to the real objects/events and we call them *features of interest*. In Figure 16.3 we can see three levels: the real world, the content, and the social web levels, and the relations among objects in the different levels. The entities in the bottom part (the real world) are the person, the device, and the feature of interest. Each of them has a spatial location and an extension.

Any information produced is called content. A piece of information associated to a content describing some properties of the information is generally referred to as metadata. A geographic content, or CGI, has associated *geospatial information* that represents a spatial reference and geometry in any format. In other words, the metadata also contain the geographic reference. The metadata can be automatically generated by the device (such as the date for a digital photo) or manually added by a person (such as the title or the tags). A GPS device can record the coordinate of the device and associate the geographic information to the content metadata. Content can also include *implicit geographic information* such as place name in a textual message or the object represented in a photo. The implicit information can be made explicit using different applications and strategies and added to the content as metadata. The content with its metadata is published on the web and becomes *published content*: a shared resource for a certain community. At the web level it is also worth noticing that a person has a virtual identity. His or her personal information on the social web can include geographic information related to his or her usual living place, visited places, and/or actual location. We call them *geographic user information*. These data, especially the actual position, can be manually set or automatically updated from

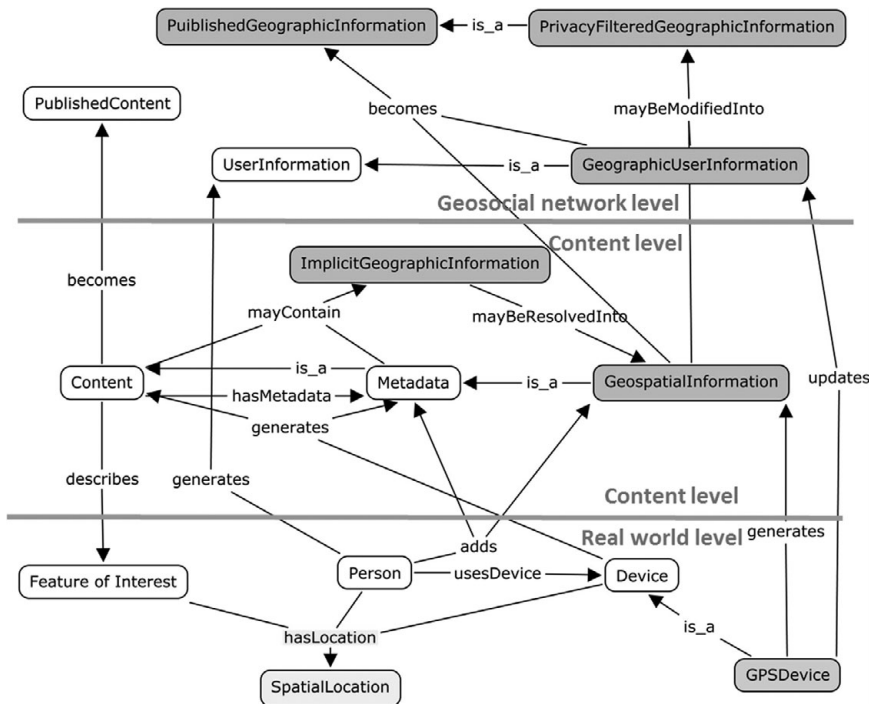


Figure 16.3 Conceptual model of CGI. The shaded concepts represent where the spatial information could be retrieved. The *SpatialLocation* concept represents the physical extension of the entities in the real world (person, device, and feature of interest).

a GPS device. The geographic information contained in the different levels may be synchronized and coincident. To summarize and answer the first question in the introduction, spatial location associated with a feature of interest or a person in the real world may have a representation in the geo-social network, very often associated to content as metadata.

16.4.2 Comparison among GPS, GSM, and Online Geo-Social Network Data

In the last few years, many researchers studied geo-tagged data such as GPS traces, GSM data, and data coming from geo-social networks. These data coming from GSM and GPS sources look different, and may seem not even comparable with the geo-tagged data present in online social networks. However, we believe that it is useful to the reader to highlight and compare some aspects. Therefore, the purpose of this section is to shed some light on differences and similarities, in both the data sources and the final tasks of analysis that can be performed on them. Let us first review the different kinds of data that we refer to. Our first

source, broadly discussed in the book, is the GPS (Global Positioning System) data. We have seen that there is a large variety of devices dealing with this kind of data: mobile GPS navigation systems, GPS loggers, GPS anti-theft systems, GPS units for photo cameras, and so on. Clearly, the final use may differ, but these data sets have some features in common: they all take the global coordinates (latitude, longitude, and time) of the device and store it for a specific purpose. Most of them (loggers and navigators, for example) take the information at regular intervals of time spanning from one second to a few hours, depending on the final application (e.g., turn-by-turn navigation systems continuously collect and process the GPS signal), and store it for the final purpose. As we have seen also in Chapters 1, 2, and 3, a single line extracted from these data typically includes at least the following information:

```
ID, timestamp, latitude, longitude, quality of signal
```

where `ID` is the device id, `timestamp` is the current time, usually expressed in seconds since the 1970 (higher resolutions may be needed depending on the application), `latitude` and `longitude` are the GPS spatial coordinates, and `quality of signal` may give information on the accuracy of the measurement. Depending on the application, a user may personally produce small to large amount of data of this kind. Also, different sources of this kind of data are available, most of which are not public: anti-theft systems, GPS loggers, and navigators, for example, are meant to be for personal use, and generally these data are not publicly available unless the owners explicitly share them on some social media. Another different source of data is the GSM CDR (Call Details Record): when placing mobile phone calls, the users generate a large amount of data about their calls, such as the number called, time, and duration. As seen in Chapter 2, a single record of these data has usually the following format:

```
callerID, receiverID, time, antennaID, start, stop, callID
```

where `antennaID` is the ID of the GSM base station the phone is attached to in that moment, and `callID` is used to track the call through different antennas in the case of a user moving in space. As we see, the spatial information of these data is much less precise: while the GPS can be accurate to the centimeter, in GSM data we can only use the antenna ID as geographic information, and this is very rough. In fact, a single antenna can cover a round region of very diverse radii, depending on the power of the antenna, the placement of it (within the city or countryside), and other factors. Therefore, the position of the caller is estimated with a precision that usually is on the order of hundreds of meters. Moreover, this is clearly privacy-sensitive information, and these kinds of data are usually not publicly available. Lastly, there are the data coming from the online geo-social networks and services. These data are of a different form with respect to the previous two types as, in addition to the potential geographic

information contained in it, they also typically include the content. We can, in fact, detect two blocks of information in this type of data: the geo-location and the media payload, the latter being either a text message, or a picture, a video, and so on. The content contained in the second block is then said to be “geo-tagged” according to the first block of information. The geographic information contained in this kind of data is usually a derivative of the GPS data coming from mobile devices (e.g., smartphones, PDAs, cameras) from which the user generated the content. Thus we can still consider this as a GPS data source, even if, given the very particular features of the application (i.e., no need to continuously track the user, no need for a specific precision, and so on), these data are differentiable from the one coming from the GPS navigators and loggers, in terms of precision, temporal resolution, and final volume of data collected. Given the large amount of geo-social networks and services available nowadays, it would be impossible to list all the possible information available online. We can, however, present here three different types of services, representative of a large set of available online social networks: Twitter, Flickr, and Foursquare. Twitter is a social network where users can post short messages in their timeline (typically publicly available) that will appear automatically within the timelines of all their followers. A typical message is a text message no longer than 140 characters, which may contain text and URLs for attached media such as pictures or videos. The messages can also be geo-tagged if the user has enabled this feature. A typical piece of data then contains the following information:

```
userID, messageID, text, geo-location, timestamp
```

Flickr is a photo sharing service with a social network layer where users can post pictures and video in their profile. Tags, comments, geo-location, and EXIF data (technical data about the picture) are usually associated with the pictures (or videos). A typical piece of data regarding a picture contains the following:

```
pictureID, userID, geo-location, timestamp, tags, comments
```

Foursquare is a location-based social network where users can post their current location and share this with all their friends. The service includes game features to incentivize the users to share their location. A typical piece of data contains the following:

```
userID, geo-location, locationID, timestamp
```

Table 16.1 summarizes some properties of the data we have described. Note that they are typical properties and individual examples for real-world scenarios, therefore may differ depending on the application. As we see, the three sources of data differ in public availability, volume of data usually generated per user,

Table 16.1 *Summary of Typical Properties of Mobility Data from GSM, GPS, and Geo-Social Network Sources (Real-World Scenarios may Differ Depending on the Application).*

Source	Public	Volume Per User	Accuracy	Privacy Sensitive	Social Layer
GPS	No	High	1cm	Yes	No
GSM	No	Low to high	100m	Yes	Yes
Geo-social nets	Yes	Low to moderate	1cm to 1m	No	Yes

accuracy of the data, whether the data should be considered as sensitive for privacy reasons (in online social networks usually the data are sparse and provided intentionally by the users, bringing this type of data to a reasonably nonsensible status), and the social dimension (i.e., there exists a social connection between two users). Clearly, given the above characteristics, the tasks of analysis to be performed on the different data are very different, and each task should be conducted on the most appropriate data. For example, assessing the validity of an urban transportation system using online social network data may provide inappropriate results, as the data do not contain enough and precise information.

16.4.3 CGI Retrieval from Geo-Social Networks

While GPS and GSM data are typically collected by private entities, telecom providers or citizens storing their trajectories on their personal devices, geo-social data are characterized by being publicly available on social media platforms. All major geo-social networking systems offer access to their huge corpus of data via several Web APIs (“application programming interface”). Many developers have created and made freely available libraries that do a lot of the heavy lifting needed to interact with the APIs, allowing researchers and data analysts to reconstruct and explore portions of social graphs and users’ movements. An API provides methods to access almost every feature of the system, and is typically defined as a set of HTTP request messages along with a definition of the structure of response messages (usually in an XML or JSON format). Each API is in constant evolution and represents a facet of the system, allowing developers to integrate specific functions or to build upon and extend their applications in new and creative ways. However, as regards the downloading of data, it presents some limitations due to compliance with privacy policies or the management of server load. Such restrictions define the level of detail and accuracy with which is possible to get data. We present a quick discussion of the API challenges of three very popular geo-social networks.

Twitter, for example, currently provides three APIs. Two of them offer methods to access status data and user information (name, profile, following/

followers, tweets), with a maximum rate limit of 350 requests per hour. The third one, the streaming API, is the most suited access for data mining or analytic research, allowing one to retrieve a 1% filter of all tweets that users are actually carrying out, possibly using some filtering fields such as keywords, tags, users, and geographic bounding box. Such a rate limit can be raised by asking Twitter for a “gardenhose” access, in order to receive a steady stream of tweets, very roughly 10% of all public statuses. Note that these proportions are subject to unannounced adjustment as traffic volume varies.

Unlike Twitter, the Foursquare API allows one to view all friends of an individual but does not allow one, for reasons of privacy, to “stalk” a specified user. The only way to collect information about users’ activity is to select a set of venues in one or more specific regions, and download all the activities (check-ins) performed in those locations, with a rate limit of 5,000 requests per hour. Both Twitter and Foursquare have severe limitations to data retrieval, enabling one to gather only a very partial subset of users’ activities.

Among many other geo-social networks, Flickr poses the fewest limitations. In such online photo management system, practically all the valuable meta-data such as tags and geolocation can be accessed by API programs. Anyway, some experiments carried out by the authors using the same query in different moments lead to retrieve slightly different results, leaving some uncertainty on the soundness of results. Applications can produce raw or derived data. Raw data can be the coordinates (latitude and longitude) of the message generated by the mobile device, and derived data can be the coordinates’ bounding box, the place type, the place name, or the street name. This information is produced by the social network application using the coordinates passed by the mobile device. The information that can be produced and retrieved changes depending on the system, the device, the privacy settings, and so on.

16.4.4 Geographic Uncertainty of CGI

As we already pointed out, geo-social data can have several sources of uncertainty. Uncertainty aspects should be taken into account when performing statistical analysis and when developing systems based on these kinds of data. In this section we discuss some of the most important uncertainty aspects of CGI data. Other aspects of uncertainty are covered in Chapter 5.

Uncertainty about Precision

In this category we join issues related to data generation. The first source of uncertainty is information granularity: each point of the trajectory can have a different scale, sometimes coordinates of a specific place, sometimes a bounding box area. The second one is related to devices. A third one is that the precision can be modified by the social network system: in some applications (such as

Foursquare), the GPS data are used to infer higher-level information, such as the address of a place, and the passed coordinates are hidden. Another source of uncertainty is the location system used by the device. Figure 16.1 shows an example of this error. On the left bottom corner of the trajectory there is a point located in Portoferraio, Elba island. The user never went to the island. She set off the GPS for energy saving and the data were retrieved using the GSM antenna. Her mobile phone was attached to the Portoferraio antenna but she was physically on the coast, some kilometers north. There is no way to extract this information from the generated data.

Uncertainty about Credibility

In some cases we can witness the presence of “spammer” users, who bombard the system with tweets and/or randomly change the GPS coordinates relative to their geographic positions to cheat the anti-spam system. In other cases people can voluntarily publish different locations just for fun or to protect their privacy.

Uncertainty Due to Privacy Settings

At the social network level, the geographic information can be filtered and modified for privacy reasons using a less detailed geographic level, although the device transmits coordinates.

Uncertainty Due to Multiple Data

Tweet example: “Two very large forest fires in the mountains behind Funchal clouds of smoke covered the sun turn sunlight deep yellow ash coming down.”⁴ It is possible to associate this Tweet to three locations spatially not coincident. Referring to the conceptual model described in Figure 16.3, the Feature of Interest described by the content is the forest fire that has itself a spatial location (the forest location). The content has also implicit geographic information with the toponym Funchal, or more precisely with the mountains behind the city. It is worth noticing that there is a certain level of uncertainty both in the definition of the forest on the mountain and in the toponym Funchal, which represent two different locations. The third location is the user/device location. Suppose the Tweet message itself has the coordinates originating from the source GPS device, we can suppose that the user was sending the message a safe distance away from the forest fire.

Uncertainty Due to User and Content Location

Photo example: Let us consider a person taking a picture with a camera or a smartphone with GPS-integrated system. The person and the device coordinate overlap, while the subject of the photo coordinate has a distance from the camera.

⁴ Twitter, posted by user “Kevin bulmer” on Fri., Aug. 13, 2010 h20:21.

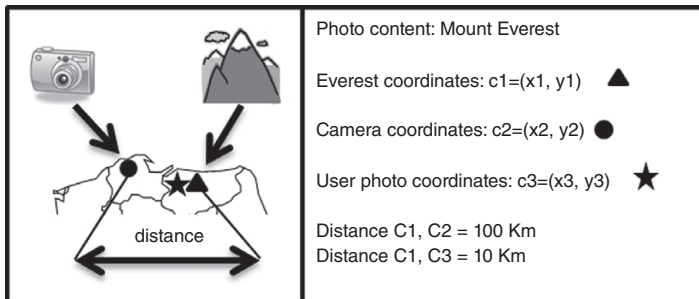


Figure 16.4 Comparison of device precision and semantic precision.

This distance could be considerable. Let us imagine that the content represented in the photo is Mount Everest. The user with the camera is necessarily far from the mountain peak to include it into the photo. As shown in Figure 16.4 on the right side, the mountain and the camera could be consistently far one from the other.

These last two examples illustrate that there is a discrepancy between the location of the content (the registered device location at the time the message is sent or the photo is taken) and the geographic content contained in the message itself. The location of the device is not necessarily equal to the location of the reported content: they can overlap or be far away, as in the examples. This inconsistency is not of a technological nature, but will always include semantic aspects.

16.5 Open Issues

Finally, we want to raise new key questions that we leave open for future research, which we believe will constitute interesting problems for the communities of computer scientists, sociologists, physicists, and economists in the years to come. Huge amounts of socially generated media resources on the Internet are a result of experience sharing by web communities. This fast-growing media collection records our culture, society, and environment, and provides opportunities to mine semantic and social knowledge of the world. Moreover, recent popularity of location-based social services, such as Foursquare, Gowalla, and Hot-Potato, has generated a huge amount of detailed location and event tags. It covers not only popular landmarks, but also obscure places, thus providing broad coverage of locations in unprecedented scales. This large amount of information, often unstructured, opens a first research issue in the field of real time analysis of data flow. The research broadly covers several aspects, such as very large data repository in nonstandard data structures, extracting semantic aggregation of tags,

detecting places and events from unstructured text, and finding automatic ways to link data from different sources. These topics are strictly linked to the development of the so-called Semantic Web and big companies such as Google and Yahoo! are constantly researching and developing in these fields. Investigation of place and event semantics of geo-referenced tags, in addition to the representativeness, is a prerequisite to using geo-social data. A place tag is defined as a one that exhibits significant spatial patterns, while an event tag refers to one that exhibits significant temporal patterns. Both definitions are vague and subject to some geographic region. For example, “carnival” may not be able to indicate any event, but will be very specific if only carnivals in New York City are considered. Analyzing the spatial and temporal distribution of tags and identifying the distributions of events and places with relative geographic scales can be useful to many applications, such as image search, collection browsing, tag visualization, and, of course, mobility analysis. Another open issue is the multilingualism of (geo-referenced) web media. Geo-referenced media is, in fact, multilingual in nature. However, most systems take English as the sole processing language. This effectively excludes the media resources in other languages. The consequence is that the knowledge and patterns mined from geo-referenced media are biased toward English-speaking countries and regions, though people are more comfortable using their local language (also dialects and slang) to communicate with friends, especially in colloquial sentences such as the ones used in chats, SMS, or status updates, or in stressful, demanding situations such as disasters or danger. The geographic locations of photos on the Internet have opened up a new host of research and application possibilities. As described in the photo example in Section 16.4.4, a spatial gap can exist between the GPS camera position and the position of the subject in the photo. Knowing the geographic orientation of photos, that is, in which direction the cameras are pointing, will be useful to fill the gap. Though most cameras are not equipped with sensors to measure the orientation and inclination of the device, smart photos, with the iPhone and HTC Magic as prime examples, have started to embrace digital compass technologies. In addition to hardware sensors, software solutions to estimate photo orientation also exist, for example estimating the relative translation and orientation between photos, by leveraging the visual redundancy among photos. Till now, geographic orientation of photos was rarely available. Nevertheless, with the development of compass-equipped cameras and smartphones, such kind of metadata is expected to emerge in the near future. With the availability of photo orientation metadata, many compelling applications can be accomplished. For example, with the photo alignment information, visual summarization and browsing of photo collections can be adaptive to the user direction and perspective on the map. Moreover, 3D reconstruction of geolocation can be much more efficient.

16.6 Conclusions

We have discussed mobility and geo-social networks, a very promising field of research nowadays, in which wide and multidisciplinary studies have been conducted in the last few years. We have seen how the interest in such topics is widely motivated by the close relationships that may reside between the social and mobility behavior of humans: people move, they move with friends or relatives, they share experiences, they propagate information about new places to friends, and so on. Moreover, in the last years, it is clear how this process, supported by the large amount of online (geo)social networks and services, is extensively conducted online, in near-real time, with a clear social and participative trend. These kinds of interactions and behaviors clearly produce massive amounts of data about human actions, related to both social and mobility aspects, and open the way for many interesting research challenges. However, despite the large interest and the large amount of data produced, we have seen how there is a clear disproportion between the results obtained so far and the vast quantity and diversity of issues that are still open. We believe that the issues and peculiarities related to the data (availability, privacy, granularity, and so on) and the rapid explosion of the availability of new services and trends are two clear reasons why it is still hard for the research in this direction to take off and to produce large and strong analytical results. The preliminary work conducted by many researchers so far is, however, very promising, and it seems clear that we are facing the start of a new era in the research on society and individual human behaviors.

16.7 Bibliographic Notes

In order to complete this chapter, we present some related work. We suggest to read them to deeper understand some ongoing research in the field of geo-social networks. Only the last three are related to trajectories; the others deal with the geographic aspect of geo-social data. The work of [Warf and Sui \(2010\)](#), in between geographic science and philosophy, mainly discusses how in practice neogeographers use geo-spatial technologies in multiple ways as opposite to conventional GIS. [Craglia et al. \(2012\)](#) describe how the use of CGI contributes to the vision of Digital Earth, extending the paradigm of spatial data infrastructures by advocating an interactive and dynamic framework based on near-to-real-time information from sensors and citizens. In [Chorley et al. \(2011\)](#), the authors, analyzing a data set composed by check-in data from Foursquare, reveal some individual characteristics of the cities. [Cho et al. \(2011\)](#) investigated the interaction of a person's social network structure and his or her mobility using data sets that capture human movements from Gowalla, Brightkite, and phone location trace data. They tried to understand if friendships influence where people travel,

or if it is more traveling that influences and creates social networks. The major contributions of [Gaito et al. \(2011\)](#) are the definition of the so-called geocommunity and the creation of a complex network-based methodology to extract geocommunities from GPS data applying clustering algorithm. [Kisilevich et al. \(2010a\)](#) propose an approach for analyzing trajectories of people, using geo-tagged photos collected from the photo-sharing site Flickr and a Wikipedia database of points of interest (POI). In his article, [Purves \(2011\)](#) discusses the utilization of user-generated content (UGC) as a data source for studying geographic questions, and proposes two examples: the derivation of vernacular regions and trajectory analysis. [Jankowski et al. \(2011\)](#), in order to discover itineraries and preferences of landmarks in an urban context, aggregated geo-tagged photos downloaded from Flickr. They were able to find precise events that attracted the attention of photographers. A spatial analysis of movement trajectories led to interesting findings related to photographers' itineraries. [Lucchese et al. \(2012\)](#) were able to extract, from photos published on Flickr, touristic points of interest in a city and provide automatically generated, personalized recommendations.

17

Conclusions

Chiara Renso, Stefano Spaccapietra, and Esteban Zimányi

Mobility data management and analysis have emerged in the last decade as a very active research domain, promoted by academic events (e.g., several dedicated conferences, journal issues, and seminars), international R&D projects (e.g., GeoPKDD,¹ MODAP,² MOVE³), and industrial initiatives (e.g., the multiple mobility contests that have been organized recently by several organizations). This book documented the richness and significance of the main research achievements in a variety of domains related to mobility data management and showed how several application domains have already benefited from these achievements. It also highlighted two very important areas for new applications related to most advanced technological environments, namely social networks and network sciences. Yet there is much room for further work in all aspects of movement analysis. The following concluding remarks aim at showing some further developments that are expected within the short-term future and that build on the mobility technologies discussed in this book.

Basic Trajectory Framework

A first evidence is that new research projects are needed to expand the scope and coverage of mobility studies, currently mainly restricted to the limited set of basic concepts described in the first chapter of this book. For example, analyzing movement of deforming areas (e.g., floods, pollution clouds, storms, diseases) has received relatively little attention up to now. Yet its economic importance is rapidly increasing as the disastrous effects of natural phenomena linked to current climate change are influencing government policies to promote better analyses of such phenomena. Several types of movement remain to be investigated,

¹ <http://www.geopkdd.eu/>

² <http://www.modap.org/>

³ <http://www.move-cost.info/>

including raster representations of movement, constrained movements, and relative movement.

In a complementary direction, it is important to push much further the study of collective behavior, that is, coordinated movement of persons, of animals, and of any moving object driven by humans (e.g., cars, planes, ships). While animals are obviously important to ecologists, humans' collective movements characterize a large number of applications, including the national security and intelligence domain that has become so critical in our current society.

Collective movement illustrates a more general research question: How to analyze relationships among trajectories. Current advances in this domain are mainly in terms of clustering, classification, and similarity analyses. Yet, other relationships could be defined and useful: inverse trajectories, useful for identifying return trips, and concatenation of trajectories, to make a global sense out of a sequence of trajectories, are just two examples of how the knowledge about trajectory understanding could be expanded. Also somehow related to collective movement is the study of interactions among trajectories. Indeed, a trajectory of a moving object may influence the trajectories of nearby moving objects. In a car traffic situation, for example, the behavior of a driver can influence nearby drivers. Open research questions include detecting such interactions and identifying the actors, their roles, and how influences propagate among moving objects.

Another direction for future research is investigating the new concept of user-centered mobility data, where all the footprints left by a moving user and collected via different means (e.g., GPS, social networks and mobile phones) are combined together to form a global vision on the user's movements. This raises clear interoperability and integration issues that have not been addressed in this book.

Trajectory Reconstruction

Data acquisition depends on the sensing technologies that are available and appropriate for the application at hand. Usual technological evolution will certainly introduce new features that will prompt innovation in trajectory reconstruction approaches. Future work in this domain may include the exploration of intelligent ways to automatically extract proper values of trajectory reconstruction parameters according to a number of characteristics of data sets, as well as the extension of this technique to be able to identify different movement types (pedestrian, bicycle, motorbike, car, truck, etc.) so as to enable application of customized reconstruction techniques, resulting in better identification of trajectories.

Existing techniques have to be reconsidered, taking a more global approach. For example, map matching can be significantly improved by taking into account semantic aspects (e.g., the purpose of stops). More sophisticated analyses can

be enabled via careful tuning (e.g., tuning stop identification and interpretation to make it efficient even for short stops), and via consistency enforcement for multiple, correlated annotations and segmentations.

Trajectory Storage

Commercial software is not yet ready to support trajectory data with the new management facilities that this new type of data requires. At this point trajectory management is appropriately supported only by research-driven prototypes that have already reached the level of operational systems. This book presented the *SECONDO* system, which is the most advanced system that has been purposely built to support mobile data management. Ongoing work in the *SECONDO* team aims at extending the model and the system in two major directions. On the one hand, discussions with ecologists have shown that it is crucial to analyze moving animals in the context of environmental data such as temperatures, elevation, and snow extent. These data are available as raster data. Hence it is necessary to handle raster data together with moving object data in a query. *SECONDO*'s high-level conceptual model needs to be extended with the data types providing continuous functions of space and the corresponding operations. A second direction is parallelization using the MapReduce approach in order to make trajectory database applications scalable. MapReduce will enable distributed execution of complex queries by controlling *Secondo* systems running on many computers in a network. A different trend is represented by efforts to complement commercial systems with an external layer providing the functionality required for mobility data management. The *Hermes* system is the best representative of this trend. At what pace the new functionality will be integrated into commercial systems depends on the DBMS industry.

Similar concerns apply to data warehousing systems, yet the situation regarding trajectory data warehouses is far less advanced than for trajectory DBMS and research still needs to clearly identify and characterize the extensions needed to upgrade the data-warehousing paradigm to make it suitable for trajectories, and eventually implement them efficiently.

Privacy Issues

The privacy solutions that have been discussed in the book are inherently limited in scope as they are drawn to target specific privacy goals under well-defined assumptions about the role of untrustworthy parties and their capabilities. A challenge for the near future is how to overcome the fragmentation of privacy technologies to achieve solid conceptual foundations. This question is of vital importance for future research on privacy. A theoretical framework centered on the concept of location privacy metric has been recently proposed to deal with the problem. By quantifying the amount of protection offered by a privacy enhancing technology, location privacy metrics pave the way to the definition

of a rigorous methodology for the comparison of privacy solutions. Recent approaches propose metrics based on the recent paradigm of differential privacy. However, the specification of generalized metrics, while fundamental to creating a corpus of rigorous concepts, cannot be seen as the panacea for privacy. Privacy is eminently a user-centric requirement. In this view we believe that any privacy solution should be eventually validated by users. Accordingly, the aspect of privacy usability is a prime issue.

Privacy usability has multiple dimensions. One of these dimensions is personalization, that is, letting the user specify the requested amount of privacy. Another one is privacy adaptability, that is, the amount of protection varies based on the context. It is also important to consider the privacy requirements emerging from novel applications. While most of existing research on location privacy focuses on privacy in location-based querying in LBS, novel applications are emerging calling for location privacy solutions. We mention in particular mobile sensing applications (i.e., acquiring geo-referenced data through sensors installed on mobile devices), location-sharing applications in geo-social network (e.g., place check-ins), and location services, that is, requesting location data to a third party.

Trajectory Analysis

As long as mere spatio-temporal trajectories are concerned, trajectory analysis significantly benefits from the existing knowledge in data mining, knowledge extraction, and visualization. Semantic enrichment of movement data leads to higher levels of analysis. Instead of just discovering movement patterns, research and applications are turning toward analyzing moving objects' behaviors. Their discovery now represents one of the most popular uses of mobility data and possibly the ultimate goal of trajectory analyses. Lifting up the analysis to the semantic level largely remains to be explored. This topic is specifically addressed, for example, in the SEEK project⁴ where methods to semantically enrich the trajectory knowledge discovery process are investigated. Understanding why and how people and animals move, which places they visit and for what purposes, their activities, and what resources they use is of tantamount importance for all kinds of decision makers, in particular public authorities in charge of managing societal resources. The relative novelty of the domain leaves many avenues for future work open. Many experiences have been made in a great variety of application domains and using a great variety of techniques, but much more work has to be done by the research community to build the scientific corpus that enables new applications to be developed easily, promptly, and on a sound base. The active involvement of domain experts is a necessary condition for such development.

⁴ <http://www.seek-project.eu>

We can confidently look forward to the continuous evolution and improvement of knowledge extraction techniques, moving from approaches mostly based on a single attribute (speed) and predefined points and areas of interest to approaches enabling multi-criteria analyses.

Similarly, we can expect an important development of the visual analytics domain. Its advantages in terms of immediacy and understandability of results make it very appealing for domain experts. These advantages include capability to rapidly switch among alternative ways to look at the same data; easiness of exploring the influence of given attributes on the behavior of trajectories; pattern detection via visual evidence; data aggregation into, for example, flows for higher-level analyses; and context-driven investigations. Therefore, visual analytics will certainly play a major role in the spread of analytical tasks in the application world.

While most of the ongoing works focus on mobility analysis, only few attempts have been made for developing explanatory and predictive models of mobility. For example, the DataSim project⁵ aims at developing next generation traffic simulation models. To improve the quality of simulation, it is necessary to model dependencies between mobility aggregates such as velocity (average or median) and counts of cars.

Person Monitoring

Person monitoring is rapidly evolving due to the widespread availability of positioning technologies. Unfortunately, gathering spatiotemporal information from large crowds remains a complex, usually labor-intensive and therefore expensive task. This book presented Bluetooth tracking as a possible solution for this niche. Bluetooth tracking is very easy to deploy, can be used for indoor/outdoor environments, and does not require any cooperation of the tracked individuals. Foreseen applications include analyzing the dynamics of visitors at a professional fair or in a shopping mall. Both domains provide excellent opportunities for marketing-oriented applications. Industry is very active in this domain (see, e.g., the recent launching of services such as Google Indoor), given the strong link between person monitoring and customized marketing.

Animal Monitoring

The last few years have witnessed a convergence between human and animal movement studies. Many concepts, data warehousing techniques, models, and analytical methods were developed independently and now we can stress that further research, leading to a growing methodological and theoretical unification, is necessary. A common and exciting challenge is represented by the analyses of networks, both social and ecological ones. Movement ecology studies have

⁵ www.datasim-fp7.eu/

demonstrated the potential of mechanistic or causal movement models that can find wide applications in many different research fields. Concepts like random walks and diffusion, for example, aim to connect individual and collective patterns.

Web

The social web has changed the way people create and use information. Particularly relevant to this book is the explosion of services based on geosocial content. So-called volunteered geographic information (VGI) allows people visiting real and virtual places to leave footprints of their movement in the social web, with their movements being recorded and analyzed. The amount of generated detailed location and event tags is huge and covers not only popular landmarks, but also obscure places, thus providing broad and wide coverage of locations in unprecedented scales. This large amount of information, often unstructured, opens a research avenue in the field of real-time analysis of data flow in the Semantic Web. Such interplay of mobility and geosocial networks research is very promising. There is indeed a strong relationship between the social and the mobility behaviors of humans. People move, they move with friends or relatives, and they use the social web to share experiences, propagating information about new places to friends.

The trajectory reconstruction in the social network opens such challenging issues as, for example, data acquisition, given that such data can be discontinuous in time, or geographically uncertain, because each point of the trajectory can have a different scale (sometimes coordinates of a specific place, sometimes a bounding box area).

Despite the large interest and the large amount of data produced, there is a clear disproportion between the results obtained so far and the vast quantity and diversity of issues that are still open. We believe that the issues and peculiarities related to the data (availability, privacy, granularity, and so on) and the rapid explosion of the availability of new services and trends are two clear reasons why it is still hard for the research in this direction to take off and to produce large and strong analytical results. The preliminary work conducted by many researchers so far is very promising, and it seems clear that we are facing the start of a new era in the research on societal and individual human behaviors.

Large Scale and Streams

Nowadays we live in a world overloaded by information. The information at our disposal is so large and complex that traditional data processing tools and paradigms are no longer capable to cope with it. This phenomenon has been dubbed “Big Data.” New computing paradigms have been proposed as a solution to this new state of affairs, MapReduce being the most prominent of all. Their aim is to enable massive parallelization of data processing in order to speed up this

process to cope with large-scale data sets. Trajectory data are not an exception to this phenomenon. The popularization of tracking technologies (e.g., GPS-enabled mobile devices) had as a consequence that huge amounts of trajectory data are being continuously collected. Therefore, all the analysis methods and software tools that have been presented in this book must be redesigned in order to scale up for much larger data sets, as most of the methods are quite restrictive with respect to data volume.

A related problem in this respect pertains to developing methods for streaming trajectory data. In many real-world applications (e.g., telecommunications, clickstream monitoring, sensor networks, traffic monitoring), data take the form of continuous data streams rather than finite stored data sets. These applications require long-running, continuous queries and analyses as opposed to single-time ones. Many aspects of data management and processing need to be reconsidered in this new setting and stream databases were developed as a possible solution for this. In a similar way to large-scale processing, new methods and tools have to be designed to enable stream-based processing of trajectory data.

Mobility Engineering

Definitely, more systematic exploration and experimentation is necessary to consolidate the theories and tools that this book has presented. Most of the experimentations that allowed researchers to assess their results have been carried out on an ad hoc basis and were limited to the data set available to the researchers. Systematic exploration of the applicability of an approach to mobility management remains to be done. Validating an approach for large-scale usage in the real application world needs repeated testing with varying parameters, varying techniques, and varying data sets. Among others, ground truth benchmarks need to be developed to create better possibilities to assess the value and portability of algorithms. Moreover, all involved tools and facilities will have to reach online availability to enable continuous analysis of and feedback for ongoing trajectories.

Turning research into engineering represents a huge challenge and calls for a strong cooperation among research, industry, users, and public authorities. The ultimate goal is to be able to develop general-purpose packages that will allow, for example, to translate GPS tracks into semantic trajectories. Similarly, general platforms should enable the tuning of the parameters for all tools that create, manage, and manipulate a trajectory data set. It is a long way to go, but as shown in this book we hopefully are on the right track.

BIBLIOGRAPHY

- Abul, O., Bonchi, F., and Giannotti, F. 2010. Hiding sequential and spatiotemporal patterns. *IEEE Transactions on Knowledge and Data Engineering*, **22**(12), 1709–1723.
- Agrawal, R., and Srikant, R. 2002. Mining sequential patterns. Pages 3–14 of: *Proceedings of the 11th International Conference on Data Engineering*. IEEE.
- Ahas, R., Aasa, A., Roose, A., Mark, Ü., and Silm, S. 2008. Evaluating passive mobile positioning data for tourism surveys: An Estonian case study. *Tourism Management*, **29**(3), 469–486.
- Alt, H., and Godau, M. 1995. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, **5**(1), 75–91.
- Almeida, V.T. de, and Güting, R.H. 2005. Indexing the trajectories of moving objects in networks. *GeoInformatica*, **9**(1), 33–60.
- Alt, W., and Hoffman, G. 1990. *Biological Motion*. Springer-Verlag.
- Alvares, L.O., Bogorny, V., Kuijpers, B., Macedo, J.A., Moelans, B., and Vaisman, A. 2007. A model for enriching trajectories with semantic geographical information. Pages 1–8 of: *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*. ACM Press.
- Andrienko, G., and Andrienko, N. 2010. A general framework for using aggregation in visual exploration of movement data. *The Cartographic Journal*, **47**(1), 22–40.
- Andrienko, G., Andrienko, N., Bak, P., Keim, D., Kisilevich, S., and Wrobel, S. 2011a. A conceptual framework and taxonomy of techniques for analyzing movement. *Journal of Visual Languages & Computing*, **22**(3), 213–232.
- Andrienko, G., Andrienko, N., and Heurich, M. 2011b. An event-based conceptual model for context-aware movement analysis. *International Journal of Geographical Information Science*, **25**(9), 1347–1370.
- Andrienko, G., Andrienko, N., Hurter, C., Rinzivillo, S., and Wrobel, S. 2011c. From movement tracks through events to places: Extracting and characterizing significant places from mobility data. Pages 161–170 of: *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. IEEE Computer Society Press.
- Andrienko, N., and Andrienko, G. 2011. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, **17**(2), 205–219.
- Andrienko, N., and Andrienko, G. 2013. Visual analytics of movement: An overview of methods, tools and procedures. *Information Visualization*, **12**(1), 3–24.

- Ankerst, M., Breunig, M.M., Kriegel, H.-P., and Sander, J. 1999. OPTICS: Ordering points to identify the clustering structure. Pages 49–60 of: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. ACM Press.
- Baglioni, M., Macedo, J.A., Renso, C., Trasarti, R., and Wachowicz, M. 2012. How you move reveals who you are: Understanding human behavior by analyzing trajectory data. *Knowledge and Information System Journal*.
- Barabási, A. L., and Albert, R. 1999. Emergence of Scaling in Random Networks. *Science*, **286**, 509.
- Barrett, G. 1996. The transport dimension. In: Jenks, M., Burton, E., and Williams, K. (ed.), *The Compact City: A Sustainable Urban Form?* E & FN Spon.
- Bertrand, F., Bouju, A., Claramunt, C., T., Devoegele, and Ray, C. 2007. Web architecture for monitoring and visualizing mobile objects in maritime contexts. Pages 94–105 of: *Proceedings of the 7th International Symposium on Web and Wireless Geographical Information Systems*. LNCS 4857. Springer-Verlag.
- Bole, A., Dineley, W., and Wall, A. 2012. *Radar and ARPA Manual: Radar and Target Tracking for Professional Mariners, Yachtsmen and Users of Marine Radar*. Third ed. Butterworth-Heinemann.
- Bonchi, F., Lakshmanan, L.V.S., and Wang, W.H. 2011. Trajectory anonymity in publishing personal mobility data. *SIGKDD Explorations*, **13**(1), 30–42.
- Bouvier, D.J., and Oates, B. 2008. Evacuation traces mini challenge award: Innovative trace visualization. Staining for information discovery. Pages 219–220 of: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*. IEEE Computer Society Press.
- Brakatsoulas, S., Pfoser, D., Salas, R., and Wenk, C. 2005. On map-matching vehicle tracking data. Pages 853–864 of: *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment.
- Bremm, S., Andrienko, G., Andrienko, N., Schreck, T., and Landesberger, T. 2011. Interactive analysis of object group changes over time. Pages 41–44 of: *Proceedings of the International Workshop on Visual Analytics*. Eurographics.
- Brockmann, D., Hufnagel, L., and Geisel, T. 2006. The scaling laws of human travel. *Nature*, **439**, 462–465.
- Bruzzese, D., and Davino, C. 2003. Visual post-analysis of association rules. *Journal of Visual Languages & Computing*, **14**(6), 621–635.
- Burke, R.R. 2005. The third wave of marketing intelligence. Pages 103–115 of: Krafft, M., and Mantrala, M.K. (eds), *Retailing in the 21st Century*. Springer-Verlag.
- Cabibbo, L., and Torlone, R. 1997. Querying multidimensional databases. Pages 319–335 of: *Proceedings of the 6th International Workshop on Database Programming Languages*. LNCS 1396. Springer-Verlag.
- Cagnacci, F., Boitani, L., Powell, R.A., and Boyce, M.S. (eds). 2010. Challenges and opportunities of using GPS location data in animal ecology. *Philosophical Transactions of the Royal Society*, B. 365 (159 pp.)
- Cao, H., Mamoulis, N., and Cheung, D.W. 2005. Mining frequent spatio-temporal sequential patterns. Pages 82–89 of: *Proceedings of the 5th International Conference on Data Mining*. IEEE Computer Society Press.
- Card, S.K., Mackinlay, J.D., and Shneiderman, B. (eds). 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.
- Cho, E., Myers, S.A., and Leskovec, J. 2011. Friendship and mobility: User movement in location-based social networks. Pages 1082–1090 of: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.
- Chorley, M.J., Colombo, G.B., Williams, M.J., Allen, S.M., and Whitaker, R.M. 2011. Checking out checking in: Observation on Foursquare usage patterns. Pages 28–39 of: *Proceedings of the International Workshop on Finding Patterns of Human Behaviors in Networks and Mobility Data*.

- Chow, C., Mokbel, M.F., and Aref, W.G. 2009. Casper*: Query processing for location services without compromising privacy. *ACM Transactions on Database Systems*, **34**(4), 24.
- Coscia, M., Giannotti, F., and Pedreschi, D. 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, **4**(5), 512–546.
- Craglia, M., Ostermann, F., and Spinsanti, L. 2012. Digital Earth from vision to practice: making sense of citizen-generated content. *International Journal of Digital Earth*, **5**(5), 398–416.
- Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. 2010. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*.
- Cranshaw, J., Toch, E., Hong, J., Kittur, A., and Sadeh, N. 2010. Bridging the gap between physical location and online social networks. Pages 119–128 of: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*. ACM Press.
- Damiani, M.L., Bertino, E., and Silvestri, C. 2010. The PROBE framework for the personalized cloaking of private locations. *Transactions on Data Privacy*, **(3)2**, 123–148.
- Damiani, M.L., Silvestri, C., and Bertino, E. 2011. Fine-grained cloaking of sensitive positions in location-sharing applications. *IEEE Pervasive Computing*, **10**(4), 64–72.
- Dee, H.M., and Velastin, S.A. 2007. How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications*, **19**(5–6), 329–343.
- Devogele, T. 2002. A new merging process for data integration based on the discrete Fréchet distance. Pages 167–181 of: *Proceedings of the 10th International Symposium on Spatial Data Handling*. Springer-Verlag.
- Dodge, S., Weibel, R., and Lautenschütz, A.-K. 2008. Taking a systematic look at movement: Developing a taxonomy of movement patterns. In: *Proceedings of the AGILE Workshop on GeoVisualization of Dynamics, Movement and Change*.
- Domingo-Ferrer, J., and Trujillo-Rasua, R. 2012. Microaggregation- and permutation-based anonymization of movement data. *Information Sciences*, **208**, 55–80.
- Douglas, D. and Peucker, T. 1973. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *The Canadian Cartographer*, **10**(2): 112–122.
- Dricot, J.-M., Bontempi, G., and Doncker, P. 2009. Static and dynamic localization techniques for wireless sensor networks. Pages 249–281 of: Ferrari, G. (ed), *Sensor Networks: Where Theory Meets Practice*. Springer-Verlag.
- Düntgen, C., Behr, T., and Güting, R.H. 2009. BerlinMOD: A benchmark for moving object databases. *VLDB Journal*, **18**(6), 1335–1368.
- Eagle, N., and Pentland, A. 2005. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, **10**(4), 255–268.
- Egenhofer, M. 2003. Approximation of geospatial lifelines. In: *Proceedings of the Workshop on Spatial Data and Geographic Information Systems*.
- Erdős, P., and Rényi, A. 1959. On random graphs. *Publicationes Mathematicae (Debrecen)*, **6**, 290–297.
- Etienne, L., Devogele, T., and Bouju, A. 2012. Spatio-temporal trajectory analysis of mobile objects following the same itinerary. Pages 47–58 of: Shi, W., Goodchild, M., Lees, B., and Leung, Y. (eds), *Advances in Geo-Spatial Information Science*. CRC Press.
- Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P. 1996. From data mining to knowledge discovery: An overview. Pages 1–34 of: *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence.
- Focardi, S., Montanaro, P., and Pecchioli, E. 2009. Adaptive Lévy walks in foraging fallow deer. *PLoS ONE*. doi 10.1371/journal.pone.0006587.
- Fraenkel, G.S., and Gunn, D.L. 1961. *The Orientation of Animals: Kineses, Taxes, and Compass Reactions*. Dover Publications.
- Fryxell, J.M., Hazell, M., Börger, L., Dalziel, B.D., Haydon, D.T., Morales, J.M., McIntosh, T., and Rosatte, R.C. 2008. Multiple movement modes by large herbivores at multiple

- spatio-temporal scales. *Proceedings of the National Academy of Sciences*, **105**, 19114–19119.
- Gaffney, S., and Smyth, P. 1999. Trajectory clustering with mixture of regression models. Pages 63–72 of: *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*. ACM Press.
- Gaito, S., Rossi, G.P., and Zignani, M. 2011. From mobility data to social attitudes: A complex network approach. Pages 52–65 of: *Proceedings of the International Workshop on Finding Patterns of Human Behaviors in Networks and Mobility Data*.
- Ghinita, G., Damiani, M.L., Silvestri, C., and Bertino, E. 2009. Preventing velocity-based linkage attacks in location-aware applications. Pages 246–255 of: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM Press.
- Giannotti, F., and Pedreschi, D. (eds). 2008. *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer-Verlag.
- Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. 2007. Trajectory pattern mining. Pages 330–339 of: *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*. ACM Press.
- Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., Renso, C., Rinzivillo, S., and Trasarti, R. 2011. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *VLDB Journal*, **20**(5), 695–719.
- Gómez, L.I., Gómez, S., and Vaisman, A. 2012. A generic data model and query language for spatiotemporal OLAP cube analysis. Pages 300–311 of: *Proceedings of the 15th International Conference on Extending Database Technology*. ACM Press.
- González, M.C., Hidalgo, C.A., and Barabási, A.L. 2008. Understanding human mobility patterns. *Nature*, **454**, 779–782.
- Gould, J.L., and Gould, C.G. 2012. *Nature's Compass: The Mystery of Animal Navigation*. Princeton University Press.
- Granovetter, M.S. 1973. The strength of weak ties. *American Journal of Sociology*, **78**, 1360–1380.
- Greenfeld, Joshua S. 2002. Matching GPS observations to locations on a digital map. *Proceedings of the 81st Annual Meeting of the Transportation Research Board*, **1**(3), 164–173.
- Gruteser, M., and Grunwald, D. 2003. Anonymous usage of location-based services through spatial and temporal cloaking. Pages 31–42 of: *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*. ACM Press.
- Gudmundsson, J., van Kreveld, M.J., and Speckmann, B. 2004. Efficient detection of motion patterns in spatio-temporal data sets. Pages 250–257 of: *Proceedings of the 12th International Workshop on Geographic Information Systems*. ACM Press.
- Guo, D. 2007. Visual analytics of spatial interaction patterns for pandemic decision support. *International Journal of Geographical Information Science*, **21**(8), 859–877.
- Guo, H., Wang, Z., Yu, B., Zhao, H., and Yuan, X. 2011. TripVista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. Pages 163–170 of: *Proceedings of the IEEE Pacific Visualization Symposium*. IEEE Computer Society Press.
- Güting, R.H., and Schneider, M. 2005. *Moving Objects Databases*. Morgan Kaufmann.
- Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., and Vazirgiannis, M. 2000. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, **25**(1), 1–42.
- Güting, R.H., Almeida, V.T. de, and Ding, Z. 2006. Modeling and querying moving objects in networks. *VLDB Journal*, **15**(2), 165–190.
- Güting, R.H., Behr, T., and Xu, J. 2010. Efficient k -nearest neighbor search on moving object trajectories. *VLDB Journal*, **19**(5), 687–714.
- Hägerstrand, T.H. 1970. What about people in regional science? *Papers of the Regional Science Association*, **24**, 7–21.

- Haghani, A., Hamed, M., Sadabadi, K.F., Young, S., and Tarnoff, P. 2010. Data collection of freeway travel time ground truth with Bluetooth sensors. *Transportation Research Record: Journal of the Transportation Research Board*, **2160**, 60–68.
- Hurter, C., Tissoires, B., and Conversy, S. 2009. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, **15**(6), 1017–1024.
- IALA, International Association of Marine Aids to Navigation & Lighthouse Authorities. 2004. *The Automatic Identification System (AIS), Volume 1, Part I, Operational Issues, Edition 1.3*. IALA Guideline No. 1028.
- IHO, International Hydrographic Organization. 2000. *Transfer Standard for Digital Hydrographic Data, Edition 3.1*. Special Publication No. 57.
- IMO, International Maritime Organization. 2008. *Development of an E-Navigation Strategy*. Reports of Sub-Committee on Safety of Navigation, 54th session.
- Inan, A., and Saygin, Y. 2006. Privacy preserving spatio-temporal clustering on horizontally partitioned data. Pages 459–468 of: *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery*. LNCS 4081. Springer-Verlag.
- Jankowski, P., Andrienko, N., Andrienko, G., and Kisilevich, S. 2011. Discovering landmark preferences and movement patterns from photo postings. *Transactions in GIS*, **14**(6), 833–852.
- Jensen, C.S., Lu, H., and Yiu, M.L. 2009. Location privacy techniques in client-server architectures. Pages 31–58 of: Bettini, C., Jajodia, S., Samarati, P., and Wang, S.X. (eds), *Privacy in Location-Based Applications: Research Issues and Emerging Trends*. Springer-Verlag.
- Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., and Shen, H.T. 2008. Discovery of convoys in trajectory databases. *Proceedings of the VLDB*, **1**(1), 1068–1080.
- Johnson, D. 1980. The comparison of usage and availability measurements for evaluating resource preference. *Ecology*, **61**, 65–71.
- Kalnis, P., Mamoulis, N., and Bakiras, S. 2005. On discovering moving clusters in spatio-temporal data. Pages 364–381 of: *Proceedings of the 9th International Symposium on Spatial and Temporal Databases*. LNCS 3633. Springer-Verlag.
- Kapler, T., and Wright, W. 2005. GeoTime information visualization. *Information Visualization*, **4**(2), 136–146.
- Karamshuk, D., Boldrini, C., Conti, M., and Passarella, A. 2011. Human mobility models for opportunistic networks. *IEEE Communication Magazine*, **49**(12), 157–165.
- Keim, D., Andrienko, G., Fekete, J.-D., Carsten, G., Kohlhammer, J., and Melançon, G. 2008. Visual analytics: Definition, process, and challenges. Pages 154–175 of: Kerren, A., Stasko, J., Fekete, J.-D., and North, C. (eds), *Information Visualization*. LNCS 4950. Springer-Verlag.
- Kellaris, G., Pelekis, N., and Theodoridis, Y. 2009. Trajectory compression under network constraints. Pages 392–398 of: *Proceedings of the 11th International Symposium on Advances in Spatial and Temporal Databases*. LNCS 5644. Springer-Verlag.
- Kimball, R. 1996. *The Data Warehouse Toolkit*. J. Wiley and Sons.
- Kisilevich, S., Keim, D.A., and Rokach, L. 2010a. A novel approach to mining travel sequences using collections of geotagged photos. Pages 163–182 of: *Proceedings of the 13th AGILE International Conference on Geographic Information Science*. Lecture Notes in Geoinformation and Cartography. Springer-Verlag.
- Kisilevich, S., Mansmann, F., Nanni, M., and Rinzivillo, S. 2010b. Spatio-temporal clustering. Pages 855–874 of: Maimon, O., and Rokach, L. (eds), *Data Mining and Knowledge Discovery Handbook*, 2nd ed. Springer-Verlag.
- Klug, A. 1982. Equivalence of relational algebra and relational calculus query languages having aggregate functions. *Journal of the ACM*, **29**(3), 699–717.
- Koubarakis, M., Sellis, T., Frank, A., Gutting, R., Jensen, C.S., Lorentzos, A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H.-J., Scholl, M., Theodoulidis, B., and Tryfona, N. 2003. *Spatio-Temporal Databases: The Chorochronos Approach*. Springer-Verlag.

- Kuijpers, B., and Othman, W. 2009. Modeling uncertainty of moving objects on road networks via space-time prisms. *International Journal of Geographical Information Science*, **23**(9), 1095–1117.
- Kuijpers, B., Moelans, B., Othman, W., and Vaisman, A.A. 2009. Analyzing trajectories using uncertainty and background information. Pages 135–152 of: *Proceedings of the 11th International Symposium on Spatial and Temporal Databases*. LNCS 5644. Springer-Verlag.
- Laube, P., van Kreveld, M., and Imfeld, S. 2005. Finding REMO: Detecting relative motion patterns in geospatial lifelines. Pages 201–214 of: *Proceedings of the 11th International Symposium on Spatial Data Handling*. Springer-Verlag.
- Lee, J.-G., Han, J., Li, X., and Gonzalez, H. 2008a. TraClass: Trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment*, **1**(1), 1081–1094.
- Lee, J.-G., Han, J., and Li, X. 2008b. Trajectory outlier detection: A partition-and-detect framework. Pages 140–149 of: *Proceedings of the 24th International Conference on Data Engineering*. IEEE Computer Society Press.
- Leitinger, S., Gröchenig, S., Pavelka, S., and Wimmer, M. 2010. Erfassung von personenströmen mit der Bluetooth-tracking technologie. Pages 220–225 of: *Angewandte Geoinformatik 2010*.
- Li, N., Li, T., and Venkatasubramanian, S. 2007. t-Closeness: Privacy beyond k-anonymity and l-diversity. Pages 106–115 of: *Proceedings of the IEEE 23rd International Conference on Data Engineering*. IEEE Computer Society Press.
- Lucchese, C., Perego, R., Silvestri, F., Vahabi, H., and Venturini, R. 2012. How random walks can help tourism. Pages 195–206 of: *Proceedings of the 34th European conference on Advances in Information Retrieval (ECIR'12)*. Lecture Notes in Computer Science. Springer-Verlag.
- Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkatasubramanian, M. 2007. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, **1**(1), 3.
- Malinowski, E., and Zimányi, E. 2008. *Advanced Data Warehouse Design: From Conventional to Spatial and Temporal Applications*. Springer-Verlag.
- Mannila, H. 1997. Inductive databases and condensed representations for data mining. Pages 21–30 of: *Proceedings of the 1997 International Symposium on Logic Programming*. MIT Press.
- Marketos, G., Frentzos, E., Ntoutsis, I., Pelekis, N., Raffaetà, A., and Theodoridis, Y. 2008. Building real-world trajectory warehouses. Pages 8–15 of: *Proceedings of the 7th ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM Press.
- MarNIS, Maritime Navigation Information Services. 2009. *Final Report*. MarNIS/D-MT-15/Final Report/DVS/05062009/version 2.0.
- Meratnia, N., and de By, R.A. 2004. Spatio-temporal compression techniques for moving point objects. Pages 765–782 of: *Proceedings of the 9th International Conference on Extending Database Technology*. LNCS 2992. Springer-Verlag.
- Milgram, S. 1967. The small world problem. *Psychology Today*, **2**, 60–67.
- Miller, H.J. 1991. Modeling accessibility using space-time prism concepts within geographical information systems. *International Journal of Geographical Information Systems*, **5**(3), 287–301.
- Miller, H.J. 2010. The data avalanche is here. Shouldn't we be digging? *Journal of Regional Science*, **50**(1), 181–201.
- Miller, N.P. 2003. Transportation noise and recreational lands. *Noise/News International*, **11**(2), 9–20.
- Mlich, J., and Chmelar, P. 2008. Trajectory classification based on hidden Markov models. Pages 101–105 of: *Proceedings of the 18th International Conference on Computer Graphics and Vision*.

- Mokbel, M.F., Ghanem, T.M., and Aref, W.G. 2003. Spatio-temporal access methods. *IEEE Data Engineering Bulletin*, **26**(2), 40–49.
- Monreale, A. 2011. *Privacy by Design in Data Mining*. Ph.D. thesis, Department of Computer Science, University of Pisa, Italy.
- Monreale, A., Pinelli, F., Trasarti, R., and Giannotti, F. 2009. WhereNext: A location predictor on trajectory pattern mining. Pages 637–646 of: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.
- Monreale, A., Pedreschi, D., and Pensa, R.G. 2010. Anonymity technologies for privacy-preserving data publishing and mining. Pages 3–33 of: Bonchi, F., and Ferrari, E. (eds), *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*. Chapman & Hall/CRC Press.
- Moore, B.E., Ali, S., Mehran, R., and Shah, M. 2011. Visual crowd surveillance through a hydrodynamics lens. *Communications of the ACM*, **54**(12), 64–73.
- Nanni, M., and Pedreschi, D. 2006. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, **27**(3), 267–289.
- Nathan, R., Getz, W.M., Revilla, E., Holyoak, M., Kadmon, R., Saltz, D., and Smouse, P.E. 2008. A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, **105**, 19052–19059.
- Newson, P., and Krumm, J. 2009. Hidden Markov map matching through noise and sparseness. Pages 336–343 of: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM Press.
- Nguyen-Dinh, L.-V., Aref, W.G., and Mokbel, M.F. 2010. Spatio-temporal access methods: Part 2 (2003–2010). *IEEE Data Engineering Bulletin*, **33**(2), 46–55.
- Okubo, A. 1980. *Diffusion and ecological problems: Mathematical models*. Springer-Verlag.
- Onnela, J.P., Saramäki, J., Hyvönen, J., Szabó, G., Lazer, D., Kaski, K., Kertész, J., and Barabási, A.-L. 2007. Structure and tie strengths in mobile communication networks. *Proceedings of the National Academy of Sciences*, **104**(18): 7332–7336.
- Orlando, S., Orsini, R., Raffaetà, A., Roncato, A., and Silvestri, C. 2007. Spatio-temporal aggregations in trajectory data warehouses. *Journal of Computing Science and Engineering*, **1**(2), 211–232.
- Ortúzar, J., and Willumsen, L.G. 2002. *Modelling Transport*. John Wiley and Sons.
- Othman, W. 2009. *Uncertainty Management in Trajectory Databases*. Ph.D. thesis, Transnationale Universiteit Limburg.
- Parent, C., Spaccapietra, S., and Zimányi, E. 2006. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer-Verlag.
- Pauly, A., and Schneider, M. 2010. VASA: An algebra for vague spatial data in databases. *Information Systems*, **35**(1), 111–138.
- Pelekis, N., and Theodoridis, Y. 2005. *An Oracle Data Cartridge for Moving Objects*. Tech. rept. University of Piraeus.
- Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsis, I., Andrienko, G., and Theodoridis, Y. 2007. Similarity search in trajectory databases. Pages 129–140 of: *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society.
- Pelekis, N., Frentzos, E., Giatrakos, N., and Theodoridis, Y. 2008a. HERMES: Aggregative LBS via a trajectory DB engine. Pages 1255–1258 of: *Proceedings of the ACM SIGMOD Conference on Management of Data*. ACM Press.
- Pelekis, N., Raffaetà, A., Damiani, M. L., Vangenot, C., Marketos, G., Frentzos, E., Ntoutsis, I., and Theodoridis, Y. 2008b. Towards trajectory data warehouses. In: Giannotti, F., and Pedreschi, D. (eds), *Mobility, Data Mining and Privacy*. Springer-Verlag, p. 189–211.
- Pelekis, N., Kopanakis, I., Kotsifakos, E.E., Frentzos, E., and Theodoridis, Y. 2011. Clustering uncertain trajectories. *Knowledge and Information Systems*, **28**(1), 117–147.

- Peterson, B.S., Baldwin, R.O., and Kharoufeh, J.P. 2006. Bluetooth inquiry time characterization and selection. *IEEE Transactions on Mobile Computing*, **5**(9), 1173–1187.
- Pfoser, D., and Jensen, C.S. 1999. Capturing the uncertainty of moving-object representations. Pages 111–132 of: *Proceedings of the 6th International Symposium on Advances in Spatial Databases*. LNCS 1651. Springer-Verlag.
- Pfoser, D., Jensen, C.S., and Theodoridis, Y. 2000. Novel approaches in query processing for moving object trajectories. Pages 395–406 of: *Proceedings of the 26th International Conference on Very Large Data Bases*. Morgan Kaufmann.
- Potamias, M., Patroumpas, K., and Sellis, T. 2006. Sampling trajectory streams with spatio-temporal criteria. Pages 275–284 of: *Proceedings of the 18th International Conference on Scientific and Statistical Database Management*. IEEE Computer Society.
- Purves, R.S. 2011. Answering geographic questions with user generated content: Experiences from the coal face. Pages 297–299 of: *Proceedings of the 27th Annual ACM Symposium on Computational Geometry*. ACM Press.
- Quddus, M.A., Ochieng, W.Y., and Noland, R.B. 2007. Current map-matching algorithms for transport applications: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, **15**(5), 312–328.
- Raffaetà, A., Leonardi, L., Marketos, G., Andrienko, G., Andrienko, N., Frenzos, E., Giatrakos, N., Orlando, S., Pelekis, N., Roncato, A., and Silvestri, C. 2011. Visual mobility analysis using T-Warehouse. *International Journal of Data Warehousing and Mining*, **7**(1), 1–23.
- Ratti, C., Sobolevsky, S., Calabrese, F., Andris, C., Reades, J., Martino, M., Claxton, R., and Strogatz, S.H. 2010. Redrawing the map of Great Britain from a network of human interactions. *PLoS One*, **5**(12), e14248+.
- Rietveld, P. 1994. Spatial economic impacts of transport infrastructure supply. *Transportation Research-A*, **28A**, 329–341.
- Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., and Andrienko, G. 2008. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, **7**(3–4), 225–239.
- Rinzivillo, S., Mainardi, S., Pezzoni, F., Coscia, M., Pedreschi, D., and Giannotti, F. 2012. Discovering the geographical borders of human mobility. *Künstliche Intelligenz*, **26**(3), 253–260.
- Ruiter, E.R., and Ben-Akiva, M.E. 1978. Disaggregate travel demand models for the San Francisco Bay Area. *Transportation Research Record*, **673**, 121–128.
- Sakoe, H., and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **26**(1), 43–49.
- Sakr, M.A., and Güting, R.H. 2011. Spatiotemporal pattern queries. *GeoInformatica*, **15**(3), 497–540.
- Sakr, M.A., Güting, R.H., Behr, T., Andrienko, G., Andrienko, N., and Hurter, C. 2011. Exploring spatiotemporal patterns by integrating visual analytics with a moving objects database system. Pages 505–508 of: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM Press.
- Salomon, I., Bovy, P., and J.-P., Orfeuill. 1993. *A Billion Trips a Day*. Kluwer.
- Samarati, P., and Sweeney, L. 1998. Generalizing data to provide anonymity when disclosing information (abstract). Page 188 of: *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM Press.
- Scellato, S., Noulas, A., Lambiotte, R., and Mascolo, C. 2011. Socio-Spatial Properties of Online Location-Based Social Networks. *Proceedings of the 5th International Conference on Weblogs and Social Media*. The AAAI Press.
- Scheaffer, R.L., Mendenhall, W., III, Ott, R.L., and Gerow, K. 2005. *Elementary Survey Sampling*. Richard Stratton.
- Shneiderman, B. 1983. Direct manipulation: A step beyond programming languages. *Computer*, **16**(8), 57–69.

- Shu, H., Spaccapietra, S., and Quesada Sedas, D. 2003. Uncertainty of geographic information and its support in MADS. In: *Proceedings of the 2nd International Symposium on Spatial Data Quality*.
- Smouse, P.E., Focardi, S., Moorcroft, P.R., Kie, J.G., Forester, J.D., and Morales, J.M. 2010. Stochastic modelling of animal movement. *Philosophical Transactions of the Royal Society B*, **365**, 2201–2211.
- Song, C., Qu, Z., Blumm, N., and Barabási, A. L. 2009. Limits of predictability in human mobility. *Science*, **327**, 1018–1021.
- Song, C., Koren, T., Wang, P., and Barabási, A.L. 2010. Modelling the scaling properties of human mobility. *Nature Physics*, **6**, 818–823.
- Spaccapietra, S., Parent, C., Damiani, M.L., Macedo, J.A., Porto, F., and Vangenot, C. 2008. A conceptual view on trajectories. *Data & Knowledge Engineering*, **65**(1), 126–146.
- Stange, H., Liebig, T., Hecker, D., Andrienko, G., and Andrienko, N. 2011. Analytical workflow of monitoring human mobility in big event settings using Bluetooth. Pages 51–58 of: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*. ACM Press.
- Tan, P.-N., Steinbach, M., and Kumar, V. 2005. *Introduction to Data Mining*. Addison-Wesley.
- Tobler, W.R. 1987. Experiments in migration mapping by computer. *The American Cartographer*, **14**(2), 155–163.
- Tomkiewicz, S.M., Fuller, M.R., Kie, J.G., and Bates, K.K. 2010. Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transaction Royal Society B*, **365**, 2163–2176.
- Trajcevski, G. 2011. Uncertainty in spatial trajectories. Pages 63–107 of: Zheng, Y., and Zhou, X. (eds), *Computing with Spatial Trajectories*. Springer-Verlag.
- Trajcevski, G., Wolfson, O., Hinrichs, K., and Chamberlain, S. 2004. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems*, **29**(3), 463–507.
- Trasarti, R., Giannotti, F., Nanni, M., Pedreschi, D., and Renso, C. 2011. A query language for mobility data mining. *International Journal of Data Warehouse and Mining*, **7**(1), 24–45.
- Tufte, E. 1990. *Envisioning Information*. Graphics Press.
- Tukey, J.W. 1977. *Exploratory Data Analysis*. Addison-Wesley.
- Turchin, P. 1998. *Quantitative analysis of movement*. Sinauer Associates, Inc.
- Urbano, F., Cagnacci, F., Calenge, C., Cameron, A., and Neteler, M. 2010. Wildlife tracking data management: A new vision. *Philosophical Transactions of the Royal Society*, **365**, 2177–2186.
- Vaisman, A., and Zimányi, E. 2009a. A multidimensional model representing continuous fields in spatial data warehouses. Pages 168–177 of: *Proceedings of the 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*. ACM Press.
- Vaisman, A., and Zimányi, E. 2009b. What is spatio-temporal data warehousing? Pages 9–23 of: *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery*. LNCS 5691. Springer-Verlag.
- Van der Spek, S., Van Schaick, J., De Bois, P., and De Haan, R. 2009. Sensing human activity: GPS tracking. *Sensors*, **9**(4), 3033–3055.
- Versichele, M., Neutens, T., Delafontaine, M., and Van de Weghe, N. 2012. The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent festivities. *Applied Geography*, **32**(2), 208–220.
- Viswanathan, G.M., da Luz, M.G.E., Raposo, E.P., and Stanley, H.E. 2011. *The Physics of Foraging*. Cambridge University Press.
- Wang, D., Pedreschi, D., Song, C., Giannotti, F., and Barabási, A.L. 2011. Human mobility, social ties, and link prediction. Pages 1100–1108 of: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.

- Warf, B., and Sui, D. 2010. From GIS to neogeography: Ontological implications and theories of truth. *Annals of GIS*, **16**(4), 197–209.
- Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of ‘small-world’ networks. *Nature*, **393**, 440.
- Willems, N., Van De Wetering, H., and Van Wijk, J.J. 2009. Visualization of vessel movements. *Computer Graphics Forum*, **28**(3), 959–966.
- Wood, Z., and Galton, A. 2009. Classifying Collective Motion. Pages 129–155 of: Gottfried, B., and Aghajan, H. (eds), *Behaviour Monitoring and Interpretation – BMI: Smart Environments*. IOS Press.
- Wood, Z., and Galton, A. 2010. Zooming in on collective motion. Pages 25–30 of: *Proceedings of the ECAI 2010 Workshop on Spatio-Temporal Dynamics*.
- Xu, J., and Güting, R.H. 2013. A generic data model for moving objects. *GeoInformatica*, **17**(1): 125–172.
- Yan, Z., Parent, C., Stefano, S. and Chakraborty, D. 2010. A hybrid model and computing platform for spatio-semantic trajectories. Pages 60–75 of: *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications – Volume Part I*. LNCS 6088. Springer-Verlag.
- Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., and Aberer, K. 2011. SeMiTri: A framework for semantic annotation of heterogeneous trajectories. Pages 259–270 of: *Proceedings of the 14th International Conference on Extending Database Technology*. ACM Press.

GLOSSARY

abstract model: A model for representing *time-dependent data types* in terms of infinite sets. For example, a time-dependent point value is represented as a function from time into point values. This is to be contrasted with *discrete model*.

anonymization: A process that transforms data about a person to prevent the identification of the person from his/her data.

association rule: A *pattern* that represents relationships between variables that occur frequently in a data set.

attack model: The capabilities that an *attacker* has to attempt discovering some *sensitive information*.

attacker: An unauthorized agent who accesses data to infer *sensitive information* about persons.

cell: In *data warehouses*, the elementary unit of decomposition of a *cube* that contains the *measures* to be analyzed. Each cell is defined by a set of coordinates, one per *dimension* of the cube. It is also referred to as a fact.

classification: A process that associates an entity to a class from a predefined set of classes. In *data mining*, rules to classify entities are usually inferred directly from the data through an automatic learning step.

cloaked algorithm: An algorithm to generate *cloaked locations*.

cloaked location: An area defined for the purpose of blurring the exact position of a moving object. It is also referred to as obfuscated location.

clustering: A process that groups a set of entities into homogeneous groups, referred to as clusters, such that entities in the same cluster share a common property, that is, they are similar with respect to some similarity measure, and are dissimilar (with respect to the same measure) from the entities in the other clusters.

compact representation: A relational representation of *time-dependent data types* where each time-dependent value is stored as a single attribute value within a single tuple.

cube: In *data warehouses*, a multidimensional data structure composed of *cells*, where each cell contains a set of *measures*. Cubes are used to implement *online analytical processing* (OLAP). It is also referred to as a hypercube or multidimensional cube.

data mining: A step of the *knowledge discovery* process that analyzes large amounts of data to identify unexpected or unknown *patterns* that might be of value to an application.

data postprocessing: A step of the *knowledge discovery* process that is applied after *patterns* are extracted by the *data mining* algorithms. This step typically includes *pattern* evaluation, interpretation, and visualization.

data preprocessing: A step of the *knowledge discovery* process where data are prepared before *data mining* algorithms can be applied. This step usually includes data cleaning, where noise in data is reduced, and data preparation, where data are formatted to be mined.

data warehouse: A data repository specifically designed to support the decision-making process. In a data warehouse the information is conceptually represented as a *cube* containing facts and *measures* organized according to *dimensions* and *hierarchies*.

density map: A map that shows the distribution of a phenomenon within the space covered by the map. For example, the distribution of moving objects in a given area may be represented as the number of objects per area unit (i.e., density). Densities are often represented by color-coding, where brighter colors correspond to higher densities.

dimension: In *data warehouses*, a dimension materializes a specific viewpoint for analyzing the facts. For example, space, time, and product are frequently used dimensions. Dimensions may be composed of *hierarchies* of levels. For example, a time dimension may be composed of levels hour, day, week, month, and year.

discrete model: A model for representing *time-dependent data types* in a finite representation. For example, a time-dependent point value can be represented as a polyline in the (x, y, t) space. This is to be contrasted with *abstract model*.

episode: A maximal subsequence of a *trajectory* such that all its *spatio-temporal positions* comply with a given predicate. Examples include stop and move episodes and transportation means (walk, bus, metro, train, car) episodes.

extraction-transformation-loading (ETL): The process that populates a *data warehouse* from one or several data sources. It is a three-step process that extracts data from the data sources, transforms the data, and loads the data into a data warehouse. An ETL process also refreshes the data warehouse at a specified frequency in order to keep it up to date.

flow: An aggregate of multiple movements all starting from the same location and ending at the same location. Examples include count of commuting people or amount of transported goods. A flow can be seen as a vector connecting two locations and associated with one or more aggregate attributes derived from the individual movements that have been aggregated.

flow map: A cartographic representation of *flows* shown in a geographic space. Typically, flows are represented by straight or curved lines connecting the start and end locations with the thickness proportional to the value of the aggregate attributes. Alternatively, the attribute values can be represented by varying levels of transparency or by color-coding.

frequent pattern: In *data mining*, a *pattern* that occurs frequently in a data set.

fuzzy spatial object: A spatial object whose spatial extent is represented by a membership function indicating the membership degree of each point in the extent of the object. The uncertainty is due to imprecision of the borderline of the spatial object. For example, it is not possible to define with certainty the line separating a mountain from the valley beneath. This is to be contrasted with *probabilistic* and *vague spatial objects*.

hierarchy: In *data warehouses*, a set of hierarchically correlated levels of a *dimension* that define the desired aggregation paths for the *measures*.

identifier: In a database, an attribute (or a combination of attributes) whose value uniquely identifies objects. For example, each value of a Social Security Number uniquely identifies the person it is associated to. This is to be contrasted with *quasi-identifier*.

knowledge discovery: The process of extracting useful and nontrivial knowledge from data. It includes three main steps: *data pre-processing*, *data mining*, and *data post-processing*. When applied to trajectory data it is often referred to as mobility knowledge discovery.

lifting: In spatio-temporal databases, a technique used to derive operations for data types varying on space or time from the operations on the corresponding base data types by allowing each argument type to be space or time dependent. For example, a function to compute the distance between two fixed points is lifted to obtain the computation of the distance between two points, fixed or moving.

location-based service: An information service accessible through a mobile device that makes use of the geographical position of the mobile user to determine the most appropriate answer to a user's query.

location k -anonymity: A privacy paradigm for the protection of the mobile user's identity. A user is k -anonymous with respect to position if his/her position is indistinguishable from the position of at least $k - 1$ other users.

location prediction: A predictive model specific to moving objects that is able to forecast the future locations that the object will visit. These models are usually built from the history of past behaviors.

location privacy: An information privacy concern that addresses the protection of personal location information.

map matching: For objects moving within a network, the process of combining the recorded location of the object with the digital map of the network to obtain the real position of the object within the network.

measure: In *data warehouses*, a metric that quantifies facts in a *cube*. For analysis, measures are aggregated along the *dimensions* of the cube. For example, a measure that states the price at which a product is sold in a given branch of a retail store can be aggregated along a branch dimension to obtain the average retail price of the product among all branches.

movement track: The sequence of *raw data* representing the movement of an object for the whole duration of the movement.

online analytical processing (OLAP): Interactive analysis of data contained in a *data warehouse*. It comprises a set of operations such as drill down, roll up, slice, and dice.

ontology: In computer science, a formal representation of a set of concepts within a domain and the relationships between these concepts. The formal representation is equipped with an inference mechanism to perform logical inferences on the ontology.

origin-destination matrix (OD-matrix): A representation of *flows* in the form of a matrix where the rows and columns correspond to different locations and the cells contain aggregated values from the attributes of individual trajectories.

pattern: A representation that characterizes a set of data in a summarized way. In *data mining*, a pattern is a model that represents a summary of the analyzed data set with respect to some criteria. See also *trajectory behavior*.

point of interest (POI): A specific location that is of interest in a particular context. Examples include monuments, hotels, and restaurants. Notice that point of interest is a generic term which does not necessarily mean that the specific location has a point geometry; it can be a line or a region. It is also referred to as place of interest.

privacy: The fact that it is impossible to discover the identity of a person on the basis of the stored data, and that the personal *sensitive information* is protected from unauthorized disclosure.

privacy by design: The approach of embedding *privacy* protection into the design, operation, and management of information processing technologies and systems.

privacy-enhancing technologies: Information and communication technologies for the protection of information *privacy*. They are also referred to as privacy-preserving technologies.

privacy personalization: Users' preferences about the requested type and level of *privacy*.

probabilistic spatial object: A spatial object whose extent is represented by a probability (or density) function determining the probability that a given point be inside the extent of the object. The uncertainty is due to the lack of knowledge, as in the case of the area of sea covered by oil in case of an oil spill at a given position. This is to be contrasted with *fuzzy* and *vague spatial objects*.

quasi-identifier: One or several attributes that from a pragmatic viewpoint (possibly with the use of an external source) can be used to identify a person (or a small set of persons) within the data set at hand. An example is zip code and birth date. This is to be contrasted with *identifier*.

raw data: Data as captured by the sensing devices and transmitted to the receiver. It is a sequence of *spatio-temporal positions*.

raw trajectory: A *trajectory* that holds only *raw data*. Antonym: *semantic trajectory*.

segmented trajectory: A semantic representation of a *trajectory* that segments the trajectory into *episodes*, on the basis of the value of a given expression computed on the attributes of the *spatio-temporal positions*. For instance, a trajectory may be segmented into stop and move episodes on the basis of the instant stillness or speed.

semantic trajectory: A *trajectory* for which semantic information has been recorded: geo-objects, events, semantic annotations. Antonym: *raw trajectory*.

sensitive information: Personal data, such as medical or salary data, that should not be disclosed in association to the person's identity. It is also referred to as private information.

sliced representation: A representation for *time-dependent types* using a *discrete model*. In this representation, a *trajectory* is split into slices defined by disjoint time intervals. The trajectory within the slice is represented by a simple function (e.g., a straight line).

sound trajectory: A *trajectory* that has been preprocessed, making it clean (i.e., without noise), accurate (i.e., *map matched*), and possibly compact (i.e., *compressed*).

space-time cube (STC): A visual representation of space and time as a three-dimensional cube in which two dimensions represent space and one dimension represents time. In an STC, *spatio-temporal positions* are represented as points and *trajectories* as lines.

space-time prism: The set of all spatio-temporal points that can be reached by a moving object given a maximum possible speed and starting and ending spatio-temporal points. A space-time prism can represent the uncertainty about the position of a moving object between two known (measured) positions.

sparsely sampled movement data: Data about spatial positions of moving objects where the positions between the measurements cannot be reliably reconstructed by means of *interpolation*, *map matching*, or other methods, due to too large time intervals between the measurements. An example is the positions of mobile phone calls.

spatial event: An *event* having a specific position in space, which is not necessarily fixed during the time of event's existence. An event may be considered as spatial or not depending on the spatial scale of the analysis.

spatio-temporal position: A position of a moving object at a given instant, represented by a tuple containing at least two data (instant, point), where point is a 2D (x, y) or 3D (x, y, z) spatial point. Other features may complement the tuple: spatio-temporal data such as instant speed or stillness, direction, rotation, acceleration, or semantic annotations that have been captured or inferred such as activity or transportation means.

time-dependent data type: A data type that represents data whose values change over time. For example, the location of a moving object is represented by a time-dependent point.

trajectory: A part of the movement of an object that is of interest for a given application and is defined by a time interval that is included inside the lifespan of the object. The two extreme spatio-temporal positions of the trajectory are referred to as its Begin and End positions.

trajectory behavior: A trend characterizing some *trajectories*. From a data management viewpoint, a trajectory behavior is a Boolean predicate on trajectories that can rely on any characteristic of the trajectories (e.g., *spatio-temporal positions*, *episodes*); contextual data linked to the trajectories (e.g., attribute values of geo-objects linked to stop episodes); and relationships to geo-objects, events, or other moving objects. Examples are the Loop trajectory behavior and the Flock trajectory behavior. It is also referred to as trajectory *pattern*.

trajectory clustering: The process of *clustering* a set of *trajectories* into homogeneous groups according to one or more properties characterizing them. These properties can be spatial (e.g., begin point, end point, length), temporal (e.g., begin time, end time, duration), or dynamic (e.g., *spatio-temporal position*, direction, speed at some instants).

trajectory collective behavior: A *trajectory behavior* that bears on a set of trajectories, that is, a Boolean predicate $p(S)$ where S is a set of trajectories containing more than one trajectory. An example is the Flock trajectory behavior.

trajectory compression: The task of reducing the size of the data stored for a *raw trajectory* by removing as many *spatio-temporal positions* as possible without warping the trend of the trajectory or distorting the data set.

trajectory data mining: A specific type of *data mining* process applied to a set of *trajectories*. It is also referred to as mobility data mining.

trajectory data warehouse: A specific type of *data warehouse* that stores *trajectory* data.

trajectory database: A specific type of *database* that stores *trajectory* data. It is also referred to as a moving object database.

trajectory individual behavior: A *trajectory behavior* that bears on a trajectory, that is, a Boolean predicate $p(T)$ where T represents a trajectory. An example is the Loop trajectory behavior.

trajectory interpolation: Reconstruction of the most probable *spatio-temporal positions* of a moving object between two recorded spatio-temporal positions.

unit representation: A relational implementation of the *sliced representation* where a time-dependent value is stored as a set of tuples, each tuple representing a slice.

vague spatial object: A spatial object whose extent is represented by the spatial extent of the kernel part, which is certainly part of the object, and the extent of the conjecture part,

which could contain part of the object. An example is the habitat of a wild animal: The kernel part is the known habitat, whereas the conjecture part is the region that we suppose to be part of it. This is to be contrasted with *fuzzy* and *probabilistic spatial objects*.

visual analytics: The science of combining automated analysis techniques with interactive visualizations for an effective understanding, reasoning, and decision making on the basis of very large and complex data sets.

AUTHOR INDEX

- Andrienko, G., 149, 240
Andrienko, N., 149, 240
- Behr, T., 42
Berlingiero, M., 315
- Cagnacci, F., 259
- Damiani, M.L., 23
Devoegele, T., 221
Düntgen, C., 42
- Etienne, L., 221
- Focardi, S., 259
- Giannotti, F., 174, 297
Güting, R.H., 42, 240
- Hurter, C., 240
- Janssens, D., 197
- Marketos, G., 23
Monreale, A., 174
- Nanni, M., 105, 197
Neutens, T., 277
- Pappalardo, L., 297, 315
Parent, C., 3
Pedreschi, D., 174, 297
Pelekis, N., 23
- Ray, C., 221
Renso, C., 127, 334
Rinzivillo, S., 197
- Sakr, M., 240
Silvestri, C., 83
Spaccapietra, S., 3, 334
Spinsanti, L., 3, 315
- Theodoridis, Y., 23
Trasarti, R., 127
- Vaisman, A., 62, 83
Van de Weghe, N., 277
Versichele, M., 277
- Wang, D., 297
- Yan, Z., 23
- Zimányi, E., 62, 334

SUBJECT INDEX

- abstract data type (ADT), 43
- abstract model, 51
- ADT, *see* abstract data type
- Air Traffic Control (ATC), 240
- airplane traffic, 240–58
- animal tracking, 263
- association rule discovery, 289
 - Apriori algorithm, 290

- behavior, *see* trajectory pattern
- BerlinMOD, 61
- biologging, 264–5
- Bluetooth, 25
 - tracking methodology, 279–80
- Brownian motion, 270–3, 299–300
- Brownian walk, *see* Brownian motion

- car traffic, 197–220
- clustering, *see* trajectory clustering
- compact representation, 57
- context, 5, 128, 150, 167–71, 172
- continuous field, 73–6
 - data warehouse, 76
- continuous trajectory, 9
- continuously changing geometry, 42
- convoy pattern, 109
- crisp boundary, 84
- crowd management, 282–6
- cryptography-based models, 177

- data cleaning, 26–7
- data compression, 28–30
 - for Bluetooth, 281
- data mart, 64
- data mining, *see* trajectory mining
- data validation, 130–1

- data warehouse, 63–4
 - continuous field, 76
 - definition, 70
 - spatial, 71
 - spatio-temporal, 73
 - trajectory, 62–82
- density map, 160
- diffusion pattern, 266
- direct manipulation, 240, 241, 244–9
- discrete model, 51
- discrete trajectory, 9
- dispersal pattern, 261

- episode, 12, 66, 67
 - move, 12, 13, 33
 - stop, 12, 13, 33, 34
- Extraction, Transformation, and Loading (ETL), 63, 76

- field type, 74
- Flickr, 326, 328
- flock pattern, 108, 138
- flow, 159, 163–7, 173
- Foursquare, 326, 328
- FromDaDy tool, 245–7

- geo-localization, 11
- geo-social data, 315
 - applications, 317–20
- geo-social network, 318
- geo-tagging, 317
- geographic propinquity, 309
- Geographical Information System for Moving Objects (GisMo), 281, 284
- GeoPKDD trajectory data warehouse, 76–81
- Global Positioning System (GPS), 24, 278

- Global System for Mobile communications (GSM), 25
- GPS, *see* Global Positioning System
- GSM, *see* Global System for Mobile communications
- Hermes, 59–60
- home range, 259, 260
- identifier, 176
- IMAGE system, 243
- intentional accuracy degradation, 85
- k*-anonymity, 176
- knowledge hiding, 183
- l*-diversity, 176
- Lévy flight, 272–3, 299–300
- Lévy walk, *see* Lévy flight
- lifting, 50, 71, 74
- linkage attack, 185
- M-Atlas, 129–46, 207, 211
- m-event, *see* movement event
- map matching, 27–8, 56, 95–8
- maritime positioning system, 223
 - Automatic Identification System, 223
 - radar, 223
- maritime traffic, 221–39
- mass event, 282–6
- migration, 259–62
- mobility attractors, 212
- mobility borders, 216–18
- mobility data mining, *see* trajectory mining
- mobility knowledge discovery, 127–9
- mobility measures
 - distribution of displacements, 301
 - distribution of visiting time, 303
 - entropy, 304
 - radius of gyration, 303
- mobility prediction, 215
- mobility statistics, 201
- MOD, *see* moving object database
- motion model, 299–300
- move episode, 12, 13, 33
- movement ecology, 266
- movement event, 249
- movement track, 5, 44
- moving cluster pattern, 109
- moving object database (MOD), 42, 241
- moving point, 45
- network measures
 - average path length, 307
 - clustering coefficient, 307
 - degree distribution, 307
 - tie strength, 308
- OLAP queries, 69–76
- origin–destination matrix (OD matrix), 206
- origin–destination matrix (OD matrix), 163, 173, 198, 207, 211, 212, 220
- outlier detection, 123, 236
- pattern, *see* trajectory pattern
- pedestrian traffic, 277–93
- point of interest (POI), 14, 174, 333
- position accuracy, 86–8
- possible motion curve, 90
- privacy, 34–40
 - of identity, 36
 - of location, 37
 - of semantic location, 38
- privacy by design, 184–92
- progressive mining, 134
 - clustering, 154, 172
- proximity principle, 280
- qualitative movement measurement, 277
- quasi-identifier, 176
- Radio Frequency Identification (RFID), 26
- random walk, 271, 299
- randomization, 177
- raw trajectory, 10, 264
- Réseau de la Navigation Aérienne (RENAR), 243
- RFID, *see* Radio Frequency Identification
- road network, 93
- sampling design, 267–8
- SECONDO, 46, 253–7
 - GUI, 52
 - kernel, 52
 - optimizer, 52
 - pipelining, 53
 - query optimization, 54
- segmented trajectory, 13, 321
- semantic gap, 10
- semantic pattern, 15, 141
- semantic trajectory, 11, 280
- sensitive attribute, 176
- sequence pattern, 19
- sequential pattern, 113
- sharp boundary, 84
- sliced representation, 51
- social environment, 311

- social network, 298, 306, 309, 315, 316
 - geo-social network, 318
- space-time cube (STC), 153, 154, 159, 167, 169, 170
- space-time prism, 88, 92–8
- spatial data warehouse, 71
- spatio-temporal data warehouse, 73
- spatio-temporal database, 42
- spatio-temporal pattern, 14, 140–1, 253, 254–6
- stop episode, 12, 13, 33, 34
- Système de Traitement Radar (STR), 243

- t*-closeness, 177
- tessellation, 73, 164, 169, 173, 187, 207, 211
- time-dependent types, 49, 65, 71
- time geography, 149
- time graph, 154
- time transformation, 154
- tracking technology, 24–6
 - Bluetooth, 25, 279–80
 - Global Positioning System (GPS), 24, 264–5, 267–8, 278
 - Global System for Mobile communications (GSM), 25, 264–5, 267–8
 - Radio Frequency Identification (RFID), 26
- traffic
 - airplane, 240–58
 - car, 197–220
 - maritime, 221–39
 - pedestrian, 277–93
- trajectometry, 264
- trajectory
 - continuous, 9
 - definition, 8
 - discrete, 9
 - raw, 10, 264
 - segmented, 13
 - semantic, 11, 32, 280
- trajectory aggregation, 159, 160, 166
- trajectory anonymity, 185
- trajectory behavior, *see* trajectory pattern
- trajectory classification, 118–21
 - TraClass, 120
- trajectory clustering, 114–18, 153–4
 - density-based, 109, 115, 159
 - flow, 159, 163–7, 173
 - hierarchical, 115
 - k*-means, 114
 - maritime, 232
 - median, 233
 - of locations, 162, 169
 - of situations, 162
 - uncertainty in, 98–100
- trajectory data warehouse, 62–82
 - definition, 73
 - GeoPKDD, 76–81
- trajectory database, 42–61
- trajectory distance function, 116
- trajectory generalization, 159
- trajectory indexing, 58–59, 60
- trajectory location prediction, 121–2
 - WhereNext, 122
- trajectory mining, 105, 233–6, 266
 - global model, 106, 113
 - local pattern, 106, 107
- trajectory pattern, 15, 106, 111, 140
 - Brownian motion, 270–3, 299–300
 - collective, 17
 - convoy, 109
 - diffusion, 266
 - dispersal, 261
 - flock, 108, 138
 - individual, 17
 - interpretation of, 139–40
 - Lévy flight, 272–3, 299–300
 - moving cluster, 109
 - random walk, 271, 299
 - semantic, 15, 141
 - sequence, 19
 - sequential, 113
 - spatio-temporal, 14, 140–1, 253, 254–6
 - validation of, 138–9
- trajectory postprocessing, 138–40
- trajectory reconstruction, 30–4, 131, 268
 - maritime, 229–32
- trajectory uncertainty, 89–90
- Twitter, 326, 327, 329

- uncertainty
 - causes, 85–8
 - definition, 83–4
 - in trajectory, 89–90
 - in trajectory clustering, 98–100
 - of Contributed Geographic Information (CGI), 328–30

- video surveillance system, 277
- visual analytics, 149, 171, 172, 241, 256–7
- visualization, 149, 172

- WEKA, 290