UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Reinaldo BEZERRA BRAGA

Thèse dirigée par **Mr Hervé Martin**

préparée au sein **Laboratoire d'Informatique de Grenoble (LIG)**
et de **École Doctorale Mathematiques, Sciences et Technologies de l'Information, Informatique (EDMSTII)**

LIDU: Location-based approach to IDentify similar interests between Users in social networks

Thèse soutenue publiquement le **19 octobre 2012**,
devant le jury composé de :

Mr Christophe Claramunt
Professeur, École Navale, Rapporteur
Mr Eric Gaussier
Professeur, Université Joseph Fourier, Examinateur
Mr Hervé Martin
Professeur, Université Joseph Fourier, Directeur de thèse
Mme Michela Bertolotto
Professeur, University College Dublin, Irlande, Examinateur
Mr Robert Laurini
Professeur émérite, INSA-Lyon, Rapporteur
Mme Rossana Maria de Castro Andrade
Professeur, Federal University of Ceará, Brésil, Examinateur

Acknowledgments

Over the last three years of my graduate school time at Grenoble Informatics Laboratory, I have learned wonderful things, not only about science, but also about the life. I must say that I have enjoyed every moment of learning and I am very grateful to have different kinds of experience after this period in my life. I really believe this is possible only due to all the people supporting me in several ways and I would like to acknowledge them.

First of all, I would like to thank my supervisor and friend, Professor Hervé Martin, for all the time, attention and support throughout the last three years. He was a great research advisor, giving me many opportunities, which are extremely important to evolve my academic career. He was always besides me to discuss about new research topics as well as to bring important topics of ongoing study.

A special thank to Professor Michela Bertolotto and the PhD student Ali Tahir, mainly for their support and contributions in my work. They had an important participation in my research activities, supporting me in my research activities during my PhD internship at University College Dublin. My short-term mission, sponsored by the MOVE-COST program, was very important to my thesis.

I would like to extend a very special thank to professor Rossana Maria de Castro Andrade. I could say that Rossana is the supervisor of my academic career. She is always encouraging me in all my decisions related to my research activities. I will always be grateful for her help, attention and support.

I also thank to the committee members, Professor Robert Laurini, Professor Christophe Claramunt and Professor Eric Gaussier. Also, I want to thank the staff members of Steamer team, Jérôme Gensel, Marlène Villanova, Sylvain Bouveret, Paule-Annick Davoine, Philippe Genoud et Danielle Ziebelin. Also, I want to thank to other members, Benoit Le Rubrus, Laurent Poulenard and Cécile Saint-Marc. A very special thank to my lovely and great friends and office mates Mouna Snoussi, Betul Aydin, Sócrates de Moraes Medeiros da Costa and Raffaella Balzarini. I will always remember the moments that we shared.

I would like to thank my friends from Grenoble Informatics Laboratory, Maria Eugênia Berezin, Maria Isabel Vergara Gallego, "Los papitos" (Nazim Abdeddaim and Azzeddine Amiar), Asif Iqbal Baba, Ana Bildea, Jun-Young

Bae, Taha Triki, Sofia Bekrar, Franck Rousseau, Béatrice Buccio, Elsa Hollard, Valerie Heitz, Felicette Boero, Marie-Jo Corminier, Bogdan Pavkovic and Maciej Korczynski. I also thank to the visiting students that worked with me, Marc Volaine and Anton Possylkine. We had many fun group activities together.

A very special thank to my Brazilian, Portuguese and French friends in Grenoble, Raquel Benevides, Elton Leite, Isaac Barreto, Shirley Silva, Tales Paiva, Joseane Vale, Samuel Modolon, Renato Eising, "meus guris" (Lucas Haas and Letícia Mazzarino), Soraia Zaioncz, Lucas Schnorr, Fabiane Basso, Luiz Carlos Stefano, Elizandra Britta, Talitha Stefanello, André Ricardo Fajardo, Clayton Fernandes de Souza, José Bringel Filho, Windson Viana, Patrick Gonçalves, Fany Sarrecchia, Hyane Trigueiro (in memoriam), Bernard Priem, Aida Priem, Alexandre Guerry, Nathalie Claisse, Nicole Chetail, Serge and Gisele Rouveyrol and Djamel Hadji. I cannot imagine how I could enjoy all the moments in Grenoble without you.

I also thank my family in Brazil, mainly to Antonio, Lucielma, Telma, Peixoto Junior, Raimundo, Sandra, Bruno Marques, Liduina, Mauro, Karol, Carolina and João Paulo. With their consistent love and support, I achieved all my goals.

Finally, I dedicate this work to my beautiful, lovely and wonderful wife, Carina Teixeira de Oliveira. Carina, you are my best friend, biggest supporter, closest partner and my idol. Thank you for giving me your unwavering love and permanent support. I strive everyday of my life to make you proud!

Contents

List of Publications

- **Reinaldo Bezerra Braga, Ali Tahir, Michela Bertolotto and Hervé Martin**. A Multi-layer Data Representation of Moving Object Trajectories based on Points of Interest. Accepted in the 12th Web Information and Data Management (WIDM'12), collocated with the CIKM 2012 conference. Maui, Hawaii.

- **Reinaldo Bezerra Braga, Ali Tahir, Michela Bertolotto and Hervé Martin**. Clustering user trajectories to find patterns for social interaction applications. In Proceedings of the 11th international conference on Web and Wireless Geographical Information Systems (W2GIS'12), Sergio Martino, Adriano Peron, and Taro Tezuka (Eds.). Springer-Verlag, Berlin, Heidelberg, 82-97. DOI=10.1007/978-3-642-29247-7_8

- **Reinaldo Bezerra Braga, Sócrates de Moraes Medeiros da Costa, Windson Viana de Carvalho, Rossana Maria de Castro Andrade and Hervé Martin**. A context-aware web content generator based on personal tracking. In Proceedings of the 11th international conference on Web and Wireless Geographical Information Systems (W2GIS'12), Sergio Martino, Adriano Peron, and Taro Tezuka (Eds.). Springer-Verlag, Berlin, Heidelberg, 134-150. DOI=10.1007/978-3-642-29247-7_11

- **Reinaldo Bezerra Braga, Sócrates de Moraes Medeiros da Costa and Hervé Martin**. A trajectory correlation algorithm based on users' daily routines. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11). Demonstration paper. Divyakant Agrawal, Isabel Cruz, Christian S. Jensen, Eyal Ofek, and Egemen Tanin (Eds.). ACM, New York, NY, USA, 501-504. DOI=10.1145/2093973.2094059

- **Reinaldo Bezerra Braga, Sócrates de Moraes Medeiros da Costa and Hervé Martin**. A trajectory correlation algorithm based on users' daily routines. In Proceedings of the Third International Conference on Advanced Geographic Information Systems, Applications, and Services (Geoprocessing 2011). Think Mind, Gosier, Guadeloupe, France, 17-22. ISBN: 978-1-61208-118-2. URL=http://www.thinkmind.org

- **Reinaldo Bezerra Braga, Windson Viana De Carvalho, Rossana Maria De Castro Andrade and Hervé Martin.**. Uma ferramenta para

geração de diários de bordo digitais usando anotação contextual e conteúdo multimídia. In Proceedings of the XVII Brazilian Symposium on MultiMedia and Web (Webmedia 2011). Florianópolis, Santa Catarina, Brazil.

- **Reinaldo Bezerra Braga and Hervé Martin**. CAPTAIN: A Context-Aware system based on Personal TrAckINg. In Proceedings of the 17th International Conference on Distributed Multimedia Systems (DMS 2011). Firenze, Italy. URL=http://www.ksi.edu/seke/dms11/DMS

- **Reinaldo Bezerra Braga and Hervé Martin**. Trajectories of Interest (Position paper). In the Workshop on Movement Research: Are you in the flow?, 13th AGILE International Conference on Geographic Information Science, 2010. Guimaraes, Portugal.

List of Figures

List of Tables

Part I

General Introduction

Chapter I

Introduction

Contents

1 The Study of Location-Based Social Network

We observe that mobile phones are not simple call-making devices anymore. They have already become real information centers. With all the embedded sensors like GPS, accelerometer, Internet connection, digital camera, among others, a user easily creates and publishes personal content. Nowadays, any user can quickly take a picture and put it in his/her web-based photo album, as well as register his/her itinerary to go from home to work everyday.

The embedded features of mobile phones allow the creation of several mobile services, called Location-Based Services (LBSs). The LBSs allow to use positioning information anywhere and anytime in order to provide new details about people, events and others [8]. Despite the advantage of LBSs in providing the position and the context information about a user, another type of service has become popular, the Mobile Trajectory Based Services (MTBS) [9]. MTBS is related to users' mobility profiles, or simply users' trajectories, which are fundamentally collections of mobile traces that can reveal moving patterns. A simple MTBS example is a mobile user who finishes his/her work and would like to know if a friend is passing close to his/her work building.

In parallel with mobile phones, social network platforms have emerged as a collaborative solution to provide social connectivity, giving people the capability to create virtual communities and share interests, opinions, and personal information with other users. A social network is generally defined as a social structure composed by users, which are connected due to one or more types of interests, such as friendship, professional activities, locations, and others [10].

The combination of location based services and social networks platforms led to a new research area, named Location Based Social Network (LBSN). This new research area allows to fill the gap between virtual communities (social networks platforms) and real communities (physical world), providing an extensive knowledge about users' interests and behaviors based on their locations. At the same time that a mobile phone provides the embedded features to register, store and publish personal information, the social network becomes an important platform for relating, enriching and sharing user interests.

According to [11], the types of location-embedded and location-based social structures are currently known as location-based social networks and formally defined as follows:

"A location-based social network (LBSN) does not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and texts. Here, the physical location consists of the instant location of an individual at a given timestamp and the location history that an individual has accumulated in a certain period. Further, the interdependency includes not only that two persons co-occur in the same physical location or share similar location histories but also the knowledge, e.g., common interests, behavior, and activities, inferred from an individual's location (history) and location-tagged data".

Based on the study of different types of location-based social networking services, we classify LBSN services according to the combination of social network with Location-Based Services (LBS) and Mobile Trajectory Based Service (MTBS). Hence, we point out two groups: LBSN associated with LBS; and LBSN associated with MTBS.

- **LBS-based:** Applications classified in this group are frequently used by users to share their current positions, such as a bakery, a shopping centre or any place associated with a GPS coordinate. Facebook encourages users to share their current position among friends in its social network. Making use of this service, a user can locate a friend around his/her physical position and interact with him/her (e.g. inviting a friend at the University to lunch at a specific restaurant). In addition, the user can add a note about a visited place, as well as multimedia content (e.g. photo, video or audio). Another example is Foursquare, which a user "check-in" at a place using a mobile device and share his/her current position between friends in the Foursquare network. A differential service provided by Foursquare is the ranking created by place, where the user with the most number of "check-ins" at the same place is crowned "Mayor".

- **MTBS-based:** For this group, the LBSN applications consider the existence of a mobility profile of a mobile user, which is composed by one or a set of trajectories of his/her travel paths. With this service, a user does not only receive basic information about a trajectory. For example, the user A can receive a message from a friend alerting an accident in a certain position of his/her itinerary, because this friend knows the daily itinerary of user A to go from work to home. In comparison to the LBS-based classification, MTBS-based services provide "how and what" information in addition to "where and when" [4].

The context of this PhD thesis is focused on the meaning of Location-Based Social Networks (LBSN) taking into account the use of MTBS-based classification. In short, we try to understand user behavior and find similarities between friends in social networks, making use of users' trajectories as additional information. This study provides the designing of a layer of services, making possible the generation of reasonable data from raw data in order to help the implementation of recommendation systems, as well as the development of a large number of applications.

2 Research motivation

Nowadays, there is a tendency for people to switch from real to virtual communities. Virtual community platforms such as Facebook [12] and LinkedIn [13] provide solutions to social connectivity by giving people the capability to share interests, opinions, and personal information with family, friends, colleagues and others. However, due to the reduction of social interactions in real communities and the absence of context-aware mechanisms in virtual communities, social opportunities are frequently missed. We have noticed that people work or live in different places but have trajectory correlations in their daily routines. The users' daily routines, therefore, can be captured by mobile social applications and shared in virtual communities in order to increase social interactions in real communities.

Mobile social applications have the advantage of using mobile computing services, sensors, Internet connection, accelerometer, and Global Positioning System (GPS) to capture context information about the real environment of mobile objects. With the widespread use of robust smartphones, context-awareness has emerged as a key requirement for the success of mobile social applications, changing the way that users represent or obtain their interests. This adaptation in the way that users register and share their interests can be used anytime and anywhere by requesting location-based services.

Figure 1.1 introduces the research concept of Location-Based Social Networks (LBSN's). As we can see, three users visit locations in a certain region, registering their locations with mobile devices (e.g. Tablet, Smartphone). Therefore, each user will generate his/her trajectory if we sequentially associate the locations with time. Taking into account all trajectories generated, we are able to find the location graph, based on the links between locations. In addition to the location graph, these users are related to a social network graph, which is generated based on the social connection between users.

In the location graph (demonstrated in the bottom-right of Figure 1.1), each point represents a location and the edge between two points denotes that a user have visited two locations in the trajectory. With these edges, we can define a weight for each relation, which represents the correlation strength between two locations connected by the edge [11]. In the social network graph (presented in the top-right of Figure 1.1), each node is a user and the edge between two nodes represents the social connection between

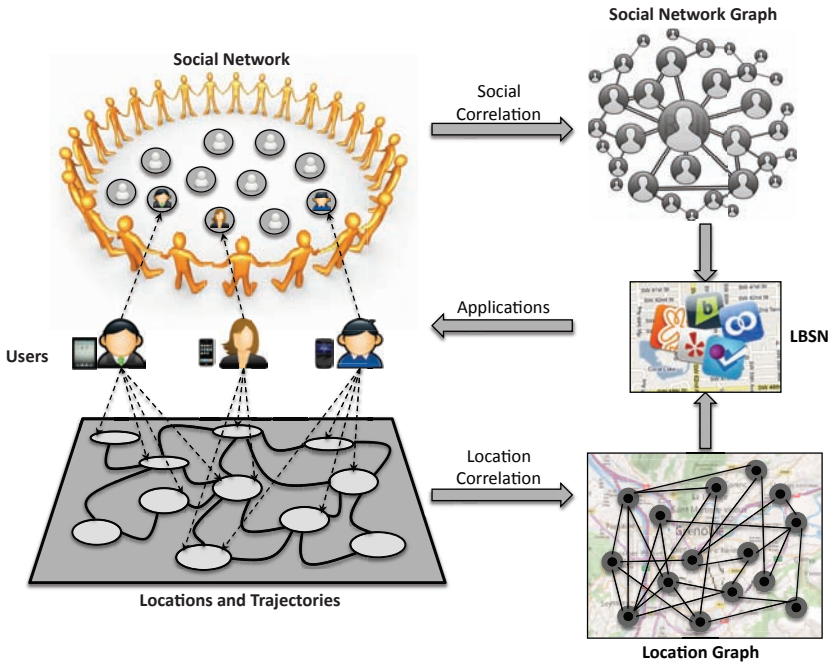


Figure 1.1: Overview of Location-Based Social Networks.

two users. Similarly to the location graph, we can represent the connection strength between two users based on the spatial similarities of them.

The fusion of these graphs provides the spatial and social data, which are the principal requirement for designing location-based social networking services. Making use of these data, LBSN's provides intelligent services, taking into account the social network correlation and location correlation. For instance, with a proactive LBSN service, the user can receive a message or alert from a service provider, noticing that a friend in his/her social network (e.g. Facebook, LinkedIn) passes close to his/her home each Monday and Friday to go from home to work between 09:00 AM and 10:00 AM.

Additionally, with these graphs, LBSN explores the users' interests in terms of social relations and locations in order to provide service for a large number of applications. Some examples of applications are presented as follows:

- **Searching:** Location-based search applications provide a robust interface to the Web, which allow users to handle and request search results in an intuitive way, by focusing a query on a specific geographic region. Location-based search technologies have recently received signif-

icant academic attention, as well as there has been a noteworthy issue for commercial interests, for example:

  - Making use of users' locations, LBSN applications allow to find a skilled person based on the knowledge that he/she has about a specific region. For example, we can use a social network to find skilled friends (who know about a specific location where they have been) in order to obtain touristic suggestions about museums, restaurants, parks and others.
  - Social networks platforms provide different manners to receive a favorable mention about a specific place (e.g. button "Like" on Facebook). With this solution, LBSN applications are able to find most popular places of user's friends in a certain region. For instance, a user would like to know what is the most popular Italian restaurant in Grenoble according to his/her friends' indications in the social network. Hence, taking into account the number of "Like" for each Italian restaurant in Grenoble, the user could receive a ranking from the most to the less popular restaurant.

- **Similar interests:** Social network platforms started the idea for creating groups of interest. Facebook and MySpace named "Groups" and Orkut called "Communities". This feature connects users with similar interests in social networks. With this feature in mind, LBSN applications have been implemented to recognize similar interests between users and automatically create groups of interests. Ipoki[1], Google Latitude[2], Carticipate[3] and Daily Places[4] are the most knew LBSN applications. The following points show other examples of LBSN applications based on similar interests between users.

  - Human location history can reveal habits and interests, making possible the identification of users' daily routines. Therefore, users who have similar daily routines can share common activities (e.g. carpooling solutions). When similar interests between users are

[1] ipoki.com
[2] google.com/latitude
[3] carticipate.com
[4] dailyplaces.com

recognized, both users can receive alerts of possible places to find friends. [14], [15] and [16] present three different approaches to identify similarities between users inferred from their location histories.

- Making use of user similarities inferred from locations, we can create groups of interest in order to assist in the execution of similar activities, like going to the cinema. Hence, a user can easily start a group action (e.g. buy tickets at a group price), by alerting his/her appropriate friends in the social network.

- **Recommendation:** Although the introduction of the two previous groups of LBSN applications, all the enriched data generated by them can be applied and extended in this classification. Recommendation systems have been studied as a solution to "predict" or "recommend" information based on the user history. In general, the approaches in this area focus on personalized recommendation, which the information is recommended according to the user's past behavior. However, in LBSN applications, the information is frequently recommended taking into account the past behavior of a similar group of users. We present some examples of recommending applications in location based social networks [11]:

  - In [16], the authors propose a user-based collaborative filtering. In summary, they assume the existence of a scenario where the similarity between each pair of users is incorporated into a collaborative filtering model. They intend to organize a personalized location recommendation system, which allows to perform locations matching and find user's preferences. The main feature is related to the participation of similar users, who vote in a similar manner on similar items. Therefore, with the use of collaborative filtering [17][18], they identify similarities between users and items. Consequently, recommendations can be sent to users with similar interests. For example, if two users are friends in the social network and they have similar behaviors in their location histories, the recommendation system can send an alert to one user about a near place where the other user has already been.

- User-based collaborative filtering can be affected when the number of users increases in the system. It occurs due to comparison of each pair of users, increasing the processing time and decreasing the performance. As a continuation of this work, [19] and [20] address their researches by proposing a location-based collaborative filtering. Since we have the limited geographical space, this model is more practical for a real application. That is possible because the comparison is performed between locations based on users' histories. The authors state that the main challenge is how to embody a user's behavior to the location-based model.
- In general, tourists would like to receive a maximum amount of information before traveling to an unknown city. In [14], the authors introduce a data mining solution for finding the most interesting locations in a city as well as some possible travel sequences among locations. The data mining process is performed based on the friends' location history in order to recommend relevant information.
- Taking into account that users want to know about the route conditions to go from a departure position to a arrival position, a LBSN application can use location histories of friends in social network to recommend trip plans for users, by observing specific days of the week as well as according to the time of the day. In [6][21], the authors present two different issues for travel planning. The authors make a trip plan in terms of the knowledge learned from many users.
- Another interesting way to use LBSN applications for recommending information was introduced by [22]. They show two different points of view to recommend information through the relation between location and activity. On the one hand, the recommendation is done for the most popular activities that can be performed in a given location. On the other hand, the application can recommend the most popular locations for executing a given activity. These two points of view for recommendation systems based on LBSN services led the development of a large number of LBSN applications.

In spite of the existence of several solutions in terms of LBSN applications, this new research area also brings scientific challenges related to traditional

and new problems involving social networks, mobile computing technologies and spatial data representation. These challenges can be classified according to the following groups:

- **Virtual communities x Real communities:** We have noticed that social network platforms could make use of context-aware mechanisms in order to improve social contacts in real communities [23]. We argue that these virtual platforms should be based on users' daily routines to increase social interactions among mobile users in real communities. Therefore, the main challenge is related to the definition of the relationships between users of social networks.

- **Raw data x Reasonable data:** The capability to capture a sequence of positions is the starting point of managing movement in LBSN. However, the statement of a structured approach of data is the key for making reasoning based on trajectory in order to fill the gap between raw data and reasonable information for designing a LBSN service. Indeed many applications need a more structured recording of movement and semantics, e.g. as a temporal sequence of journeys, each one occupying a time interval in the object's lifespan and taking the object from a departure point to a destination point [24]. Therefore, after obtaining the social network and location graphs, modeling data becomes necessary for important operations, such as: i) to indentify patterns between two or more correlated users; ii) to query information about friend in the social network; iii) to optimize intelligent transport systems and reduce pollution from vehicle emissions (e.g. motivating users to use car pooling alternatives based on the routes used to their friends). In summary, the main challenge is related to provide reasonable data, taking into account the semantics of data.

- **User interests x Connecting strength:** As a social network should reveal a large number of user interests, the location becomes only an additional feature of the user, but also important information in a LBSN. Besides the location, the presence of annotations, videos and photos can be used by LBSN services to increase the connecting strength between two users. However, this connecting strength in a LBSN depends on the information provided by other graphs, linking them with the graphs

presented in Figure 1.1. Hence, the correct definition of user profiles may facilitate the association between user trajectory and context information.

- **Points of Interest (PoI) x Properties of data:** In general, PoI is a location about which information is available. PoI can be represented by an identifier containing a set of coordinates or a three-dimensional model of a building with names in different languages, information about opening hours, and the address. While the PoI data model is an important starting point for the data representation, the relation between data at various abstraction levels is still a challenge. For instance, the data model has to be able to answer a query with exact knowledge of the data abstraction level, as well as to compute representations of different types of data, taking into account each abstraction level. The data model helps the classification of spatial knowledge based on points of interest (e.g., bakery, apartment, campus), spatial relations (e.g., near my apartment) and geographic entities (e.g., Grenoble). Therefore, it is important to design a robust data model for describing the spatial environment and the contextual data, taking into account their different abstraction levels.

In addition to these challenges, we identify other significant research issues in location-based social networks, such as privacy, quality of information and streaming databases. Researchers have explored these points, but they will not be extensively discussed in this thesis.

Taking into account these recent progresses and the mentioned challenges involving location based social networks, we have focused our work to provide a layer of services able to fill the gap between data acquisition and enriched data application (see Figure 1.2). While studies in context-aware systems have reached sufficient maturity, in location-based social network and especially in mobile social networking applications, where no correlated data is directly assumed, many issues and questions are still open. It occurs mainly due to the recent combination of different research areas, such as social networks, location data management and mobile computing technologies. However, for the same reasons, the objective of developing a layer of services for mobile social networks is, at the same time, challenging and stimulating, since we

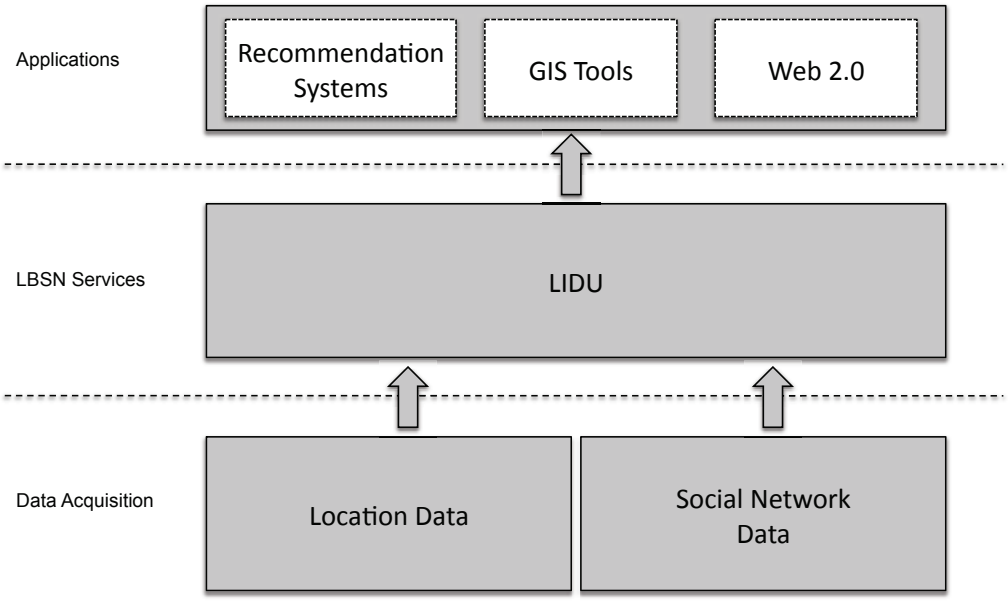


Figure 1.2: Context of our layer of services.

have to deal with the combination of personal interests as well as an extreme setting characterized by a large number of data.

The rapid increase of diverse kinds of space-associated data, such as measurements from mobile sensors, GPS tracks, or georeferenced multimedia provides prospective sources of useful knowledge and information. These data require scalable methods of analysis, which need to consider the particular features of the geographical space, such as heterogeneity, diversity of characteristics and relationships, spatio-temporal autocorrelation. In parallel, the frequent activities of multiple users in social networks enable us to extract collective social knowledge that provides enriched information based on user's interests to be associated with particular features of the geographical space. Figure 1.2 shows these two data sources in the bottom level of our research context.

3 Thesis contribution

In this thesis, we present LIDU: a Location-based approach to IDentify similar interests between Users in social networks (see Figure 1.2). Since we observe the necessity to increase social interactions in real communities and the large widespread of smartphones and social networks, we propose a layer

approach for location-based social network based on user's daily routines. The key idea is to offer a service layer that allows to capture, store and process users' daily routines in order to find similarities between multiple user trajectories and near points of interests between two or more users. Firstly, we use smartphones and their sensors to capture users' daily routines and context information. Secondly, all information is transferred and stored in a relational database located on a server application. Finally, we use a trajectory correlation algorithm to find the similar interests among two or more friends. Therefore, the main contributions of this thesis is classified in three parts:

- The first and general contribution of this thesis is associated with the designing of a flexible multi-layer data model for mobile social applications based on Points of Interest (PoI). We propose a conceptual model to be adaptable and acceptable to a set of generic features as well as to assist developers in designing solutions with the inherent complexity of trajectory semantics. Besides that, we show how our trajectory modeling could offer mobile social applications with direct support for trajectories. Therefore, this data model makes available two main contributions, which are: i) the analysis of requirements for mobile social applications according to their data representation; ii) the proposal of a multi-layer data model to support the knowledge of trajectory semantics.

- A second and intermediary contribution is related to the study and employment of the capabilities provided by clustering algorithms to analyze user trajectories and extract relevant information from them. Our approach has focused on clustering and aggregating multiples trajectories generated by the same user in order to identify habits or interests. Based on each user interest, we find similarities between multiple user trajectories. Consequently, the near points of interests between two or more users are identified. The use of clustering algorithms in our approach provided important avenues for providing services to Location-based Social Network (LBSN) applications.

- The last and most specific contribution is linked to the designing of a correlation algorithm to identify similar interests between users. While there has been an increase in virtual communities (e.g. facebook, twitter, others), we can use this source of knowledge to increase the number

of social interactions in real communities. We have noticed that social network platforms do not make use of correct context-aware mechanisms in order to improve social contacts in real communities. Therefore, this correlation algorithm considers that these platforms should be based on users' daily routines to increase social interactions among mobile users in real communities.

In summary, we have been always motivated to offer a reasonable and useful source of knowledge and information to system designers and developers. Nevertheless, we know that these solutions require a scalable data representation in terms of points of interests, which need to consider the particular features of the geographical space, such as heterogeneity, diversity of characteristics of relationships, and spatio-temporal autocorrelation.

4 Thesis outline

In the second chapter, we introduce the context of movement representation and its main features. After that, we present a conceptual view on moving object trajectories, which brings the necessary support to understand and analyze spatio-temporal data. We point out a review of the state of the art involving similarity analysis of moving object trajectories, in the context of spatial, temporal and spatio-temporal resemblance. Finally, we give an overview of the main challenges related to frequent problems in analyzing dissimilar trajectories as well as some solutions to solve these problems.

In the third chapter, we focus on the context of clustering methods as a promising solution to identify trajectory patterns of individual and a group of moving objects. We address our attention on spatio-temporal clustering methods to find trajectory patterns of moving objects in geographic spaces. We present a review of the state-of-the-art of existing spatio-temporal clustering approaches, by showing the application of these algorithms in different scenarios.

In the fourth chapter, we present the conceptual definitions of social networks and their virtual communities. Next, we introduce the main definitions of points of interests according to the representation specified by W3C PoI working group [3]. After that, we show other concepts of points of interest that have been used in some approaches. Finally, we introduce the current

works in this area as well as we show the main concepts and definitions related to LBSN's.

In the fifth chapter, we introduce our location-based approach to identify similar interests between users in social networks (LIDU). The key idea is to provide a layer of services to acquire daily routines in order to find near points and, consequently, increase social interactions in real communities. We also present a flexible multi-layer data model for mobile social application context based on Points of Interest (PoI). Besides that, we discuss how our trajectory modeling could offer mobile social applications with direct support for trajectories. Next, we present an algorithm to execute the trajectory correlation process based on Minimum Bounding Rectangles (MBRs) and the Hausdorff distance (HausDist) for finding spatial similarities. Furthermore, we used Parzen-window technique to identify similarities between temporal data.

In the sixth chapter, we present the evaluation of our approach in different scenarios. We start showing the implementation of our mobile application, which acquires trajectory data and context information of users. Next, we show the results obtained in the evaluation of the OPTICS algorithm to discover the best representative trajectory of each user, which determines the user's daily routine. Finally, we introduce the results of our trajectory correlation algorithm, which finds similarities between multiple user trajectories based on each user preference and PoI. The results demonstrate that the near points of interest between two or more users can be identified. Therefore, we conclude that our research provides interesting avenues for exploring Location-based Social Network (LBSN) applications.

Finally, the last chapter introduces the final conclusion of this thesis and presents the avenues for exploring LBSN services.

Chapter II

Moving object trajectories and their similarities

Contents

A spatial trajectory can be defined as a trace generated by a moving object in a geographical space, which is represented by a sequence of geospatial coordinate set and a timestamp [11]. Sharing of trajectory data has increased over the last years due to the availability of sophisticated Web and mobile applications (including social networks). For instance, users can easily record and share their trajectories over time based on their daily trips.

While the capability to capture and record a sequence of positions is the starting point of managing movement, designing an application based on trajectory data requires a structured and well-defined data representation. Indeed many applications need a more structured recording of movement and semantics, e.g. as a temporal sequence of journeys, each one occupying a time interval in the object's lifespan and taking the object from a departure point to a destination point [24]. Moving objects can represent vehicles delivering posts within a given region, migration of animals, and a person that goes from home to work and back everyday.

Moving object trajectories are complex entities that combine unprocessed movement features (when and where the moving object is) with a diversity of semantic data, which determines a specific knowledge. The semantic data is needed to support a significant understanding of the moving object trajectory, such as a user's daily routine. Hence, a data representation of moving object trajectories has to provide facilities to save, maintain and operate queries about spatial, temporal and enriched information in the database [25] [26].

In this way, a data model is a representation of a set of information, which the data are semantically related. A data model provides an easy way to manipulate trajectory data, to use structured query languages, to specify profiles through movements, to create and compare profile groups, and others. In addition, it is important to consider a diversity of semantic data that enriches the knowledge on these trajectories. Therefore, the definition of a data model associated with trajectories is an important step to design an application based on moving object trajectories.

Making use of these data, the analysis of movement data can be performed. The similarity analysis of moving object trajectories has been one of the most important research topics when we consider the study of movement. In the last years, the number of approaches involving the identification of similarities between trajectories has grown significantly. In general, the approaches have focused on similarity analysis based on spatial, temporal and spatial-temporal information in order to find similar features of moving object trajectories.

All things considered, we start this chapter by introducing the context of movement representation and its main features. After that, we present a conceptual view on moving object trajectories, which brings the necessary support to understand and analyze spatio-temporal data. Along this line, we point out a review of the state of the art involving similarity analysis of

moving object trajectories, in the context of spatial, temporal and spatio-temporal resemblance. Finally, we give an overview of the main challenges related to frequent problems in analyzing dissimilar trajectories as well as some solutions to solve these problems.

1 Movement representation

The study of moving objects has led to important advances in many fields such as transportation security administration, computer networks, recommender systems, weather forecasting, etc. Moving objects is generally defined as the continuous evolution of a spatial object over time, all along the lifespan of the object [27]. This characteristic related to continuous change in positions makes the data analysis more complex than static objects in different aspects, mainly when it involves the study of similarities among spatial objects.

Guting et al. support the approach of moving regions, which allows for example recording the changing geometry of pollution clouds and flooding waters [28]. In other words, the geometry of a moving object can be of any spatial type and is defined by a function from a temporal domain to a range of spatial values [24]. Therefore, the size or feature of a moving object affects in several aspects the study of moving objects.

In the spatio-temporal context, we may consider as an example the observation of an airplane that is disjoint with a hurricane at a certain instant, and later it goes in direction to the hurricane, and finally locates inside the hurricane. This relationship between these two moving objects is generally named as enter [29]. A second example is related to the study of a moving object that changes its position in a free movement space, such as the bird migration tracking. On the other hand, the spatial constraint networks come as a third example of the study of moving objects. Normally, this last study corresponds to the analysis of moving objects (e.g. cars, trucks, buses) that travel in routes.

While the features (e.g. shape, speed, dynamicity) of a moving object can affect in the movement analysis, the identification of the geographic space is also an important step to design the services that will be provided. On the one hand, there is the unconstrained space, which the moving object does not have constraints to move from a position to another (e.g. birds in the sky). On

the other hand, we have constrained spaces, where the moving objects have limitations to move from a position to another (e.g. cars in road networks). Since the geographic space is defined, two other features have to be well defined and represented: the position (space and time) and the movement.

A moving object position is defined according to the reference related to space and time. In other words, the space where the moving objects progress can be geo-referenced or not. Geo-referenced objects move in a geographic space, such as vehicles, animals and people. However, non-geo-referenced objects perform movements in no geo-referenced spaces, for example, a pen rolling over a paper surface [30]. In general, a point ($\mathbf{p}$) represents the geographic position in a specific space is and the coordinates are planned taking into account the selected reference. Therefore, the coordinates can be denoted in different forms, for example: geographic (World Geodetic System - WGS84 [31]); Cartesian reference system; and in terms of (latitude, longitude).

The time, similarly to the space reference, has to be defined according to a reference system. To better understand the time reference, we give a brief explanation about absolute time and relative time. Absolute time (A_{time}) is based on the Coordinated Universal Time (UTC), which is the primary time standard by which the world regulates clocks. Relative time (R_{time}) can be determined by the elapsed time between two absolute times. Hence, the association of a point ($\mathbf{p}$) in the space and the absolute time (A_{time}) represents the position of a moving object.

Once space and time are described in order to represent a moving object position, we can start our explanation about movement representation. As previously mentioned, understanding of movement has noteworthiness in several areas of science, such as biology, transportation engineering, meteorology, sociology, and others. In the geographic domain, movement is defined as a change in location of a moving object over time, while the object maintains the same identity [32] [30].

The identification of the movement characteristic is an important step to represent a movement, taking into account the nature of change. Table 2.1 shows the types of changes and their respective characteristics in a geospatial context [33].

Temporal properties are also represented by different characteristics of change, which are birth, movement and death [34][35]. Therefore, a behavior of a moving object can be determined taking into account all these represen-

Changes	Characteristics	Examples
Location	Movement	Trajectory movement of a person
Existence/ non-existence	Appearance/ disappearance	The existence of a road and its non-existence
Attribute	Increase/ decrease	Acceleration of a vehicle
Geometric	Expansion/ contraction/ deformation	Change in shape, size or extent of a region
Identity	Transformation	Change from one characteristic to another, e.g.: forest to grassland

Table 2.1: The types of change and their respective characteristics in a geospatial context.

tations and definitions previously presented.

The authors in [36] have addressed their researches in the aspects that can affect the behaviors and features of moving object movements. They classify these aspects in four categories: aspects of space; aspects of time; aspects involving the activities of moving objects; and aspects involving diverse spatial, temporal, and spatiotemporal circumstances. These aspects are explained as follow:

- **Aspects of space:** This first classification is composed of different types of aspects, which are: the accessibility in terms of distance, roads and availability; presence of objects at a specific location (e.g. buildings, trees and other objects); the physical surface situation (e.g. land, forest, water); the elevation and/or quality of the terrain; the use of the location for a certain type of activity (e.g. industry, agriculture or service); and the sense meaning of a place for a moving entity (work, home, school).

- **Aspects of time:** comprises temporal routines, such as daily, weakly and monthly; statement of conventional activities (e.g. working day); and physical features (e.g. duration of daylight).

- **Aspects involving activities of moving objects:** This can indicate the individual characteristics, such as gender, age, occupation and ed-

ucational level; types of moving objects (e.g. vehicles, bike, animals); geographic space constraints (constrained or unconstrained); purpose and/or causes of movement; and activities executed during movements.

- **Aspects involving diverse spatial, temporal, and spatiotemporal circumstances:** these last aspects involve the information about rush hours (e.g. from work or shopping centers); and the influence of weather, traffic news, and others.

[37], [38] and [39] have brought these aspects in different ways. They classify the behavior of a moving object only in two aspects. The first one is related to the visible aspects at a certain instant of time, such as the speed, the position, acceleration, direction, and others. The second aspect is directly linked to the relative measures in time intervals, for example, rotation angle and relative speed.

Based on the discussion of movement representation, in this thesis, we consider a moving object as a mobile user in an urban center. Besides that, we are interested to analyze changes of locations in a geospatial context (see Table 2.1). Therefore, we bring these general definitions and concepts in our scenario, always respecting their stable and solid characteristics. Finally, we address our approach in the aspects related to trajectory data, which is discussed in the following section.

2 Trajectory representation

Several approaches address their researches in the analysis of spatial and temporal aspects of moving objects in order to define the conceptual model of a trajectory. A trajectory can be defined as a trace registered by a moving object in a geographical space. It is generally illustrated by a sequence of ordered points ($\mathbf{p}$), which each $\mathbf{p}$ is composed by geographic coordinates and timestamp such as $\mathbf{p}= (x, y, t)$[40][41]. Based on this definition, Spaccapietra *et al.* describe a trajectory as the user defined record of the evolution of the position (perceived as a point) of an object that is moving in space during a given time interval in order to achieve a given goal [24]. These same authors declare that the key to a growing number of applications is associated with the analysis of trajectory data, focusing on the understanding and management

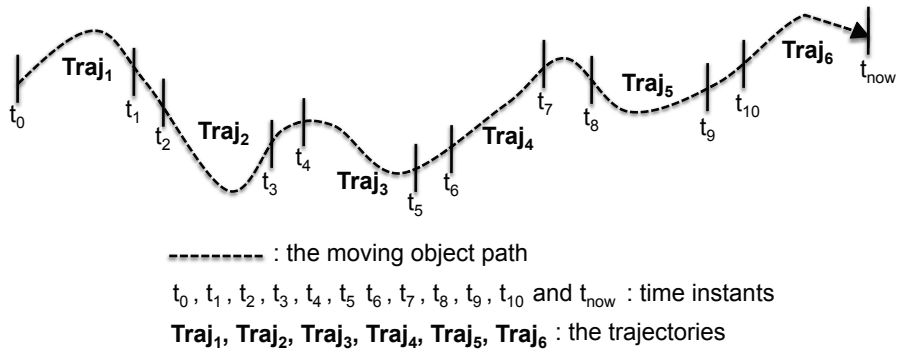


Figure 2.1: Path and trajectories of an ongoing moving object, represented by spatio-temporal information.

of complex events that involve moving objects (e.g. city traffic management, worldwide courier distribution, similar interests between users). Therefore, the study of trajectory representation has evolved over the last years.

In [40], the author proposes a trajectory data representation, derived from the user activity, to typify the movement of a moving object. A historical overview of the evolution of space and time representation in terms of users trajectories was presented in [42]. These two works brought an understandable way to associate space and time with a trajectory, facilitating the creation of other approaches of trajectory representation.

Two other approaches have been proposed by [43] and [44], focusing on the trajectory representation based on geospatial lifelines. In other words, the trajectory representation is modified taking into account the detail level that is being executed. This basically means that the representation derives from time intervals that a moving object stays in a certain space.

When we observe the representations of moving objects and trajectories, we can clearly distinguish some interchangeable definitions. Therefore, before starting the discussion about trajectory similarities of moving objects, we introduce some differences between these two representations. In the case of moving object, the position records derived from time and space functions are directly related to the whole lifespan of the object. Furthermore, a trajectory is associated with a particular time interval included in the object lifespan. Figure 2.1 helps to understand these differences.

Figure 2.1 was initially presented by [24]. It shows the path of a moving object, which is composed by a set of trajectories. Hence, we conclude that a

moving object can generate many trajectories during its lifespan. For example, a person that goes from home (instant t_0) to work (instant t_1), generating the first trajectory ($Traj_1$). After that, the user goes from work (instant t_2) to a restaurant (instant t_3), he eats and goes back from the restaurant (instant t_4) to work (instant t_5). With this in mind, the moving object continues to register his path with many trajectories during its lifespan.

The representations of moving objects and trajectories provide some facilities to understand and analyze spatial-temporal data. Besides that, we can obtain enriched information about users' interests from their trajectories. As the matter in question of this thesis is to compare trajectory similarities of moving object, the following section introduces some related work in this subject.

3 Trajectory similarities of moving objects

In the literature, the use of similarity algorithms is generally related to the comparison between two objects. However, it is important to find the most relevant information that will be analyzed in order to identify the similarities between them. In general, this most relevant information is specified according to its application [45]. Likewise, [46] and [47] have introduced two groups of processes for facilitating the discovery of main information to perform similarity analysis. These two groups are classified as follow:

- **Partial analysis of similarities:** Supposing that only some features of two objects are similar, we need to discover what are these features and analyze the similarities between them.
- **Complete analysis of similarities:** In this group, the two objects are compared as a whole.

These processes of partial and complete analyses can be explored in terms of distances between trajectories of moving objects. Trajectories, as previously explained, can be used for simple and complex analysis and applied in different domains. For example, taking into account the existence of locations histories of two users, we can estimate the similarities between them. Therefore, we are able to extract user's interest derived from spatio-temporal information as well as to identify similar interests between these users.

In [35], the authors address the research in the analysis of similarities in dynamic behavior of moving objects. The main motivation is associated with the development of a conceptual and methodological framework for identifying similarities from trajectories of moving objects by performing a quantitative analysis. These authors indicate some research questions, which they consider an important step for analyzing and classifying the movement behavior of various moving objects. These questions are presented as follow:

1. Assuming the existence of different types of moving objects in space and time, how similar is the movement of these moving objects?

2. Once the authors are trying to simulate movements of particular objects in the space, they ask the question: how similar are artificial simulated proxies to the corresponding moving objects of reference?

3. Indeed, in terms of movement, how to define similarity between movements as a relevant measure to any kind of moving object? Besides that, how to automatically identify spatio-temporal similarities between trajectories of moving objects?

4. Taking into account the movement prediction, what are the main benefits obtained from similarity analysis in order to predict movement under different circumstances? Furthermore, how to recognize if objects with similar characteristics perform similar behaviors in a specific situation?

We can observe in this example that the use of questions is important to identify possible requirements according to the application requirements. Once they receive the answers for these questions, they are able to figure out some of the needed information for identifying similarities between objects.

Taking into account the discrete and continuous flow in migration maps [48] as well as the functional view of a data set [49], Andrienko and Andrienko have proposed the concept of *derivative movement characteristics* in [36]. With this concept, the authors show that several movement characteristics can be derived from the positions of moving objects at different time instants in order to help the designing of visual analytics methods for massive collections of movement data.

Firstly, they have defined some movement features for a group of moving objects, which they named the Momentary Collective Behavior (MCB). MCB

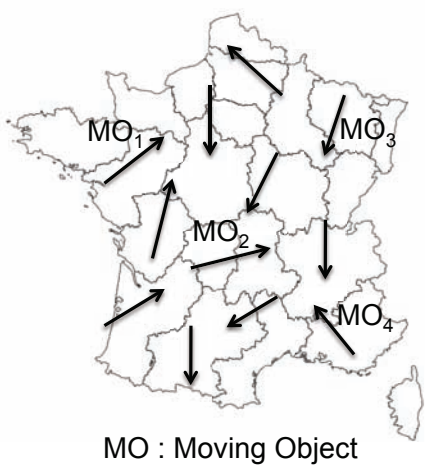


Figure 2.2: The Momentary Collective Behavior (MCB) at a single time instant.

exists when the movement of a set of moving objects (MO) occurs at some single time instant (t_i) (see Figure 2.2). Making use of the characteristics of a MCB, the authors can measure differences and similarities of moving object behaviors at different time instants or among different groups of moving objects.

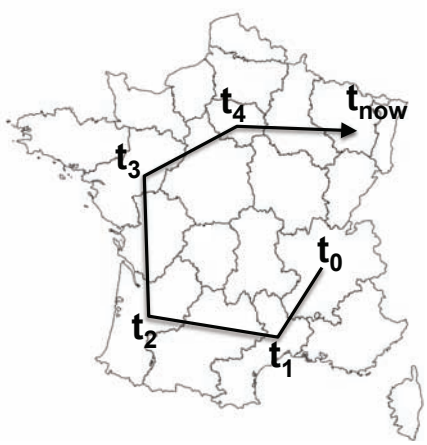


Figure 2.3: The Individual Movement Behavior (IMB) of a specific moving object.

In addition to that, the same authors have proposed a definition to represent a single movement over time of a moving object, called the Individual Movement Behavior (IMB) (see Figure 2.3). The features related to an IMB

are directly associated with its own characteristics, which are: the entire distance traveled; the path, or trajectory, recorded by the moving object in the space; the change of direction and speed; and the vector derived from the object movement (e.g. orientation from initial to final position). With two IMB's we can compare the similarities between different moving objects as well as between different time intervals of the same moving object at different time intervals.

A third group of behavior was called Dynamic Collective Behavior (DCB), which provides some facilities to analyze movement features of multiple moving objects over a time interval (e.g. many time intervals). The key idea is to offer an easy description of movement data for all moving objects during the referred time interval. Therefore, the analyst is able to compare the similarities and differences between DCB's, for example: different groups of moving objects during the same time interval or during different time intervals; and same group of moving objects during different time intervals [36].

In the last couple of years, the number of approaches involving the identification of similarities between trajectories has grown significantly. In general, the approaches have focused on similarity analysis based on spatial, temporal and, most recently, spatial-temporal information in order to find similar features of moving object trajectories. We explain in more detail some of these approaches in the following sections.

3.1 Similarity analysis based on spatial Information

Several works have focused on the analysis of spatial similarity by observing the geometric characteristic of linear objects. This manner to analyze similarities has been widely applied in visual analysis, artificial intelligence, cartography and pattern recognition. Formally, similarity analysis based on spatial information is usually applied to determine spatial similarities (e.g. distance) between two geometric objects (e.g. trajectories of moving objects).

In the literature, different algorithms for computing distance have been proposed in order to find similarities between linear objects. We start by introducing a proposal that offer a method to compute average distance between two lines [50]. The computation of average distance is performed by dividing the total surface of dislocation by the size of reference line. The surface is obtained by linking the departure and arrival points of each line. To better

understand the concept of surface and reference line (L_1) we show Figure 2.4.

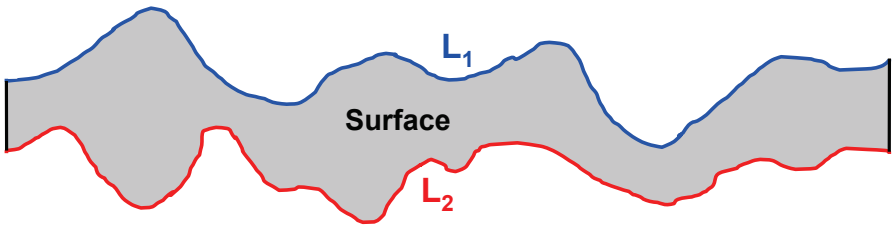


Figure 2.4: Surface representation between two lines.

The computation of average distance is an easy way to obtain derived information based on the spatial variation between two lines (or trajectories) [51]. However, the use of that information alone is not enough to obtain information about the similarities between two lines. A well-known technique to solve this problem is computing the maximum distance between two lines, as proposed in [52].

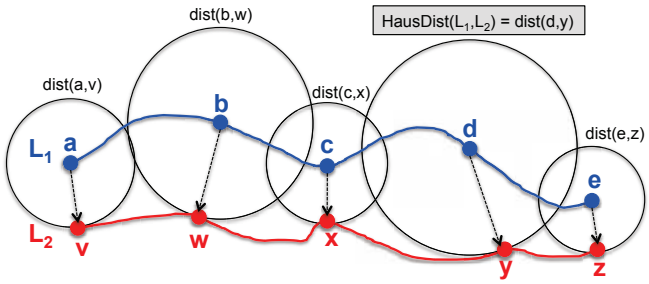
3.1.1 Hausdorff distance

The Hausdorff distance has been widely used and discussed when we talk about spatial similarities between linear objects. This technique computes the maximal gap between two lines. By definition, the Hausdorff distance is generally used to find similarities between two sets of points (e.g. trajectories) [52] [53]. Using this measure, a line (L_1) is considered similar to another line L_2 iff every point in L_1 is close to at least one point in L_2. Conventionally, the Hausdorff distance HausDist(L_1,L_2) is computed by the Max-Min distance from L_1 to L_2. Figure 2.5 illustrates two examples of Hausdorff distance, HausDist(L_1,L_2) and HausDist(L_2,L_1).

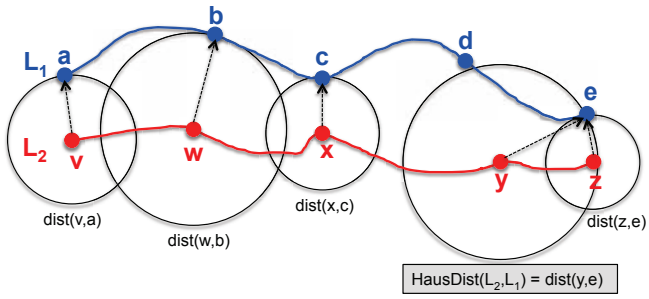
The HausDist(L_1,L_2) and HausDist(L_2,L_1) distances, presented in Figure 2.5, are computed based on Equations 2.1 and 2.2, respectively.

$$HausDist(L_1, L_2) = \underset{p_1 \in L_1}{\text{Max}} \left(\underset{p_2 \in L_2}{\text{Min}} (dist(p_1, p_2)) \right) \tag{2.1}$$

$$HausDist(L_2, L_1) = \underset{p_2 \in L_2}{\text{Max}} \left(\underset{p_1 \in L_1}{\text{Min}} (dist(p_2, p_1)) \right) \tag{2.2}$$



(a) HausDist(L_1,L_2).



(b) HausDist(L_2,L_1).

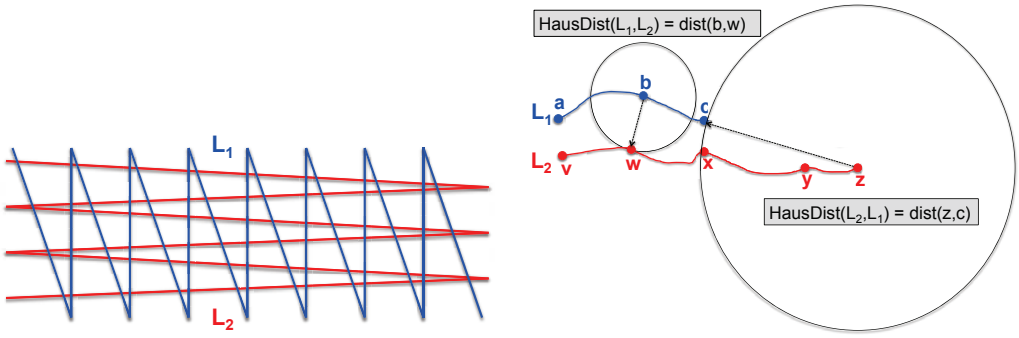
Figure 2.5: Hausdorff distance between two lines.

Based in these two equations, we use p_1 for representing a point in L_1 and p_2 as a point in L_2. Additionally, $dist(p_1, p_2)$ and $dist(p_2, p_1)$ are functions to determine the distance (e.g. Euclidian distance (Section 3.2.2)) of a point in a line with another point in the other line. Therefore, with Equations 2.1 and 2.2 we can compute the total HausDist according to Equation 2.3.

$$HausDist = \text{Max}\left(\underset{p_1 \in L_1}{\text{Max}}\left(\underset{p_2 \in L_2}{\text{Min}}\left(dist(p_1, p_2)\right)\right), \underset{p_2 \in L_2}{\text{Max}}\left(\underset{p_1 \in L_1}{\text{Min}}\left(dist(p_2, p_1)\right)\right)\right) \quad (2.3)$$

Although the Hausdorff distance is widely used in a significant number of algorithms, researchers found some problems involving the method to compute the distance between two linear objects. While the Hausdorff distance is a relevant measure in many applications, Figure 2.6 shows an example where it does not work very well.

Figure 2.6(a) shows the first problem. The authors of [54], [47] and [55] declare that the main reason for this problem is directly related to the course of curves. In other words, Hausdorff distance only considers the sets of points on both curves and does not indicates the course of the curves. The course of curves is an important feature in some applications, for example, in hand-



(a) Problem between two dissimilar lines with small Hausdorff distance.

(b) Problem involving the distance between two lines with different sizes.

Figure 2.6: Hausdorff limitations.

writing recognition.

Another problem involving the efficiency of Hausdorff distance was presented in [56]. It is associated with the maximum distance of a last point in comparison to another point when the lines have some differences (see Figure 2.6(b)). As we can observe, the final point of L_2 finds the final point of L_1, but they are very far.

The Hausdorff distance is the technique adopted in our approach, which covers the main requirements of our approach. However, there are other distances techniques widely used in terms of similarity analysis. These techniques are presented in the following sections.

3.1.2 Fréchet distance

Although the Hausdorff distance is a well-known example, one of the first methods to compute the distance between two lines was the Fréchet distance [57][58][59][60]. To better understand the Fréchet distance we present the traditional example of the dog and the man. Supposing that we have two lines (L_1) and (L_2). Now, we assume that the man walks from the first point to the end point of L_1, and the dog walks from the first point to the end point of L_2, with the man holding the dog by a dog's leash. The man and the dog try to walk closely and continuously (it is not allowed to go backward), each one with its own speed. Therefore, the Fréchet distance between L_1 and L_2 is the minimum leash length needed. By definition, if we have $dist(p_1, p_2)$ as an Euclidian distance between two points p_1 and p_2 in the plane, then we can

compute the Fréchet distance between L_1 and L_2 by

$$FrechetDist(L_1, L_2) = \underset{f:[0,1]\to L_1, g:[0,1]\to L_2}{\text{Min}} \left(\underset{t}{\text{Max}} \left(dist(f(t), g(t)) \right) \right), \quad (2.4)$$

where f and g are non-decreasing functions defining the positions of the man and the dog in each curve at every instant [61].

In general, Fréchet distance is divided in three groups, which are continuous Fréchet distance [62] [54], discrete Fréchet distance [63] and partial Fréchet distance [64].

Continuous Fréchet distance

The continuous Fréchet distance is a measure for matching two-dimensional polylines, but it is not a quite easy equation to compute. Assuming that E is a Euclidean plane and that $dist(p_1, p_2)$ is the Euclidean distance between two points ($p_1, p_2 \in E$). Besides that, the first line (L_1) is represented by the continuous function $f : [p_1, p'_1] \to E$ and the second line (L_2) is represented by the function $g : [p_2, p'_2] \to E$. Hence, the continuous Fréchet distance between two lines is

$$FrechetDist_{cont}(f, g) = \underset{\substack{\alpha:[0,1]\to[p_1,p'_1] \\ \beta:[0,1]\to[p_2,p'_2]}}{\text{Inf}} \left(\underset{t\in[0,1]}{\text{Max}} \left(dist(f(\alpha(t)), g(\beta(t))) \right) \right), \quad (2.5)$$

where $(p_1, p'_1, p_2, p'_2 \in \mathbb{R})$, $(p_1 < p'_1)$ and $(p_2 < p'_2)$. In [62], the authors presented the complexity to compute the continuous Fréchet distance between two lines in the Euclidian space, which is $O(s_1 s_2 log^2(s_1 s_2))$, where s_1 and s_2 represent the number of line segments in L_1 and L_2. Later, in [54], these same authors reduced the bound to $O(s_1 s_2 log(s_1 s_2))$.

Discrete Fréchet distance

Since a time complexity of $O(s_1 s_2 log(s_1 s_2))$ can be quite high in practice, in [63], the authors presented a simple dynamic algorithm to compute a discrete Fréchet distance in $O(s_1 s_2)$. We explain about this algorithm following the same example of the man with his dog, previously introduced in Section 3.1.2. In summary, the lines (L_1) and (L_2) are represented by a set of points:

$$L_1 = \{L_{1,1}, L_{1,2}, L_{1,3}, ..., L_{1,k}\}$$
$$L_2 = \{L_{2,1}, L_{2,2}, L_{2,3}, ..., L_{2,n}\}.$$

We consider the scenario in which the man and the dog walk from the first points ($man = (L_{1,1})$) and ($dog = (L_{2,1})$) along their respective lines, in direction to the final points ($man = (L_{1,k})$) and ($dog = (L_{2,n})$). With this in mind, we can obtain an ordered sequence of points, represented by $(L_{1,i}, L_{2,j})$. Therefore, we can intuitively extract three situations based on the evolution of the dog and the man in their respective lines:

- Case $i = j$, then both the man and the dog walk continuously together;
- Case $i > j$, then the man walks and the dog stays;
- Case $i < j$, then the dog walks and the man stays.

Each set of points also includes the end points of both two lines. Therefore, the discrete Fréchet distance (d_{Fd}) between L_1 and L_2 is recursively computed according to Equation 2.6 [65].

$$d_{Fd}(L_1, L_2) = Max \begin{pmatrix} d_E(L_{1,k}, L_{2,n}) \\ Min \begin{pmatrix} d_{Fd}(<L_{1,1}...L_{1,k-1}>, <L_{2,1}...L_{2,n}>) & \forall k>1 \\ d_{Fd}(<L_{1,1}...L_{1,k}>, <L_{2,1}...L_{2,n-1}>) & \forall n>1 \\ d_{Fd}(<L_{1,1}...L_{1,k-1}>, <L_{2,1}...L_{2,n-1}>) & \forall k>1, \forall n>1 \end{pmatrix} \end{pmatrix} \quad (2.6)$$

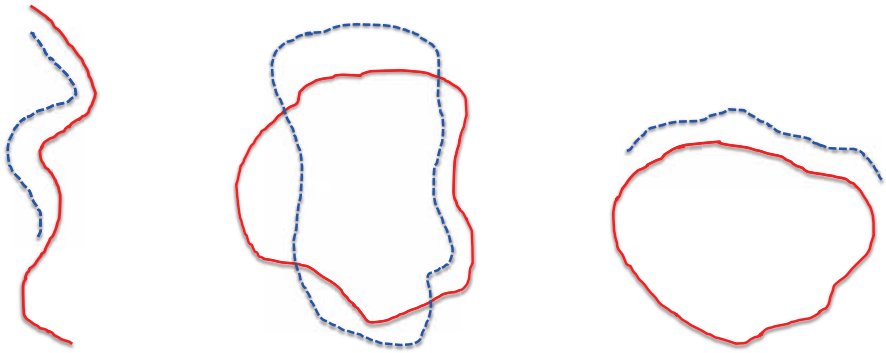
According to Equation, we have d_E to represent the Euclidian distance and $< L_{1,1}...L_{1,k-1} >$ and $< L_{2,1}...L_{2,n-1} >$ to symbolize the lines. Therefore, this equation allows to recursively use d_{Fd} algorithm with these parameters. This process is finished when the two lines are reduced to two points ($< L_{1,1} >, < L_{2,2} >$). d_{Fd} is a interesting estimation of Fréchet distance (d_F) due to the approximation that is limited by the maximum distance among two consecutive points of two lines (*FreMaxDist*) [63]. Consequently, we can represent the relation between d_F and d_{Fd} by:

$$d_F(L_1, L_2) \leq d_{Fd}(L_1, L_2) \leq d_F(L_1, L_2) + FreMaxDist.$$

In [65], we can find an example of discrete Freéchet distance between two lines, where the first line contains 8 vertices and the second contains 7 vertices. The authors generate two matrices (7x8). The first matrix presents the Euclidian distance in which each value of a cell is the distance between $L_{1,i}, L_{2,j}$. The second matrix contains the values from Fréchet distance. Based on the results obtained in these two matrices, they are able to obtain the discrete Fréchet distance between these two lines.

Partial Fréchet distance

The partial Fréchet distance is another interest method to compute the distance between two lines. It was derived from discrete Fréchet distance and introduced by [66]. The partial distance is very useful when a line L_1 does not match at some part with the other line L_2. Figure 2.7 presents three cases where partial Fréchet distance is applied.



(a) (b) (c)

Figure 2.7: Three examples for using partial Fréchet distance computation.

Firstly, Figure 2.7(a) illustrates a case where the use of partial Fréchet distance is needed. In summary, the algorithm detects the partial homologous line $< L_{2,begin}...L_{2,end} >$ and it is computed. The result of the partial Fréchet distance (d_{pF}) is equal to $d_{Fd}(L_1, < L_{2,begin}...L_{2,end} >)$.

In the second case (Figure 2.7(b)) is related to the distance between two polygon borderlines. The first step to compute the distance is to define a function T to represent polygon borderlines P_1 and P_2 into L_1 and L_2. For processing this step it is important to compute the minimum d_{Fd} between L_1 and L_2. This process also allows to inverse the ordering of points. Therefore, the discrete Fréchet distance between two polygon borderlines can be computed [65].

The third case that applies the partial Fréchet distance algorithm is presented in Figure 2.7(c). As we can observe, this example illustrates a line L_1 close to a part of the polygon borderlines P_2, which can be computed by the algorithm of partial discrete Freéchet distance (d_{pF}). The process to compute the distance is possible due to the combination between the two previous

steps.

Besides Hausdorff and Fréchet algorithms, it is possible to find many other approaches that use spatial information to perform similarity analysis between objects. In [67], the authors investigate techniques to analyze linear objects in a two or three-dimensional space. They focus on the extraction of some features from video clips, animal mobility experiments or mobile phone usage. Hence, assuming that such data are affected by a significant amount of noise, which degrades the performance of traditional metrics, they propose non-metric similarity functions based on the Longest Common Subsequence (LCSS). While these functions are robust to noise, they also provide an intuitive notion of similarity between linear objects by using a scheme of weight in order to identify the similar portions of the sequences.

Another interesting proposal was presented in [68], where the authors introduce a new distance function, called Edit Distance on Real sequence (EDR). Similarly to [67], they aim the designing of a robust method to solve the problems involving data imperfections and noise. Now, we address our attention to approaches based on temporal information to perform similarity analysis between two linear objects. These last two approaches have applied temporal functionalities that we must talk in the next section.

3.2 Similarity analysis based on temporal Information

The similarity analysis based on temporal information is an active research area, which has also been employed in a large number of commercial and academic applications. For example, time becomes an important element for supporting decision analysis, computer integrated manufacturing, computer aided design, geographic information systems, database management, and others. Making use of temporal analysis, an analyst is able to construct hypothetical situations to expect a future event [69] [70] as well as to define an interest of a single or a group of moving objects[71] [11]. Therefore, in this section, we describe four of the most used techniques of temporal analysis .

3.2.1 Dynamic Time Warping

Dynamic Time Warping (DPW) was presented in [72]. DPW provides an algorithm for measuring similarities between two objects that evolve in time

and recognizing the difference of evolution speed. Making use of this recognition, it finds an optimal alignment between two time-dependent sequences, adding some restrictions. Intuitively, we observe that the sequences have to be warped in a nonlinear way in order to match each other. In general, DTW has been applied in automatic speech recognition to compare speech patterns [73]. In fields such as information retrieval and data mining, this algorithm has been successfully used to automatically manage time deformations and different speeds of time-dependent data.

We suppose that we have two lines L_1 and L_2, of length k and n respectively, where

$$L_1 = l_{1,1}, l_{1,2}, l_{1,3}, ..., l_{1,k}$$
$$L_2 = l_{2,1}, l_{2,2}, l_{2,3}, ..., l_{2,n}.$$

In addition, it is defined N_{pair} as the number of pair of positions and $f(x)$ as the temporal function, which is used to align the temporal sequences. Taking into account these factors, the following definitions are determined:

$$f(x) = (f_1(x), f_2(x)) \text{ as a pair of positions to align, such as } x \in [1, ..., N_{pair}]$$
$$f_1(x) \in [1, ..., k] \text{ such as } f_1(x) \leq f_1(x+1) \leq f_1(x) + 1$$
$$f_2(x) \in [1, ..., n] \text{ such as } f_2(x) \leq f_2(x+1) \leq f_2(x) + 1,$$

where $f_1(x)$ and $f_2(x)$ are the representations of temporal indexes in L_1 and L_2, respectively. Another element is the weighting function ($w(k)$), which takes into account the correlation pattern between k and $k-1$. The increasing occurs in the values of f_1 or f_2 as well as in both f_1 and f_2 at the same moment.

The weighting function allows increasing the sequence alignment by denying or authorizing the duplication or deletion of elements in the sequence. Equation 2.7 shows the computation of dynamic time warping between two lines (L_1 and L_2) [51].

$$d_{DTW}(L_1, L_2) = min \left\{ \sum_{x=1}^{N_{pair}} dist(f_1(x), f_2(x)) w(x) \right. \tag{2.7}$$

A k-by-n matrix is generated to align the two sequences, where the (i^{th}, j^{th}) value of the matrix contains the distance dist($l_{1,i}$,$l_{2,j}$) between the two points $l_{1,i}$ and $l_{2,j}$ (e.g. Euclidian distance). Every matrix element (i,j) represents

the alignment between the points $l_{1,i}$ and $l_{2,j}$. Hence, the distance between each element of ($L_{1,i}$, $L_{2,j}$) is computed by Equation 2.8.

$$dist_{[i,j]} = d_E(L_{1,i}, L_{2,j}) \tag{2.8}$$

Having done that process, DTW algorithm completes the matrix by using dynamic programming. Now it is possible to minimize the total distance between the pairs of elements in checking each value in the matrix. Adding some procedure to define the checking path can optimize this process. The most important restriction is related to the path, which have to start in the position corresponding to the pair of elements that initiated the sequence of L_1 and L_2. The same limitation is applied to the final position of the checking path.

An example of the DTW algorithm is showed in Figure 2.8. We have two temporal sequences L_1 and L_2, where the pairs of positions are note aligned. They are illustrated with the same temporal factor in middle graph of Figure 2.8). Therefore, the DTW algorithm carry out the alignment in the two temporal sequences according to the minimal path illustrated with black points in the matrix of Figure 2.8. As we can observe in the last graph of Figure 2.8, the i^{th} position in the line L_1 is aligned with the $(i+2)^{th}$ position of the line L_2.

Other approaches have been introduced based on the DTW approach. In [74], the authors evaluate the efficiency of DTW algorithm in two sequences with different sizes. They propose the use of reorganization methods and techniques for increasing or decreasing the acceleration of local frequencies in the temporal sequence. However, these techniques can change the natural characteristics of each sequence.

Another improvement is associated with the use of a threshold to fix determine the maximum size of the checking path window. This method increases the performance since the checking path process does not compute all positions in the matrix [75]. However, identifying the optimal value to define the size of the window is not an easy task. Besides that, the problem involving sequences with different sizes is not considered in this case.

In summary, different methods to check sequences can be applied in combination with DTW algorithm. These methods can have symmetrical or asymmetrical characteristics and their own weights for the weighting function. A study showing some possible weighting functions is presented in [72].

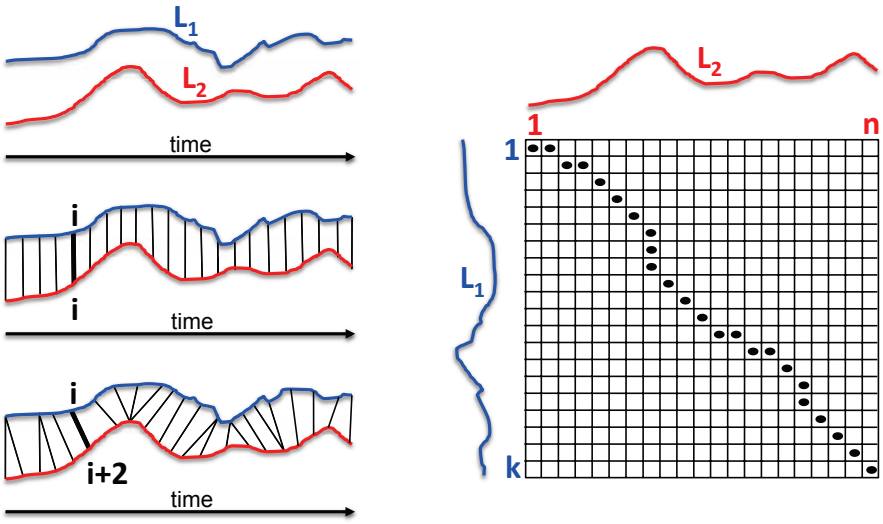


Figure 2.8: Example of dynamic time warp between two lines (L_1 and L_2).

3.2.2 Minkowski distance / ($L_p - norm$) distance

The Minkowski concept provides a parametric and adaptable distance function, which generalizes other distance functions used in the literature. The main advantage of this concept is related to its easy application in different cases of distance functions. Besides that, we may adapt the distance function by modifying Minkowski parameter in order to contemplate the requirements of each application.

The process to compute distances between elements can have different representations and measures. For example, the characteristics of an object can be represented by n variables that composes a vector H. As a result, we can create two vectors H_a and H_b with equal sizes, where $|H_a| = |H_b| = n$ and

$$H_a = [h_{a,1}, h_{a,2}, h_{a,3}, ..., h_{a,i}, ..., h_{a,n}]$$
$$H_b = [h_{b,1}, h_{b,2}, h_{b,3}, ..., h_{b,i}, ..., h_{b,n}].$$

We can use several distance-based functions for computing the difference between H_1 and H_2. Nevertheless, the function Minkowski ($L_p - norm$) is recommended when the user is interested to compute distance between two

vectors based on quantitative variables. Equation 2.9 presents this function.

$$d_{(Lp-norm)}(a,b) = \left(\sum_{i=1}^{n} |h_{a,i} - h_{b,i}|^p \right)^{\frac{1}{p}}, \text{ such as } p \geq 1. \tag{2.9}$$

Since we change the value of p to values greater or equal to 1, we obtain other well-known distance functions. For example, if we set the value of $p = 1$, we obtain the Manhattan distance function. Otherwise, when we change this value to $p = 2$, we have the Euclidian distance. We can verify these functions in Equations 2.10 and 2.11.

$$d_{(Manhattan)}(a,b) = \left(\sum_{i=1}^{n} |h_{a,i} - h_{b,i}| \right). \tag{2.10}$$

$$d_{(Euclidian)}(a,b) = \sqrt{\sum_{i=1}^{n} (h_{a,i} - h_{b,i})^2}. \tag{2.11}$$

Another interesting distance function is obtained by the change of p, called Chebychev distance. To obtain the Chebychev function we must to set $p \rightarrow +\infty$. This function is presented in Equation 2.12

$$d_{(Chebychev)}(a,b) = Max_i |h_{a,i}, h_{b,i}|. \tag{2.12}$$

While we observe that the Minkowski ($L_p - norm$) distance can be applied in a large number of scenarios and is adaptable to different types of scenarios, we also know that Euclidian ($p = 2$) and Manhattan ($p = 1$) are the most used function to compute distance between temporal sequences.

3.2.3 Edit distance

Formally introduced by Levenshtein in [76], the edit distance is frequently used in the comparison of characters. Briefly speaking, it provides an efficient way to compute the number of required operations to convert a set of characters in another set. The following equation shows how the edit distance (E_d) is

computed.

$$E_d(L_{1,1..i}, L_{2,1..j}) = \begin{cases} i & ,if(j=0) \\ j & ,if(i=0) \\ E_d(L_{1,1..i-1}, L_{2,1..j-1}) & ,if(i,j>0) \wedge (L_{1,i}, L_{2,j}) \\ 1 + Min \begin{cases} E_d(L_{1,1..i-1}, L_{2,1..j-1}) \\ E_d(L_{1,1..i-1}, L_{2,1..j}) \\ E_d(L_{1,1..i}, L_{2,1..j-1}) \end{cases} & ,otherwise. \end{cases} \quad (2.13)$$

The three types of operations to compute the distance of Levenshtein are insertion, substitution and deletion [77] [78]. The edit distance between two sequences $L_{1..k}$ and $L_{2..n}$ is computed by dynamic programming as presented in Equation 2.13. The complexity of this algorithm is $O(k * n)$. In [79], the authors modify the edit distance by removing the substitution function. The key idea is to extract similar sequences of different sizes by implementing an indexation method.

Another interesting approach based on the edit distance was quite introduced in the final of Section 3.1, which is named Edit Distance on Real sequence (EDR). Before presenting this approach, the same authors introduced the edit distance with real penalty (ERP) [80]. They propose to combine the edit distance with Manhattan distance algorithm in order to use penalties in dissimilar parts of the sequences.

3.2.4 Longest Common Subsequence

This approach is a variation of Edit distance (Section 3.2.3), which aims to describe how similar two sequences are one in comparison to the other. One of the main advantages of LCSS is its robustness involving problems of noise, by determining more weight to the similar parts of two sequences and giving less consideration to portions of dissimilarity [81][82].

We saw in the previous section that the edit distance is generally used in the comparison of characters. Consequently, LCSS distance function has to add a threshold ε due to the application of edit distance in a sequence of numbers. This threshold determines the value that considers if two values are close. Based on this threshold, the algorithm $(L_{a,i} \simeq L_{b,j})$ considers these values as equals or not. Therefore, the final result of this algorithm

is directly dependent on the value of this threshold. The LCSS distance is defined according to Equation 2.14 [67].

$$d_{LCSS}(L_{a,1..i}, L_{b,1..j}) = \begin{cases} 0 & \text{, if } (i = 0 \vee j = 0) \\ d_{LCSS}(L_{a,1..i-1}, L_{b,1..j-1}) & \text{, if } (L_{a,i} \simeq L_{b,j}) \\ max \left(\begin{cases} d_{LCSS}(L_{a,1..i-1}, L_{b,1..j}), \\ d_{LCSS}(L_{a,1..i}, L_{b,1..j-1}) \end{cases} \right) & \text{, if } (L_{a,i} \neq L_{b,j}) \end{cases} \quad (2.14)$$

Taking into account these methods of similarity analysis based on temporal information, the authors of [70] showed the efficiency of each approach in different situations. They compared the algorithms of Dynamic Time Warping (DTW), Minkowski distance ($L_p - norm$), edit distance with real penalty (ERP) and Longest Common Subsequence (LCSS). According to them, if small data sets are used, the DTW, LCSS, EDR and ERP will be more accurate than Euclidean distance and others ($L_p - norm$) distances. However, if the size of the training set increases, the accuracy of these algorithms will converge with the Euclidean distance for time series classification.

3.3 Spatio-Temporal Similarity Analysis between Trajectories

Considering all these spatial-based and temporal-based algorithms to identify geometric and temporal similarities between two linear objects, several works have been proposed to perform spatio-temporal similarity analysis between trajectories. We know that trajectories have spatial and temporal characteristics, which allows the designing of similarity analysis methods based on the study of the spatio-temporal distance between trajectories. As introduced in [11], the choice of the distance function is one of the most important steps to perform spatio-temporal similarity analysis between moving object trajectories. This choice extends to the classification of two different types of similarities: point-to-trajectory and trajectory-to-trajectory.

3.3.1 Point-to-Trajectory (P2T)

For this classification, we have the similarity analysis based on the proximity from a point p to a nearest point (q_{near}) of a trajectory (T). Hence, the

similarity factor is derived from the distance between p and T ($dist(p,T)$), as showed in Equation 2.15.

$$dist(p,T) = \min_{q_{near} \in T} d(p, q_{near}). \tag{2.15}$$

According to this equation, the distance function has to be defined in ($d(p, q_{near})$) (e.g. ($L_p - norm$) or others, as described in Sections 3.1 and 3.2).

Another interesting approach was introduced in [83], where the authors extended the concept of a single point to multiple points, by using a similarity function. In other words, this function takes into account the distance from the trajectory (T) and a set of points (P_{set}), specified by the analyst. Consequently, the analyst obtains the similarity (S) between the trajectory and each location required in P_{set}(see Equation 2.16).

$$S(P_{set}, T) = \sum_{p \in P_{set}} e^{dist(p,T)}. \tag{2.16}$$

With the use of the exponential function, it is possible to attribute some weights based on the nearest to the most distant points of the set in relation to the trajectory. Based on the final result of weights, we can determine what are the most similar trajectories in considering a required set of points.

3.3.2 Trajectory-to-Trajectory (T2T)

There are several ways to measure the similarity between two trajectories. As we showed in Sections 3.1 and 3.2, we can apply different types of spatial and temporal distance functions to obtain the information about similarities between two linear objects (trajectories). Along this line, we show some of these examples as follow.

In [53], the authors presented an incremental Hausdorff distance calculation algorithm, which is a new technique to compute the Hausdorff distance by using hierarchical indexes. Making use of an incremental method they compute the Hausdorff distance $HausDist(A, B)$ between two trajectories A and B by traversing the indexes of both trajectories. The indexes are created following the $R-tree$ method for each trajectory, such as R_A to trajectory A and R_B to trajectory B. They also use a priority queue (PQ) for controlling the order that the points P_A in R_A are verified.

A specific part of this approach was based on the algorithm proposed in [84]. This part was a modification performed in the function of upper bound

computation. For this new upper bound computation ($HausDistUB$), the authors take into account a subset (S_P) of points (P_B) within (R_B), where

$$min\,(HausDistUB(P_A, P_B) : P_B \in S_P)\,, \tag{2.17}$$

and S_P is controlled by a process that incrementally explores R_B by considering the distance from P_A.

The main contribution of this approach is presented in Algorithm 1, which efficiently explores both R_A and R_B by a loop. According to the algorithm, the decision to traverse R_A or R_B is done in each interaction.

Algorithm 1 Main algorithm (Inc-HausDist(A, B)) [53].

input: Point set A, Point set B
output: Hausdorff distance from A to B

RTree $R_A \leftarrow$ Create an R-Tree for A;
RTree $R_B \leftarrow$ Create an R-Tree for B;
MainPQ $MPQ \leftarrow$ Create a "descending order" priority queue (PQ);
SecondPQ $SPQ \leftarrow$ Create an "ascending order" priority queue (PQ);
Insert((RootOf(R_B), 0), SPQ);
Insert(((RootOf(R_A), ∞), SPQ), MPQ);
while MainPQ is **not** empty **do**
  MainPQ-Entry (P_A , UB, SPQ) $\leftarrow$ Dequeue (MPQ);
  SecondPQ-Entry (P_B , LB) $\leftarrow$ Head of SPQ;
  if P_A and P_B are both points **then**
    return UB;
  else if P_A is farther from the leaf level than P_B **then**
    TraverseA (P_A, SPQ, MPQ)
  else
    TraverseB (P_B , UB, SPQ, MPQ)

In summary, the algorithm starts by creating the R-trees as well as creating and initializing the main priority queue (MPQ) and the second priority queue (SPQ). In sequence, it inserts the root of R_B with initial distance of 0 into SPQ, the root of R_A with initial distance of ∞ and an SPQ into MPQ. After that, MPQ has a single entry containing the root of R_A as the associated point. For the while loop, the head entry (P_A, UB, SPQ) is dequeued from

MPQ. The same process is repeated in the head entry (P_B, LB) of SPQ, however, they are not dequeued. Note that the algorithm also implements a lower bound (LB) of ($HausDistLB(P_A, P_B)$), where P_B is the point of the $MainPQ$ *entry* to which the $SecondPQ$ *entry* is associated. Finally, based on P_A and P_B, a decision is made whether the execution has to terminate, to traverse R_A or to traverse R_B. These both traversing algorithms are also presented in [53]. Finally, the output of the algorithm will be the Hausdorff distance from A to B.

A second an interesting approach was presented in [85]. These authors proposed a similarity function based on Fréchet distance for performing efficient similarity join in large data sets of moving object trajectories. A new distance measure, called w-constrained discrete Fréchet distance (wDF), is presented as an adaptable extension to support lower/upper bounding, which enables the efficiency of a large number of trajectory similarity analysis. This approach adds a temporal parameter in the Fréchet distance function. Evaluating the performance of this warping window size (w), the authors concluded that the algorithm could be improved and introduced an optimization in [70]. The key idea for this optimization is to increase the performance when they have to compare similarities of long trajectories.

Another simple but efficient algorithm was presented in [86], where the authors applied the Euclidian distance to identify similarities between two trajectories. With the objective to improve this algorithm, the authors of [87] extended this work in considering the spatial and temporal features. They implement a window that searches the temporal similarities between two trajectories. The main problem related to these approaches is the fact that they consider trajectories with similar sizes and temporal features.

Taking into account the comparison of trajectories with different temporal features (e.g. different duration), the authors of [88] introduced the Temporal-Containment Similarity Distance (TCSD). They used a variation of the Fréchet distance, called Rigid Transformation Similarity Distance, to compute the temporal difference between each trajectory. This process is executed on each time that the shortest trajectory is increased along the longest trajectory.

These approaches are only some of the many examples that execute spatio-temporal similarity analysis between trajectories. We can find several other approaches involving this topic. However, in this chapter, we discuss only

the most important approaches in the context with this thesis, such as the Hausdorff distance algorithm.

4 Conclusion

The data representation of moving objects and their movements is the first requirement to understand about trajectory data. With this in mind we can define a trajectory representation for providing an easy way to organize trajectory data, to manipulate structured query languages, to specify profiles through movements, to create and compare profile groups, and others. Trajectory representation is important to consider the diversity of semantic data that enriches the knowledge on moving object trajectories. Making use of these data representation, the analysis of movement data can be intuitively explained.

In this chapter we introduced a conceptual view on trajectories, starting by the representation of moving objects, movements and trajectories. After that, we showed how the analysts and scientists use these representations to identify similarities between trajectories. Finally, we presented a review of the state of the art involving similarity analysis in different domains. Besides that, we presented an overview of the main challenges related to frequent problems in analyzing dissimilar trajectories. These approaches help to understand the different techniques to analyze the similarities between moving objects, bringing important aspects to the designing of our approach.

Based on the proposals and results presented in these works, we note that some approaches can be extended to other research areas, such as Location-Based Social Networks (LBSN). Social features can be added to spatio-temporal distances in order to strengthen the similarity between moving objects. Therefore, we conclude that these distance functions have a relevant importance when combined with other types of data in order to increase the accuracy in the identification of similarities between moving object trajectories.

Chapter III

Spatio-temporal clustering and patterns of trajectories

Contents

Nowadays, we observe that the recording of Global Positioning System (GPS) tracks generates a large amount of trajectory data. This data holds spatio-temporal information about moving objects (such as pedestrians, cars, buses, etc.). In order to analyze such data there exists several exploratory as well as clustering methods. Clustering and aggregation (data mining) methods

have generally been adopted to explore and analyze movement data when exploratory (e.g., visualization) methods are not enough to explore large spatio-temporal datasets.

In this chapter we focus on the context of clustering methods as a promising solution to identify trajectory patterns of individual and a group of moving objects. As introduced in the previous chapter, a moving object can be traced along the time, generating trajectories that represent their movements. Along this line, we address our attention on spatio-temporal clustering methods to find trajectory patterns of moving objects in geographic spaces. We present a review of the state-of-the-art of existing spatio-temporal clustering approaches, by showing the application of these algorithms in different scenarios.

Since the real trajectory of a moving object is uncertain, an estimation of the best representative trajectory can be done based on the location records. The existing clustering techniques bring important elements to reason about moving objects and their trajectories, which make possible to discover an approximation of the real trajectory according to spatio-temporal patterns (e.g., trajectory patterns). Hence, we finish this chapter showing the different types of spatio-temporal patterns in which assist in the discovery of interests of a single or a group of moving objects.

1 Spatio-temporal clustering methods

Clustering algorithms are an important element to describe trajectory data model from a large amount of data, allowing the analyst to focus on a higher representation level of spatio-temporal data. The different types of clustering methods provide facilities to explore large datasets and associate moving objects characteristics in clusters (or group of common features). Based on these clusters, the analysts can describe a moving object or a group of them according to its or their characteristics.

In the context of spatio-temporal trajectories, the clustering methods could be divided into four main groups (density-based, distance based, visual analytics and hybrid) [89] [90], which are presented as follow.

1.1 Density-based methods

Density-based clustering methods explore the density of points in a given space to identify clusters. Formally, these methods define a radius (ε) around each object in order to identify the minimum number of objects (neighbors) has to be included in each cluster. Several works have showed that these methods are efficient to generate arbitrary shapes of clusters and are robust to avoid problems such as noise and outliers [91] [92]. This is an important advantage for a large number of applications, such as data sources that have underlying parts (e.g., data acquired by observing user behaviors in urban centers), or data acquired by not-reliable resources/systems (e.g., low-resolution sensors) [93]. Since trajectory data often suffer of both the indicated problems, then noise tolerance becomes an important advantage.

1.1.1 DBSCAN

In [91], the authors introduced the DBSCAN, a density-based algorithm for discovering clusters in large spatial databases with noise. To explain the DBSCAN algorithm, we assume two parameters, (min_{obj}) as the minimum number of objects and (ε) as the maximum radius around each object. Making use of these parameters, the algorithm finds the core objects and the neighborhood of each core object. An object is a core when its number of neighbors is greater than or equal to min_{obj}. Otherwise, it is a density-reachable from another object (if its number of neighbors if less than min_{obj}) or a noise (when it does not have a neighbor).

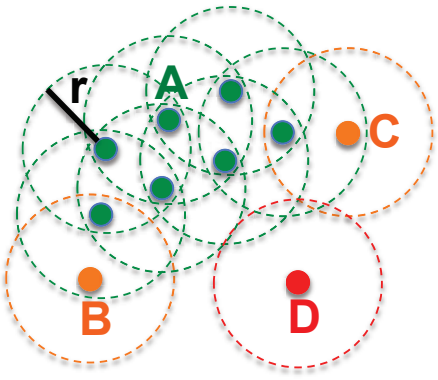


Figure 3.1: Difference between objects according to DBSCAN algorithm.

Figure 3.1 illustrates an example of different types of objects for the values of $min_{obj} = 3$ and $\varepsilon = r$. As we can observe, A is a core object, because it reaches 4 neighbors. Besides the node A, the other green objects are core objects, since they reach 3 or 4 neighbors. B and C (orange color) are *density reachable* objects of the cluster because they reach less than 3 neighbors. Finally, the object D is a noise due to the absence of neighbors. Two years later, this algorithm was improved by the Generalized DBSCAN (GDBSCAN), which implements non-spatial parameters [92]. Based on these two important approaches, several proposals were designed as variants of DBSCAN algorithm.

1.1.2 OPTICS

In [1], the authors introduced a well-known evolution of the basic DBSCAN algorithm, called OPTICS (Ordering Points To Identify the Clustering Structure). The OPTICS algorithm produces an ordering of a dataset while storing the core distance and a suitable reachability distance of each user trajectory. OPTICS provides information about the overall clustering structure unlike other method that computes a flat partitioning of data (such as K-means [94]). Furthermore, OPTICS provides an intuitive data-independent visualization of the cluster structure by generating a reachability plot, illustrated on the right side of Figure 3.2. The reachability plot brings valuable information to better understand the dataset, assigning each object to its corresponding cluster or noise. A brief overview of OPTICS is presented with the help of underlying terminologies.

Given o_a as an object from a dataset D, ε as the distance threshold, $N\varepsilon(o_a)$ as the ε-neighborhood of object o_a, $min_{(neig)}$ as a natural number to define the minimum number of neighbors, and $min_{(neig)}\text{-}distance(o_a)$ as the distance from o_a to its nearest neighbor $min_{(neig)}$. Thus, the core distance (C_{dist}) is defined as:

$$C_{dist} = \begin{cases} Undefined, & \text{if } N\varepsilon(o_a) < min_{(neig)} \\ min_{(neig)}\text{-}distance(o_a), & \text{otherwise.} \end{cases} \tag{3.1}$$

Based on that, the C_{dist} is the smallest distance ε between o_a and another object in its ε-neighborhood such that o_a would be a core object. Otherwise, the C_{dist} is $Undefined$.

Making use of the core distance, the reachability distance can be computed. We assume that o_a and o_b are objects from a dataset D, $N\varepsilon(o_b)$ is ε-neighborhood of object o_b, and $min_{(neig)}$ is a natural number to define the minimum number of neighbors. Then, the reachability distance (R_{dist}) of o_a with respect to o_b is defined as:

$$R_{dist} = \begin{cases} Undefined, & \text{if } N\varepsilon(o_b) < min_{(neig)} \\ max\left(C_{dist}(o_b), distance(o_b, o_a)\right), & \text{otherwise.} \end{cases} \quad (3.2)$$

We observe that the reachability distance ($R_{(dist)}$) of o_a is the smallest distance such that o_a is directly density-reachable from a core object o_b. Otherwise, if o_b is not a core object, even at the generating distance ε, the reachability distance of o_a with respect to o_b is *Undefined*.

Finally, the OPTICS produces a reachability plot that shows the cluster ordering and the reachability values. The reachability plot gives a graphical view of the structure of the data by providing data independent visualization. From the output plot, clustering can be obtained by choosing an appropriate threshold value of reachability distances. There are automatic techniques available to identify clusters from this plot, which is applicable when the dataset is very large. Figure 3.2 illustrates cluster ordering with the help of a reachability plot showing valleys to identify potential clusters.

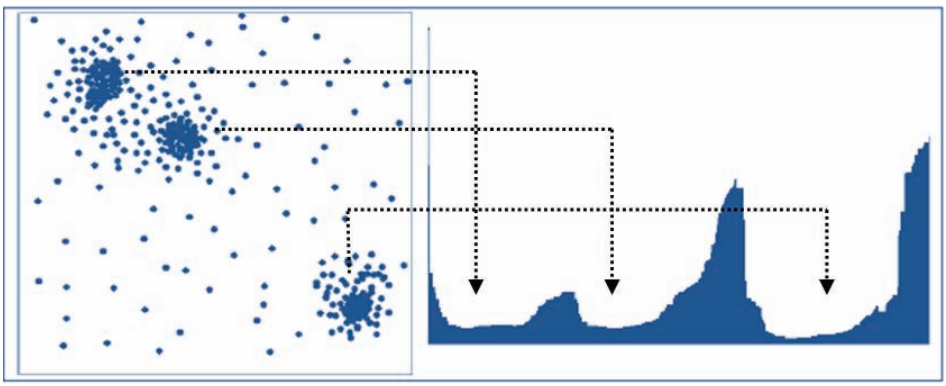


Figure 3.2: A reachability plot showing data densities and respective clusters [1]

It is important to mention that two parameters are of significant importance in OPTICS algorithm (maximum distance threshold and minimum num-

ber of neighbors). As Ankerst et al.[1] suggest the distance threshold influences the number of clustering levels, which can be seen in a reachability plot. The smaller the distance, the more objects may have undefined reachability distances. Therefore, the clusters with lower density might be less visible and hence this situation should be prevented. Similarly, the larger minimum neighbor value will yield better results.

We clearly see that the density-based methods are directly related to an efficient definition of the neighborhood parameter. Hence, it is necessary to a well-represented index data structure to in order to improve the performances of such algorithms. The density-based algorithms can be applied in different scenarios. However, when they are used in the context of spatio-temporal data, it is necessary to design an adaptable and robust algorithm for supporting the rich and complex characteristics of spatio-temporal data [95] [96].

1.2 Distance-based clustering methods

Distance-based methods are usually designed to increase the accuracy in the similarity analysis between moving object trajectories. It brings a combination of similarity distance functions (presented in Chapter II) with clustering techniques. Consequently, the distance-based methods are normally executed in two main steps. Firstly, the algorithm computes the distance between objects based on a defined distance function. After that, it applies a clustering technique to generate the clusters of objects.

Traditionally, the clustering methods execute their algorithms to find clusters in the whole trajectories, which can result in loss of essential similarity information. For example, a behavior of trajectories can be similar in the beginning, but the directions can change over time, as presented in Figure 3.3. Observing the 3 trajectories, we can clearly see that they are similar in the initial period of time, however, one of them changes the direction later.

This combination of distance functions and clustering algorithms becomes necessary, since the similarities between spatio-temporal trajectories are strongly associated with their applications (as it was presented in the previous chapter). For example, if two moving objects follow the same trajectory in the same time interval, they can be described as similar (e.g., supposing that they have visited similar places at the same time instants). Therefore, we intuitively expect the characteristics of similarity according to these spatio-temporal measures,

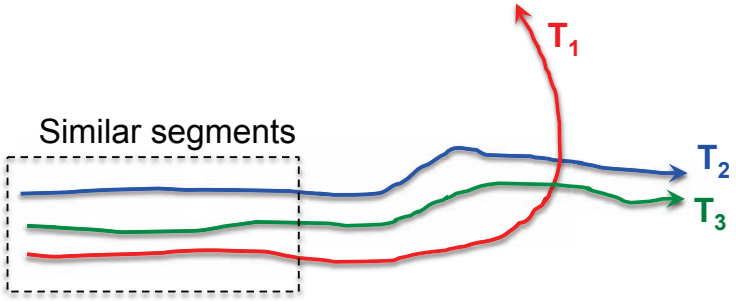


Figure 3.3: Example of trajectories partially similar.

such as the granularity of the movements (e.g., the number of spatio-temporal points for each trajectory) and the uncertainty on the measured points [90].

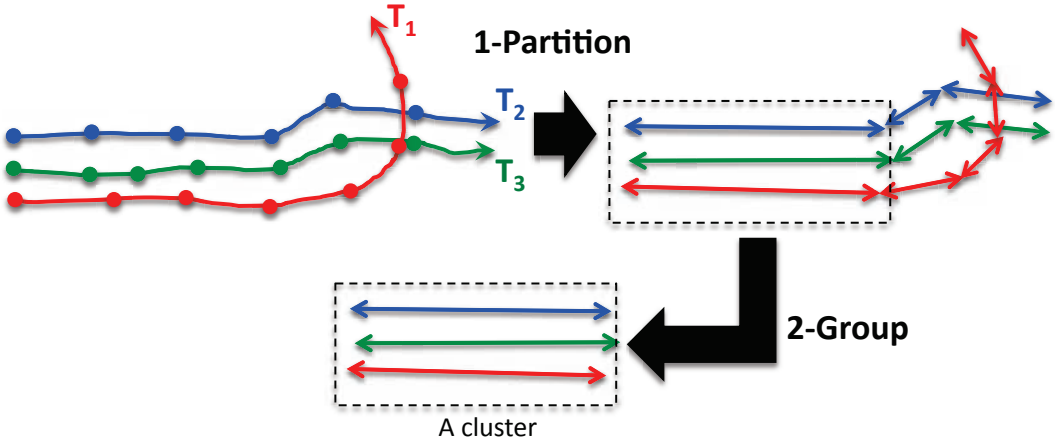


Figure 3.4: Example of the algorithm of partition-and-group.

In [97], the authors introduced the partition-and-group Framework, which the key idea is to perform the partition of a trajectory into a set of line segments at characteristic points in order to group similar line segments in a dense region. As a part of this Framework, they implemented a clustering algorithm, called TRACLUS. In summary, the algorithm executes two main processes: partitioning process and grouping process. In the partitioning process, they use Euclidian distance function and the minimum description length (MDL). The Euclidian distance function is used to support the computing of three main components: perpendicular distance, parallel distance and perpendicular distance. These three components discover common sub-trajectories. In addition, the algorithm applies the MDL as a method to find the optimal

tradeoff between preciseness and conciseness. The step 1 of Figure 3.4 shows an example of trajectory partitioning.

In the grouping process (second step of Figure 3.4), the clusters are identified by a density-based clustering method, which is based on DBSCAN algorithm. Finally, the authors show that their Framework is efficient to discover the representative trajectories from a trajectory database.

Another interesting distance-based clustering algorithm was presented in [98]. The author uses a distance parameter that describes the similarity of object trajectories over time, which is computed by analyzing the distance variation between the objects. He considers only pairs of objects with temporal resemblance (e.g., it compares the positions of objects at a certain time instant, grouping the set of distance values). Therefore, the distance between trajectories two trajectories A and B ($Dist(A, B)$) is based on the average distance between objects, as presented in the equation as follows.

$$Dist(A, B)|T = \frac{\int_T d(A(t), B(t))dt}{|T|} \tag{3.3}$$

According to this equation, $d()$ computes the Euclidean distance in a space, T represents the time interval in which both trajectories A and B are found, and $A(t)$ and $B(t)$ determine the positions of the objects in their respective trajectories at a certain time instant t. This is the generic equation proposed by the author. Additionally, he suggests the distance computation by means of Euclidian distances due to the piece-wise linearity of trajectories.

Therefore, we conclude that distance-based clustering methods are strongly used in supporting the similarity analysis of trajectory data of moving objects. According to [99], these methods bring significant contributions to the approaches that perform time-series analysis [46] and can be easily adapted to Longest Common Sub Sequence (LCSS) algorithm [67], as presented in Chapter II.

1.3 Visual analytics methods

Automatic clustering methods for trajectory data are able to recognize behavioral similarities taking into account an optimized function and/or algorithm, however, they can led to similarity errors from the point of view of the real representation of a movement behavior or phenomenon. Visual-aided methods focus on the human expert's interactions to achieve the desired clustering

result. This method allows the expert to judge if the results are satisfactory according to the application and/or analyst requirements.

Taking into account this necessity to overcome the problems of adverse or incomplete results of automatic methods, the authors of [100] have proposed visual analytics clustering methods. They introduced some frameworks, which provide several visualization procedures to analyze spatio-temporal data. Besides that, they proposed different manners for analyzing trajectory data, such as clustering, aggregation and generalization [2].

A pertinent visual analytics model introduced by these authors in [101] combines classification and clustering algorithms to extract satisfactory clusters from large databases, which are managed by a human analyst through an interactive visual interface. Given a dataset (D) and an object (o), the whole process introduced by these authors is described in the following steps:

1. Extract a subset D' of the objects from D, by preserving their actual distribution in D;

2. Execute OPTICS algorithm (Section 1.1.2) with an acceptable distance function d to obtain a set of clusters $\{C_1, C_2, ..., C_m\}$;

3. For each cluster (C_i), do

   - Select x prototypes in C_i, where $1 \leq x < |C_i|$; generate $\{p_i^1, p_i^2, ..., p_i^x\}$, with corresponding distance thresholds $\{\varepsilon_i^1, \varepsilon_i^2, ..., \varepsilon_i^x\}$, such that the cluster C_i could be described as a set of objects in D' as well as the distance to one of the prototypes p_i^j is less than the corresponding threshold ε_i^j (e.g., $C_i = \{o \in D' | \exists j,\ 1 \leq j \leq x$, such that $d(o, p_i^j) < \varepsilon_i^j\}$.
   - Then, the classifier is generated by the set of prototypes (for all clusters p_i^j), their distance thresholds εi^j and the function d.

4. Now, the analyst visually inspect and refine the classifier (modify the clusters, if necessary).

5. Apply the classifier to the remaining objects in D, for each $o \in D$, $o \notin D'$

   - Find every close prototype (e.g., p_i^j, $1 \leq i \leq m$, such that $d(o, p_i^j) < \varepsilon_i^j$). On the one hand, if there exist only one close prototype p_i^j, then attach o to the cluster C_i represented by p_i^j. On the other hand, if

there are two or more close prototypes $p_{i1}^{j1}, ..., p_{iN}^{jN}$, then select the closest of them (e.g., such prototype p_{ik}^{jk}, that $d(o, p_{ik}^{jk}) < d(o, p_{in}^{jn})$ for $\forall n : 1 \leq n \leq N,\ n \neq k$), then attach o to the cluster C_i^k represented by p_{ik}^{jk}. If there is no close prototype, then the object remains unclassified.

6. If necessary, eliminate the original and new members of clusters $\{C_1, C_2, ..., C_m\}$ from D and restart the whole process again.

These authors gave a demonstration of the approach in the analysis of moving object trajectories. After evaluating the results, they affirm that the approach can be used in different types of objects [101]. These same authors presented a method for spatial generalization and aggregation of movement data, which transforms trajectories into aggregate flows between areas, as presented in [2]. Figure 3.5 shows an example of this recent method in an enlarged fragment of the map of Milan.

In [102], the authors also introduced a visual-interactive framework, which provides procedures to guide the analyst in the execution of the Self-Organizing Map (SOM) algorithm [103]. This framework allows the user to visually inspect the clustering process and manage the algorithm at a determined level of detail. Besides that, it offers some additional interaction facilities, fore example: a function to initialize the clustering algorithm to edit the trajectories or part of them; and techniques to manipulate the training parameters during runtime.

An approach of progressive clustering techniques was proposed in [104]. Formally, they propose a progressive clustering algorithm to analyze the movement behavior of objects. Making use this algorithm, the analyst continuously modifies the distance function according to the spatial, numerical, temporal or categorical variables on the spatio-temporal data. This process brings better understanding information of the underlying data. The main contributions of this approach are related to the use of different distance functions and the combination of data mining techniques and machine learning algorithms in order to optimize the trajectory data visualization.

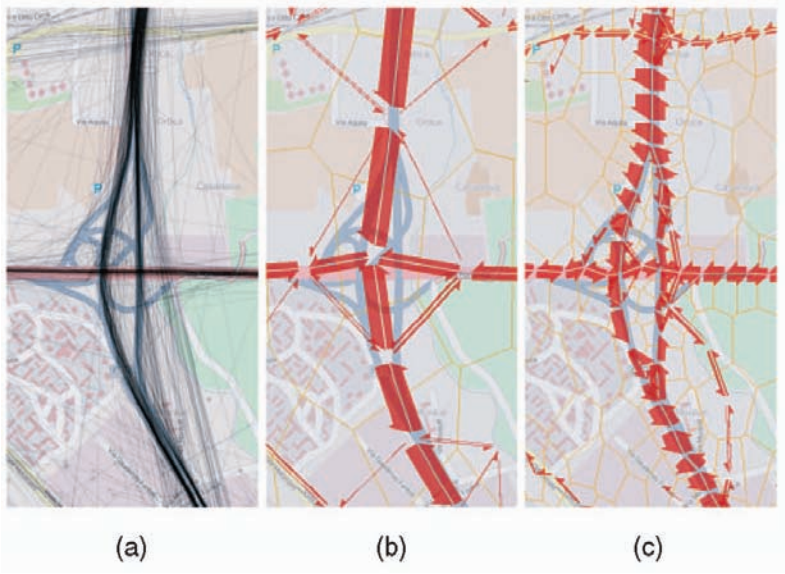


Figure 3.5: Example of the visualization tool presented in [2]. In (a), we visualize the trajectories in black color, with 10 percent opacity. In (b) and (c), we note the generalized representations of the trajectories, which are generated according to specific parameters.

1.4 Hybrid methods

A considerable number of other clustering methods have been developed in the context of spatio-temporal data, which deal with the specific application requirements of trajectory similarities. These other methods include model-based clustering algorithms, micro and macro clustering, and flocks and convoy. Since this thesis focus on the use of OPTICS, distance-based techniques and some visualization procedures (as presented before), we show only a brief overview about these other clustering methods in the following sections.

1.4.1 Model-based clustering

Model-based clustering techniques focus on the description of the whole dataset taking into account a generative data model. Consequently, the model type is often specified *a priori*, such as Gaussian or Markov models [105]. The model structure (e.g., the number of states in a Markov model) can be determined by selecting a model from a set of candidates and the parameters can be calculated by using maximum likelihood algorithms (e.g., the Expectation-

Maximization (EM) algorithm [106]. Therefore, the appropriate model is selected based on the application requirements.

Making a comparison between model and applications, it is possible to point out some examples. Multinomial models have achieved good results for text clustering approaches [107] [108]. Gaussian mixture models are frequently used in the analysis of vector data [109] [110] [111]. Other works have presented that model-based approaches that focus on the mixture models can obtain interesting results [112] [113]. Finally, Markov chains and hidden Markov chains have been widely adopted when complex data are analyzed (e.g., time sequences) [114] [115] [116].

In the context of trajectory data, two important approaches can be presented. The first one was introduced in [117], where the clustering algorithm determines groups of objects based on means of the EM algorithm. Additionally, they consider the variation of spatio-temporal information of trajectory data for each cluster. The second approach implements a model-based algorithm for computing means of a Markov model in order to find the transitions between successive positions [118].

1.4.2 Flock and Convoy

Convoy and Flocks methods are generally applied in scenarios where the analyst is interested to identify groups of moving object trajectories that evolves together during a specific time interval. For instance, a vehicle convoy or flock of birds. Although the definition is very close, these two methods have some differences. One the one hand, a flock can be defined as a subset of moving objects that evolves together along paths for a certain pre-defined time within a circular disk [119]. On the other hand, convoy method is strongly based on density constraints, such as the number of objects, the distance between objects and the lifetime. For example, a query can be "*find all groups of trucks that traveled closely together for longer than 30 minutes*". With convoy method, it is possible to receive the expected result, which it is not true by using a flock method [120].

In comparison with convoy, the flock method specifies a value to determine the size of the circular disk. Since the convoy does not specify query inputs (e.g., a given trajectory or a limited window), flock method is not able to form the flock when the group of objects is located over a wider area than the disk

size. Therefore, in the convoy method, an input can be every position in the dataset and a target can be every object [120]. Some examples of flock and convoy are presented as follow.

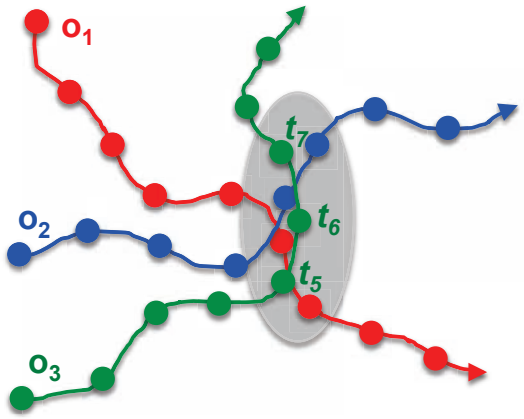


Figure 3.6: Example of a flock.

In [119], the authors presented a new algorithm to discover flocks for analyzing them theoretically. They also introduced the initial concepts to understand flock methods. Figure 3.6 illustrates an example of a flock. Given three objects o_1, o_2 and o_3, it is possible to identify a flock between their three paths in the time instants t_5, t_6 and t_7 of the object $o3$. Along this line, they achieved good results by adding a tree-based algorithm, especially when a small number of temporal information is used. However, they have a strong dependence on the features of the input parameters.

In the context of convoy methods, the authors of [120] formalized the concept of a convoy query by using density-based notions, in order to capture groups of arbitrary extents and shapes. Figure 3.7 illustrates an example of a convoy.

Given a set of trajectories (S_{traj}), the number of objects (n), a distance value (ε) and a lifetime (k), then a convoy is composed by a group that has at least n objects. These objects have to be density connected based on the ε distance during k consecutive time instants. A convoy occurs between the groups of objects that travel together in the same time interval. Taking into account the 3 objects (o_1, o_2 and o_3) during the time interval between t_1 and t_4 in Figure 3.7, we specify the parameters $n = 2$ and $k = 3$. Hence, the objects o_1 and o_2 form a convoy during the consecutive time instants (from t_1

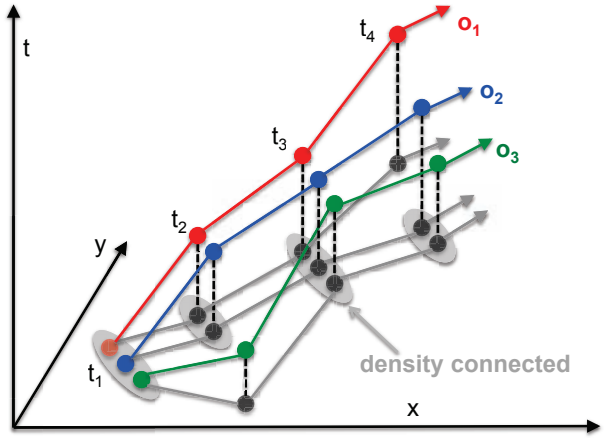


Figure 3.7: Example of a convoy.

to t_3).

1.4.3 Micro and macro clustering

Micro clustering algorithms have been applied to store tight clusters of similar segments of the trajectory. The main advantage of these algorithms is the facility to update the dataset due to their small sizes. This characteristic makes these algorithms suitable for applications that need incremental clustering features [121]. Macro clustering algorithms take the set of micro clusters to discover macro-clusters.

An interesting approach based on micro clustering is presented in [122]. In this work, the algorithm divides the segments of different trajectories into rectangles. After that, it groups only the segments within the rectangle that occur at similar time intervals. The key idea is to discover the maximal size of the cluster and temporal dimension, considering the rectangle as a threshold.

Similar to this work, the authors of [123] also proposed an algorithm that represents a trajectory by segments. The approach uses a process to determine near time intervals (e.g., a maximal time interval where the segments of trajectories are near). Therefore, the cluster is discovered according to the total time of near segments.

The micro and macro clustering methods have some similarities in comparison to the other methods previously introduced. The main difference is related to the creation of micro cluster for representing the smallest possible

size of a cluster. It is important to emphasize that micro-clustering algorithms can achieve good performance results when applied in dynamic datasets, where the data is frequently updated [124] [121].

In summary, we note that all these clustering approaches presented in this section can be directly applied in trajectories due to their robustness and adaptability in considering the existence of several spatio-temporal characteristics of a moving object trajectory, such as speed, direction, stops, acceleration, and others. Hence, we can use the obtained results to identify spatio-temporal patterns related to these data. We talk about spatio-temporal patterns in the next section.

2 Spatio-Temporal patterns

Since the real trajectory of a moving object is uncertain, an estimation of the best representative trajectory can be done based on the location records. The existing techniques presented in Chapter II and in Section 1 bring important elements to reason about moving objects and theirs trajectories, which make possible to discover an approximation of the real trajectory according to spatio-temporal patterns (e.g., trajectory patterns).

The concept of trajectory patterns is associated with different types of patterns, which can be obtained from trajectory data [125]. Therefore, the first step to detect a pattern is to understand what types of pattern features may exist in the trajectory data. Taking into account the review of the related work in literature, the authors of [30] try to maximize the use of already existing accurate definitions about patters as well as to minimize redundant and conflicting terminologies. Consequently, they defined pattern concepts to provide a comprehensible classification of trajectory data.

A good manner to discover a trajectory pattern is by understanding its classification according to the number of moving objects that is considered in the pattern (individual vs. group pattern) and the characteristics of patterns (Generic vs. behavioral patterns and primitive vs. compound patterns). We start presenting the differences between spatial and spatio-temporal patterns. After that, we address our discussion to the classification of spatio-temporal patterns.

2.1 Spatial vs. spatio-temporal patterns

One of the most important steps to discover a pattern is to comprehend the differences between spatial and spatio-temporal patterns. For example, multiple trajectories can have spatial similarities but temporal dissimilarities between them. Figure 3.8 illustrates an example of three trajectories that evolve over time (from the instant $t = 1$ to the instant $t = 4$.

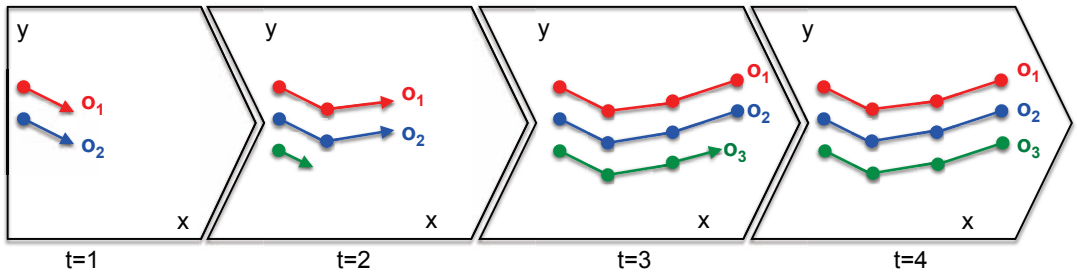


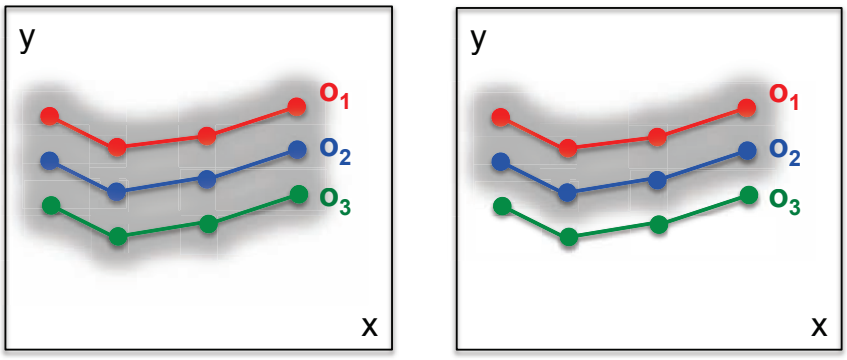
Figure 3.8: Evolution of three moving object trajectories from $t = 1$ to $t = 4$.

Following the example of Figure 3.8, we see that the moving objects o_1 and o_2 (red and blue, respectively) start their movements at the time instant $t = 1$. At the time instant $t = 2$, they continue their trajectories and the moving object o_3 (green color) initiates its movement. At $t = 3$, the moving objects o_1 and o_2 arrive at the destination point and o_3 continues its movement. Finally, the moving object o_3 arrives at the same destination point and at the time instant $t = 4$.

While we observe that the three trajectories recorded by o_1, o_2 and o_3 are geometrically similar in the space, we also note that the moving object o_3 arrived close to destination point of o_1 and o_2 at the time instant $t = 4$. Therefore, if we consider only spatial pattern for this example, we will discover a pattern for the three trajectories at the time instant $t = 4$. Figure 3.9(a) shows the spatial patterns that was discovered at the time instant $t = 4$.

However, Figure 3.9(b) shows that the trajectory of the moving object o_3 is not discovered as a spatio-temporal pattern. In spite of the three trajectories are geometrically similar, the construction of trajectory 3 (green trajectory) was later finalized. Hence, in contrast to spatial patterns, a spatio-temporal pattern always takes into account both spatial and temporal information.

For example, spatial patterns can be interesting to obtain touristic infor-



(a) Spatial pattern. (b) Spatio-temporal pattern.

Figure 3.9: Difference between spatial and spatio-temporal patterns.

mation about a specific city, based on the trajectories recorded by tourists that have visited that place. Only the spatial similarities are enough to find a trajectory pattern and recommend some sequences of visits to the tourist. For spatio-temporal patterns, we can enrich this example by adding the temporal information. Consequently, we can use temporal information to inform the instants that a group of tourist has visited some places.

Besides that, we can present other examples with spatio-temporal patterns, such as: identifying users that travel by common streets in a urban center to go from home to work everyday in order to provide car pooling recommendations; finding places in road networks where urban traffic jams could occur based on histories of vehicle trajectories; and others [125].

2.2 Classification of trajectory patterns

Since we know the methods and techniques to compare similarities between spatio-temporal data, we are able to understand the differences between spatial and spatio-temporal patterns. Along this line, the next step is to recognize the classification of trajectory patterns. We have followed the classification introduced in [30], which the concepts are pertinent to the geo-spatial domain and may be applicable for all the common types of moving objects, such as cars, animals, humans and eye movement data.

2.3 Individual vs. group patterns

The classification of moving object trajectories into individual or group is important to identify the type o spatio-temporal pattern that is being extracted. For example, assuming that we have one person as a moving object, then the individual pattern can be represented by his/her daily routines to go from home to work everyday. For the patterns of group, it is considered the spatio-temporal similarities of a group of moving objects. For instance, the common road segments that a group of employers takes to go from their homes to the company everyday.

The identification of trajectories with spatio-temporal patterns has to follow a consistent notion of similarities [126]. In general, queries are performed to find similarities between the same type of moving object, for example: cars to provide carpooling services; trucks to increase the logistic of deliveries. Since we try to discover a pattern between dissimilar moving objects, others features have to be considered, such as different speed, constrained paths, and other characteristics that can affect the pattern discovering.

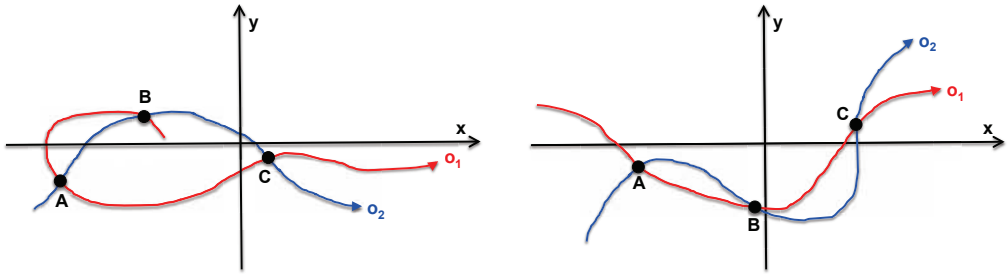
Therefore, the type of moving objects and their relations are directly associated with the interpretation of spatio-temporal patterns, which can be classified into individual and group patterns. Besides this classification, it is important to recognize the generic or behavioral pattern of a single of multiple moving objects.

2.4 Generic patterns vs. behavioral patterns

In generic spatio-temporal patterns, the relations are directly related to the trajectory data and are usually insufficient to describe a specific behavior of a moving object. In [30], the authors divided a generic pattern according to its association with primitive or compound patterns. According to them, primitive patterns describe the most basic types of trajectory patterns, where the similarities are associated with single parameters (e.g., the same type of moving object). Otherwise, compound patterns are built from several primitives' patterns, which consider a combination of relations between moving objects.

The generic patterns related to primitive patterns are dimensioned into 10 groups, which are: co-location in space [36], concentration [30], incidents [127], constancy [128], sequence [129] [130], periodicity [129] [36], meet [131] [132], moving cluster [128], temporal relations [129], and synchronization in

time [36]. Figures 3.10 shows two examples of generic patterns related to a co-location in space as a primitive pattern.



(a) Unordered co-location. (b) Ordered co-location.

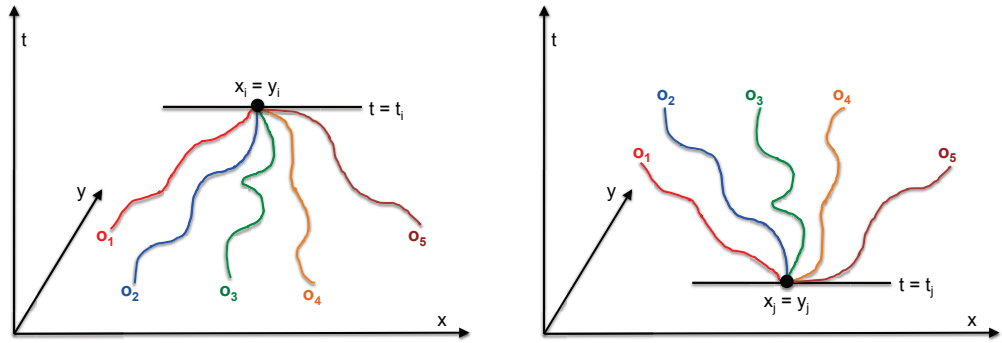
Figure 3.10: Examples of co-locations in space.

As we can see in Figure 3.10, a co-location happens when two moving objects have common positions in their trajectories [36] [30]. Besides that, it is possible to divide this co-location pattern in two types, unordered co-location (see Figure 3.10(a)) and ordered co-location (see Figure 3.10(b)). The basic difference is related to the sequence of occurrences A, B, and C in comparison to the natural sequence of points of each trajectory. In other words, in Figure 3.10(a) we have an unordered co-location because the common point B occurs before the common point A in the trajectory of the moving object o_1. Otherwise, Figure 3.10(b) shows an ordered co-location, where all the sequence of common points occurs in an ordered way for both objects.

In the case of generic patterns associated with compound patterns, they are grouped into 8 types, which are: isolated objects [127], symmetry [36], repetition [36], propagation [127], convergence vs. divergence [133] [134], encounter vs. breakup [134], trend vs. fluctuation [36], and trend-setting [134] [127]. Figure 3.11 illustrates an example of generic patterns related to compound patterns, which are encounter and breakup.

As we observe in Figure 3.11(a), this pattern occurs when moving objects move to the same place at the same time. Encounter can be also defined as a convergence [133], since the moving objects are arriving in the same point at the same time. In Figure 3.11(b) we observe the occurrence of a breakup, where can be represented as the opposite of the encounter pattern. For example, ducks flying from a lake after hearing a gunshot [30].

In contrast to generic patterns, behavioral patterns is strongly related to



(a) Encounter. (b) Breakup.

Figure 3.11: Examples of encounter and breakup.

the type of moving object. Hence, behavioral patterns allow to recognize behavior characteristics for any types of moving objects. Along this line, the authors of [30] illustrated some behavioral patterns, which are: pursuit/evasion [135], fighting [135], play [135], flock [133], leadership [127] [134], congestion [30], and saccade/fixation [136].

For example, in the pursuit/evasion pattern, we can imagine an animal that is trying to escape from its predator. It makes sense that the prey generates a trajectory in high-speed with arbitrary movements. Nevertheless, the predator tries to follow the movement of the prey. Another interesting and useful example of behavioral pattern is the congestion, which the pattern can be associated with an anomaly. For example, one or a group of cars moving with slower than usual velocity in a certain segment of route.

With all these patterns in mind, it is possible to discover similar interests or behaviors between trajectories from the same moving object or from a group of them. Nevertheless, it is necessary to understand each feature related to the type of moving object or trajectory in order to achieve accurate results from spatio-temporal patterns.

3 Conclusion

In this chapter we introduced different spatio-temporal clustering methods. The main idea was to show another manner to identify similarities between spatio-temporal data through clustering algorithms. Hence, we explained in

detail how spatio-temporal clustering can be used on trajectories, providing an overview of recent related work and showing possible applications in various scenarios, such as environmental studies, transportation systems, etc. The combination between distance functions (Chapter II) with clustering methods makes possible the pattern discovering of moving objects, based on their trajectories.

Since the techniques to find similarities of moving object trajectories were introduced, we presented some commonly agreed definitions for spatio-temporal patterns. We firstly show the differences between spatial and spatio-temporal patterns, by emphasizing that temporal information can modify the pattern of a moving object or a group of them. After that, we showed an important classification of spatio-temporal patterns according to the number of moving objects (e.g., individual vs. group patterns) and according to the relations between moving objects (e.g., general and behavioral).

These clustering algorithms and pattern definitions are indispensable as a basis for the understanding of our approach, which uses patterns to recognize user interests based on the trajectory data. These user interests are described by Points of Interest (PoI), which follows the standard proposed by W3C POI working group [3]. Next, we present the current status of this standard and the conceptual view on Location Based Social Network (LBSN).

Chapter IV

Location-Based Social Networks

Contents

As we know, with the growth of GPS-embedded modules for mobile devices, large amounts of mobility data are being collected in the form of trajectories. A trajectory data is usually presented as a segment of sample points (TId, lat, lon, t), where (TiD) is a trajectory identifier, (lat,lon) is a position in space and (t) is the time [137]. However, these sample points are usually available with very simple or no semantics. For instance, a trajectory may be related to one or many points of interest (PoI).

In general, PoI is a location about which information is available. PoI can be represented by an identifier containing a set of coordinates or a three-dimensional model of a building with names in different languages, information about opening hours, and the address. The information of PoI is usually applied in a large number of solutions, such as mapping, navigation systems, location based social networks, networking games, and augmented reality.

In parallel, we have observed a large adoption of solutions of Location-Based Social Networks (LBSN), which combine smart phones and social networks technologies. As a consequence several mobile social applications have

been developed to register social behaviors of mobile users [15] including Ipoki[1], Google Latitude[2], Carticipate[3] and Daily Places[4]. Despite the availability of these mobile social applications to register and share users' daily routines, we face a rapid increase of diverse kinds of space-associated data, such as measurements from mobile sensors, GPS tracks, or georeferenced multimedia.

As prospective sources of useful knowledge and information, these solutions require a scalable data representation in terms of points of interests, which need to consider the particular attributes of the geographical space, such as heterogeneity, diversity of characteristics of relationships, and spatio-temporal autocorrelation.

Following this idea, we start the chapter by introducing the conceptual definition of social networks and their virtual communities. Next, we present the main definition of points of interests according to the representation specified by W3C PoI working group [3]. After that, we show other concepts of points of interest that have been used in some approaches. Finally, we address our attention to the most relevant research domain of our approach, the location-based social network (LBSN). In this context, we introduce the current works in this area as well as we show the main concepts and definitions related to LBSN's.

1 Social Networks

A community can be defined as a group of individuals that is distinguished by similar characteristics or interests [138]. While the number of communities is always increasing people establish limits to their social interactions taking into account some relationship and/or social structures. Social structures have been generally represented as social networks. Hence, a social network is a social structure composed of individuals, represented by nodes, who are connected by specific kinds of relationships, such as social associations, connections, or affiliation between two or more people [139].

Therefore, we observe that social networks become a relevant source of

[1] ipoki.com
[2] google.com/latitude
[3] carticipate.com
[4] dailyplaces.com

social knowledge about the individuals. With this knowledge the individual can determine social actions according to his/her interests or derived from a group of users with similar interests. For instance, according to the interests of an individual, we can decide to interact or not with him/her. The same example can be considered in the case of meeting this individual or not at some specific place. This characteristic of social networks gives a way to measure the risks or benefits for executing an action in a community [140].

The usual communities in the real word have been extended on the Internet through the creation of virtual communities in social network platforms (e.g., Facebook, LinkedIn, Orkut, others). These platforms provide a manner to store and explore knowledge about interests of individuals (who are called users). Due to the services provided by these social network platforms, users geographically distant can interact on the Internet.

Taking into account these advantages, we note that the virtual communities became a strong element of people's life and can be used as a source of knowledge to increase the interactions in real communities [141]. The virtual communities can be considered as an extension of real communities, which expand the knowledge about users. This new manner to provide knowledge derived from social network platforms provides access to interests and relationships of users, which were not available before. Consequently, this possibility to offer services based on social relations led to the designing and development of new approaches.

In the last years several approaches involving social network platforms have been developed in computer science and related areas. Due to the growing number of web technologies, we have observed a considerable increase in the amount of interactions between users on the Internet. Along this line, we present some relevant works in the context of social network platforms.

In [142] and [143], the authors showed that Internet opens new perspectives for analyzing knowledge about users in social network platforms, specially due to the hyper-textual structure, which is responsible to link the knowledge on the web. In [144], the author examined social networks platforms and user necessities in order to understand how computer networks, particularly on the Internet, are reinforcing and expanding social networks of real communities. In addition, A. Kavanaugh also explored the role of the Internet in increasing community involvement.

In [145] the author cited some of the trends and issues involving the Web

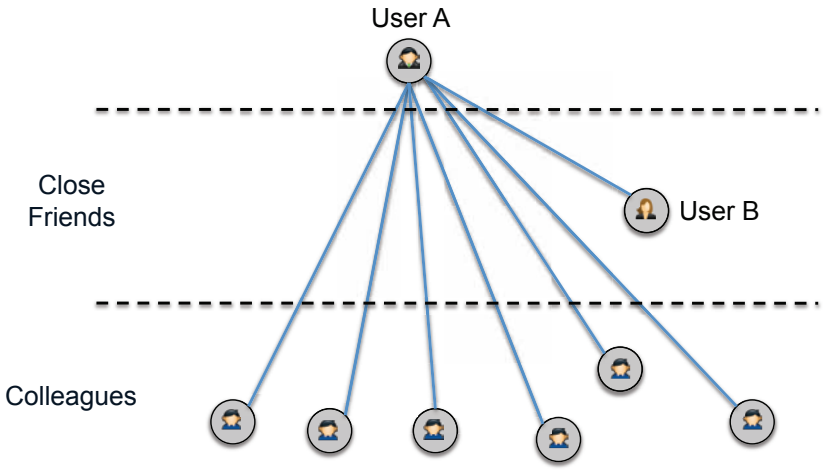


Figure 4.1: Example of relationship between users.

2.0 and social network platforms. It means that the understanding of the knowledge provided on the Internet is an important step before capturing, storing, sharing and using that knowledge. The author also provided a detailed analysis of knowledge networks, focusing on how the relations can contribute to the generation and representation of knowledge.

In the context of spatial information, the author of [146] addressed the approach to two types of social knowledge in a ubiquitous environment: social relations among users and semantics of places. Firstly, sensor and web data are used to define a social network. Next, a search retrieval system is implemented to identify collaboration between researchers. Finally, a method to represent the semantics of places is designed. It is used as an advanced navigation system, called a spatial function retrieval system. Similarly, the authors of [147] presented the MobiVis, a visualization system for exploring mobile data. The key idea of this approach was to incorporate in one heterogeneous network the manner that social and spatial data are presented for users.

Therefore, the knowledge obtained by the social networks platforms are used to provide information of relationships between users and their interests. Figure 4.1 illustrates an example of relationship links between users in social networks. Based on these links presented in this example, we can extract all the relations of user A in his/her social network as well as the level of each relationship in comparison to each user (e.g., colleague, close friend, family, etc). These data are included in our approach to enrich the user profile and define the access policies for each personal data. As we can note,

virtual communities of social network platforms became an important source of information about user's interests. The additional information about user interest is directly related to locations where the user goes, commonly called points of interests (PoI). In the following section, we show the concept of PoI according to our approach.

2 Points of Interest (PoI)

The W3C Points of Interest Working Group (W3C PoI WG) has defined a specification for PoI data that can be used in a large number of applications [3]. This specification aims at creating a flexible, lightweight, extensible PoI data model, as well as a normative syntax for this data model in order to provide best practices for sharing, organizing and serving PoI on the Web. Figure 4.2 illustrates the current PoI data model proposed by W3C PoI WG.

As we observe in Figure 4.2, the PoI data model is formed of a *POI* entity and a *POIS* grouping entity. While the POI describes its location and relevant features of the location context, the *POIS* comes to optimize the robustness of *POI*. *POIS* estimates a description expansion of a *POI*, taking into account the existing sub-entities, where each one can have the features of a single common entity. For example, *POIS* enables the designing of methods for updating time, links and authorship as well as for describing multilevel features in association with the current data model.

We note that *POIType* is the core of this specification, which has child entities derived from the *POIBaseType* entity. Making use of this *POIBaseType* entity, we are able to manage the information in different *POI* levels in association with *POIS* grouping entity. In terms of properties, both the *POI* and *POIS* entities can be composed of any number of the following child entities:

- **label:** is a human explicit label to name PoI.
- **description:** a human explicit description about the PoI.
- **category:** this entity classifies PoI into a category. For example, it can be a primary attribute (e.g., museum, bar, restaurant), a popularity ranking, or a security rating.

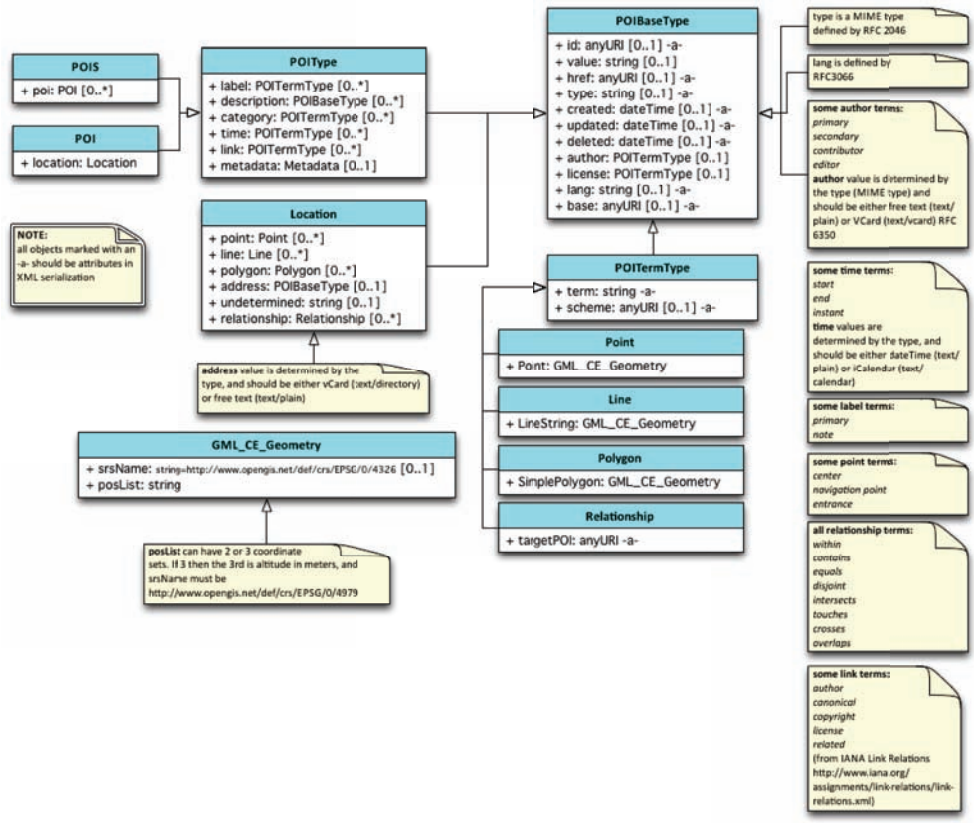


Figure 4.2: W3C POI Data Model [3].

- **time:** we can see the time as one of several contexts associated with the location. For example, we can have the moving object velocity, acceleration, wind speed, weather and time. Time is considered the most common context information related to the moving object, which is generally represented by the time instant that the location was acquired. However, a moving object can stop in a specific PoI for a long period, adding a start time and end time for this location [24]. Another example is related to the existence of PoI by respecting a regularly scheduled sequence of times. In summary, this entity manages the time instant or period based on the time specification and occurrence.

- **link:** this entity is a generic manner to represent a relationship from a PoI to another PoI, or from a web resource to a PoI, both based on the RFC 4087 technique (point-to-point link).

- **metadata:** in this entity, we can insert formal metadata to the PoI

(by reference, for example).

A more detailed description of all entity in the PoI data model can be found on the webpage [5] of W3C points of interest working group [3].

Another approach was introduced in [148], where the authors considered a point of interest (PoI) as a location in a road network. In this case, they do not consider temporal information as information into the PoI. Nevertheless, they defined another entity to consider the temporal information, called Time of Interest (ToI). With these two entities, they proposed an algorithm to find similarities between moving object trajectories in road networks. For example, if two trajectories have points (e.g., locations) with similar PoI and ToI, the algorithm will find a similarity between these trajectories. Figure 4.3 illustrates a scenario containing two moving object trajectories.

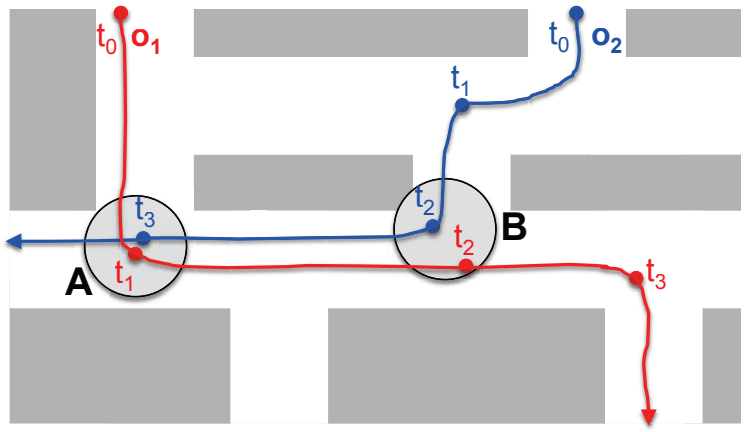


Figure 4.3: Example of similarities based on PoI and ToI.

Therefore, PoI and ToI were divided in two filtering processes, called spatial filtering and temporal filtering. Firstly, the algorithm finds the spatial similarities based on the PoI of each moving object trajectory. Since the similar positions are found, it executes the temporal filtering in order to identify temporal similarities in the spatially similar points. Taking into account Figure 4.3, we observe two trajectories containing similar PoI's (represented by the circular regions **A** and **B**). When these two spatial similarities are found, the algorithm compares if the time instant t_1 from o_1 is similar to t_3 from o_2 at the PoI **A** and if t_2 from o_1 is similar to t_2 from o_2 at the PoI **B**. Finally,

[5] http://www.w3.org/2010/POI/documents/Core/core-20111216.html

if both moving objects have similar PoI and ToI, they will be classified as moving objects that share the same interests.

In [149], the authors formally define a point of interest (PoI) as a tuple *PoI = (lon, lat, concept)*, which is composed of at least a longitude, a latitude and a concept. Similarly to the previous approaches, they defined PoI as any specific location that represents a point in a trajectory. However, they introduced a new parameter, called *concept*. By definition, the *concept* is a tuple *c = (name, children, ...)*, which is formed of a string *c:name* (that describes the concept) and a set of child concepts *c:children* (that are the features of *c*). Thus, a *concept* is indicated by its *name*. For instance, we can assume that the *concept* "*Fast food*" contains the *child concepts* "*Quick*" and "*KFC*". Finally, a distance between two concepts *c* and *d* is computed by the function *conceptDistance(c, d)*.

To better represent the distance between concepts, the authors created the *concept hierarchy*, which is defined as a forest of concepts. They suggested the possibility to have multiple roots based on this *concept hierarchy* and that the distance between any two concepts is defined as infinity when they are not sharing a common position. For instance, assuming two concepts c and d, such as $c = d$ or c is an ancestor of d, then we have $c \geq d$, and confirm that c is a super-concept of d. In sum, the authors represented this classification by the *depth* in which the *depth* of a *concept* x (*depth(x)*) is equal to the number of edges between x and the root of the *concept hierarchy* containing x. For example, we assume that z is the lowest common ancestor of x and y. Thus, if z does not exist, then $conceptDistance(x, y) = 1$. On the other hand, we have

$$conceptDistance(x, y) = max\left(depth(x) - depth(z), depth(y) - depth(z)\right),$$

where *depth()* denotes the depth of a *concept* in the *concept hierarchy*.

As we can observe, all these definitions go in the direction to the PoI data model specified by the W3C-PoI working group. The main differences are related to the insertion of temporal information to compare similarities between two interests between different moving objects. However, this difference is more associated with the algorithm and the process than with the data model. Taking all these suggestions into account, we decided to follow the specification proposed by the World Wide Web Consortium (W3C).

The main motivation to know about PoI data models is related to ob-

taining different concepts in order to design a suitable data model for our approach. This suitable data model can provide an easy way to manipulate trajectory data, to use structured query languages, to specify profiles through movements, to create and compare profile groups.

3 Location-Based Social Networks (LBSN)

Location-based social networks are a new kind of research area that focuses on the use of spatio-temporal features to process location data of users and associate these data through social networking relations. This new concept was introduced in the last couple of year by the authors of [150] as a solution to share enriched content to a large number of applications (e.g., mobile social networks). Intuitively, a mobile user (moving object) acquires and records his/her locations by using some GPS-based application and shares this information with other users connected in social networks. Based on the relation information between locations and relationship, LBSN is able to infer about new information and provide it for a large number of applications or services.

Therefore, LBSN is formally based in two main parts [4]. The first one is related to the data model of users' locations histories according to their trajectories. The second one is associated with the similarity analysis procedures to discover resemblances between users' locations or trajectories in social networks. Finally, a similarity data can be created by indicating the type of relations between two friends, who are directly connected in a social network (e.g., Facebook [12], LinkedIn [13], etc). The results can be used in several applications, which goes from simple recommending systems up to complex query-based systems.

Next, these two parts of LBSN are explained in details, taking into account the spatio-temporal data of users.

3.1 Data model of user location history

As we saw in the previous section, before starting the analysis of trajectory data, it is important to model the user location history. Some works attempt to represent relevant locations from the trajectory data, without taking into account the social relations between users [151] [152]. It means that they do not consider the comparison between locations histories and social relations

before modeling the trajectory data of different users. However, in [153] [154] [155], the authors introduced some approaches involving this domain. These approaches generally follow the following steps

$$(acquired\ data \rightarrow geospatial\ locations\ (important\ places) \rightarrow semantic\ representation\ (e.g.,\ museum,\ coffee\ shop)\ ).$$

Following these steps, it is possible to model users' location histories and provide an easy way to compare and find similarities. Besides that, the modeling of users' spatio-temporal data allows the knowledge about user's behaviors and interests, which can be represented, for example, by users' routines.

In [4], the authors proposed a framework based on a hierarchical graph (see Figure 4.4). The key idea of this graph is to model each user's history in the space. Formally, the framework is composed by three parts, which are explained as follow.

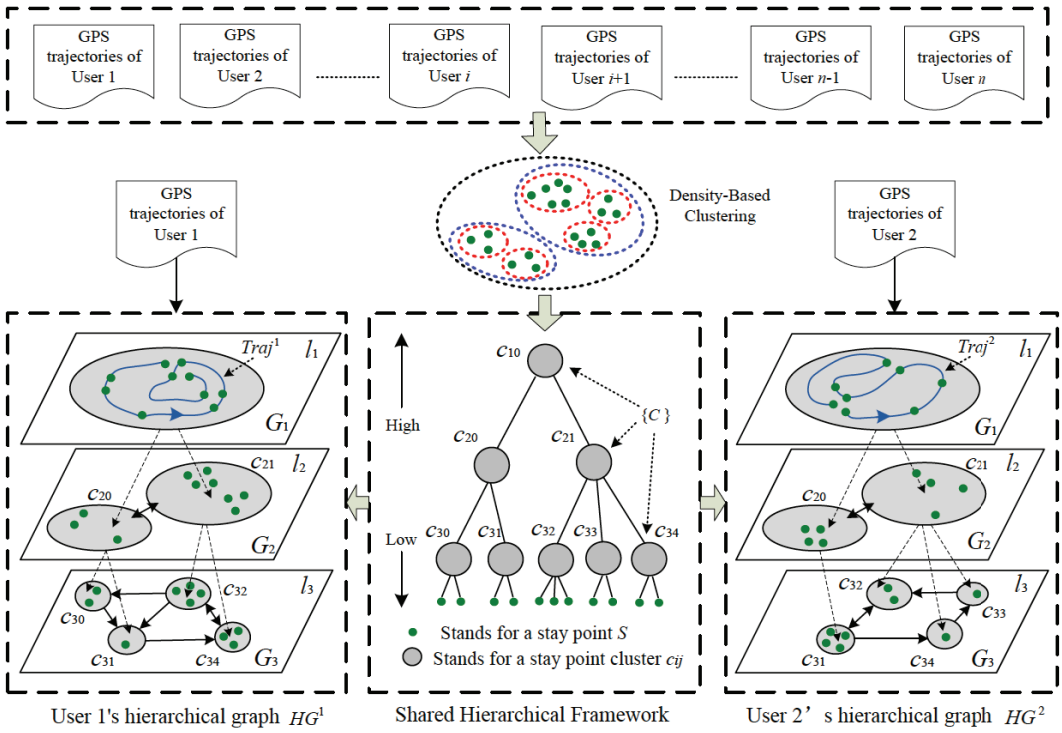


Figure 4.4: Framework for modeling users' location histories in geographical spaces [4].

- The first part is associated with the acquisition and detection of important places. Taking into account the framework and the trajectory

data, the authors define the stay points for indicating the geographic region where a user stayed for a period of time. In summary, they characterize a stay point as an element semantically comprehensible, such as restaurants, museums and parks visited by the user.

- Following the first part, every user's stay point is stored into the database in the second part. After that, a clustering method is used to recursively group the information in a well-organized manner. Consequently, the groups containing similar stay points are created and a multi-granularity model is obtained, which contains groups with different layers for representing the locations by geographical regions. This final result contains the structure of clusters, which provides different and relevant manners to construct different graphs, according to data model requirements. This part was called shared framework formulation.

- Finally, in the third part, the construction of the user's location history is done, based on the shared framework formulation. Every personal directed-graph of a user is created by estimating the user location history in the shared framework formulation. It means that a cluster of user's stay points is indicated by the graph nodes and the graph edges symbolize the movement of a user from a cluster to another (see Figure 4.4).

These three parts need to be well defined to allow the discovering of similarities between users based on their location histories. We present some related works involving this subject in the next section.

3.2 Applying user's location histories in real scenarios

We have presented a well-known manner to represent spatio-temporal data based on users' location histories in LBSN's. Since these data about users are defined, it is possible to start the process for finding similarities between user locations. In [5], the authors use the previous framework in supporting the understanding of user locations jointly with social knowledge to evaluate the model in scenarios of generic recommendations [6] [21] [22] and personalized recommendations [156] [19] [16]. According to these authors, a generic recommendation is a service to offer information about a specific location,

constrained by users' departure positions and periods of time. This recommended location-based information can be important places (e.g., restaurants, supermarkets, etc), trajectory segments, travel experts and effective itineraries in a specific region. In contrast, the personalized recommendations consider the user interests to discover the locations matching with them, which can be derived from the user's locations histories.

3.2.1 Generic recommendations

By definition, the generic recommendations process several user trajectories in order to find relevant information for providing a recommendation. This recommendation is generally described by the following sequence of processes

$$trajectories \rightarrow important\ locations \rightarrow popular\ trajectory\ segments \rightarrow itinerary\ planning \rightarrow activities\ recommendation.$$

When we observe this kind of recommending system, we note that it initially identifies the most relevant locations (by inference, for example) in a specific region based on the trajectories. After that, it performs a procedure to find popular trajectory segments according to these important locations [14]. Some examples of important places can be *Eiffel tower* or *Louvre museum* in Paris. In addition to popular touristic places, important location can also be defined as frequented public places, such as theaters, cinemas, supermarkets or restaurants. Along this line, the system is able to discover an itinerary according to the user starting position, destination and period of time [6] [21]. Users can receive recommendations about popular activities to facilitate their travel planning, such as: best hours to go to a specific supermarket or restaurant; city streets less congested; others.

We can imagine people going to an unknown city for a short trip. They would like to know some suggestions (or recommendations) about this place. One manner to suggest touristic information is discovering the most popular trajectories followed by previous tourists, derived from their location histories. Besides that, the recommendation can consider enriched information to be more accurate, such as the period of the day, the season, weather. On the other hand, the recommendation system can be interesting to the residents, informing possible time periods of the day that a certain region has a large number tourists.

In the context of social networks, the author of [150] presented a manner to provide travel recommendations automatically by obtaining temporal information from social data, such as the large number of GPS trajectories recorded by several users who have traveled to this unknown city. Therefore, the trajectories can be constructed based on users' geo-tagged multimedia content and/or check-in, which are available on the trajectory-sharing social networking service, called GeoLife [150] [20] [16].

Some works have presented different manners to perform travel recommendation by analyzing geo-tagged photos from trajectory data [157] [154] [158]. However, pertinent challenges associated with generic recommendations have been sowed in the literature. While an important location can be discovered by a popular location from many users' location histories, the knowledge of these users have to be considered before recommending something. For instance, we can implement different values to a recommendation based on each location, taking into account each user's knowledge about different regions (e.g., a qualified tourist can find best places to visit in any city easier than an unskilled one). However, we have to consider different levels of qualified tourists for this example, since a tourist who knows Paris, Rome and Chicago may have no idea about Rio de Janeiro.

Finding important locations

With these challenges in mind, the authors of [11] introduced some methods to find important locations based on users' location histories. Figure 4.5 illustrates the building process to construct a Tree Based Hierarchical Graph (TBHG).

The following steps explain the two processes to construct the TBHG presented in Figure 4.5:

1. This first process follows the second part presented in Figure 4.4 for formulating a shared hierarchical framework F. In summary, the users' stay points are obtained and stored in the dataset before applying the clustering algorithm. A density-based algorithm generates clusters of regions in different levels, where each cluster is composed by similar stay points of all users. In this example, a stay point represents a location where a user stayed for a specific time interval.

2. This process is responsible for the construction of location graphs in

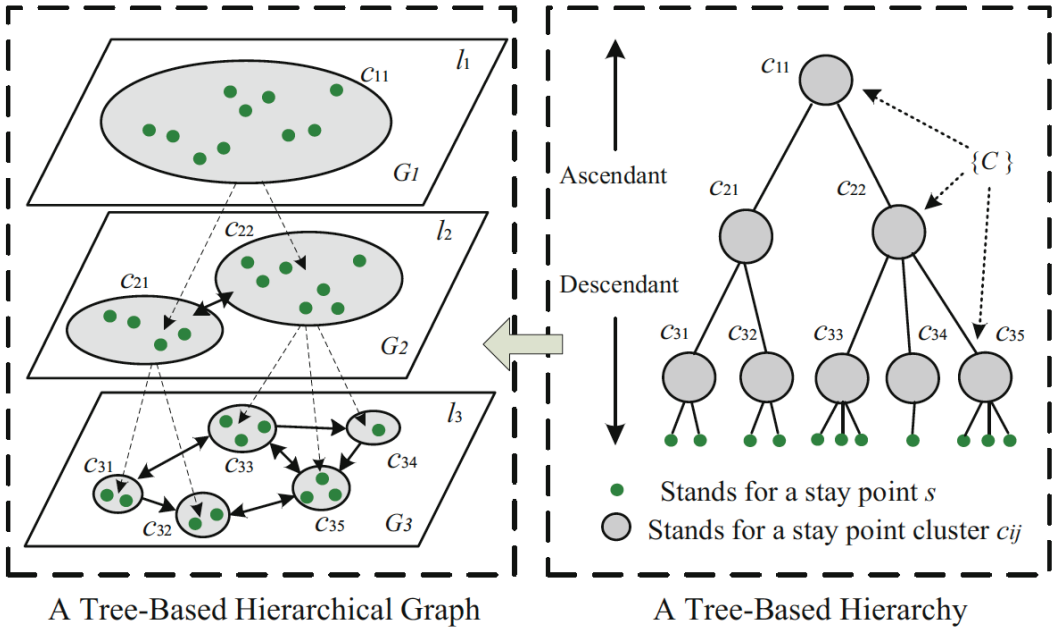


Figure 4.5: Constructing a Tree Based Hierarchical Graph (TBHG) [5].

multiple layers, taking into account the framework F and the users' location histories [5]. In other words, a link between two clusters is created when two consecutive stay points from one trip are individually included in both clusters. The approach considers the time serial of the two stay points to create a link between these two clusters in a chronological direction. Hence, in contrast to the third part of Figure 4.4, this process uses all sequences of stay points in users' location histories provided by the framework F. Finally, the location-based social networking service makes use of the constructed TBHG as the data representation of all users' location histories.

At this moment, a Hypertext Induced Topic Search (HITS) is used jointly with the TBHG to infer about new values. Formally, HITS is an inference model that verifies the association of a user with a specific location. Two models are inferred in this step, the user's travel skills and the interest level of a location. Thus, the inferred model considers how strong is the relationship between these two values and the spatial-based knowledge.

This approach is interesting in the context of LBSN services. However it brings some points that can be explored. For instance, while the number of important places in a city is constrained, then the size of the location graph

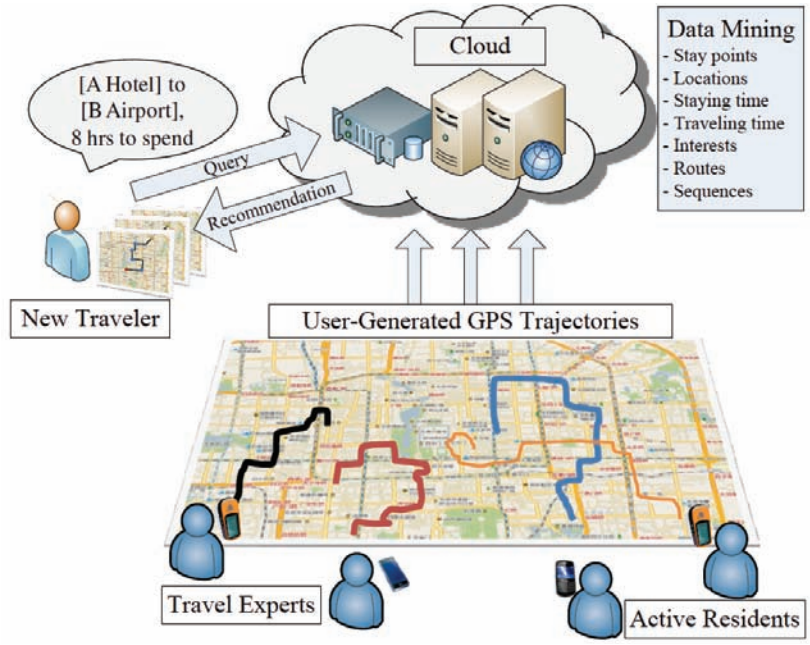


Figure 4.6: Example of a scenario to recommend itineraries [6].

becomes small. Nevertheless, the authors of [5] show that sequence with three locations becomes more interesting than big ones.

Recommending specific information

Following the example of people that visit unknown places, we note that the step to find important locations offers an easy manner to recommend relevant information. Furthermore, it is also important to facilitate the users' travel plans based on their necessities. For instance, many tourists would like to visit the maximum number of important locations during their trips, in a comfortable way, saving money and going fast from the current place to the next nearest important location.

This example of recommendation has been explored for a considerable number of researchers. In [159], the authors implemented an interactive recommending system, which the user determines his/her limitations (e.g., important locations to be considered in the recommendation as well as the periods of the day). A similar approach was introduced in [160], where the main difference is related to the locations that the user would like to avoid. As we observe, these strategies are commonly based on the user intervention and needs certain user knowledge about the location.

Based on these limitations, some works were introduced by adding automatic processes to recommend locations [161] [162]. Besides these works, the authors of [6] introduced a recommending system to send interesting itineraries based on the social relations. The approach first uses a process to generate itineraries by taking into account the user's query and social relations. Given the current position, the time period and a certain region, the social relations derive from users' trajectories to recommend important locations to visit. Figure 4.6 shows an example for this recommendation system.

According to these examples, we observe that the best way to provide a generic recommendation is to consider the period of time that a user stays in a location as well as the knowledge about the duration time from a current position to each intermediary locations and final destination. In addition, to be more accurate, the system has to be adaptable to several types of users. Therefore, we conclude that four main aspects are related to recommending systems, which are the departure position, intermediary locations, destination, and available period of time.

3.2.2 Personalized recommendations

Another way to recommend information based on LBSN service is provided by personal recommendation systems. In contrast to generic recommendation, a personalized recommendation is strongly associated with the user's interests. It means that the important locations to be recommended are based on the social relations with the user's similar interests. Considering the example of a person that visits a unknown city, the personalized recommendation system makes use of the times that a particular user has visited a location and possible rates about this location in order to estimate the interests of another user in unvisited locations. This process is performed by considering the user's location history and location histories of users that match with the user's interest [20].

This explained process can be described by a table containing users in each line and groups of users in each column. A good way to construct this table is finding the clusters most close to the interests of a specific user. Consequently, Table 4.1 can be used to infer the ratings of each cluster (in the columns) in comparison to each user (in the lines) in order to recommend some location.

Taking into account this table, we can see that each group is composed by

	Group 1	Group 2	Group 3
User 1	1	3	2
User 2	4	5	5
User 3	5	3	3
User 4	2	1	3
User 5	3	1	1

Table 4.1: Comparison between user's interests and interests of groups of users.

users with similar interests. Each group is associated with a rating of their knowledge of previous visits in a specific location. Besides that, the groups have a similarity value in comparison to the interests of the users (presented in each line of the table). Hence, the recommending system can infer about a recommendation according to the relations between each group and each user.

In [5], the authors introduced a method of collaborative filtering to design a personalized recommending system. By definition, the collaborative filtering algorithm provides a manner to make ratings for similar features [17] [18]. This rating is then compared and the recommendations are sent to users with similar interests. For example, if two users are friends in the social network and they have similar interests in their location histories, the recommendation system can use the collaborative filtering to find similarities and send an alert to one user about a near place where the other user has already been.

3.2.3 Collaborative Filtering

As previously defined, the main objective of collaborative filtering (CF) is to obtain ratings of users for supporting the identification of similar interests between users. If a similarity is discovered between users, a most-likely prediction can be performed. With this in mind, we now address our attention to talk about CF models, by showing the classification proposed by the authors of [163], which are model-based and memory-based algorithms.

Model-based algorithms make use of rating to construct a model. After that, the model is used to predict ratings [164]. For instance, the authors of [164] designed a collaborative filtering algorithm based on a machine-learning framework, where several types of machine learning techniques (e.g., neural

techniques) can be used to predict ratings. Another model-based algorithm was demonstrated in [163], where the authors implemented a probabilistic approach to CF. The key idea is to use rating values between 0 and n and a probabilistic distribution that considers the probability that a user will attribute a specific rating to a location, since there exist previous user's ratings for this location.

In the case of memory-based algorithms, the process performs ratings predictions taking into account the whole collection of previously rated locations [165]. Two groups of memory-based collaborative filtering are possible: user-based [166] and item-based [167] methods. These two groups are presented as follow.

- **User-based methods:** these methods use similarity measures between users to obtain rating predictions. For instance, we assume two users (X and Y) and the similarity between them is basically the distance, which is used as a weight. If we are interested to predict the rating of user X for a location and X and Y are very similar, then the rating of user Y will be important to determine the interesting location to user X. We can find a large number of approaches that use the rating of uses for elements in order to define the similarities between then. In [71], the authors introduced an empirical comparison of six different similarity measures to recommend virtual communities in the Orkut social network. In addition, these authors concluded that Cosine similarity measure obtained best results in comparison to the others.

- **Item-based methods:** in contrast, these methods consider the rating between items to predict the ratings. In other words, they predict the ratings of an item A according to the ratings of an item B. These methods have been widely used on applications that sell products on the Internet. The authors of [167] presented a simple manner to compute the average difference between the ratings of two items for users who rated both. In Another well-known example was presented in [168], where an item-to-item collaborative filtering is made by applying the Coisine technique to compute the similarities in a *itemxuser* matrix.

In [5], the author presented two challenges related to the designing of recommendation systems based on location-based social networking services.

These challenges are strongly associated with the cold start problems and scalability. The cold start is typical problem of collaborative systems, which is caused by the entering of new users or items in the recommending system. Although the number of locations is constrained in real geographic spaces and generally smaller than that of users, the problem related to the scalability of a location recommendation system is directly associated with the increasing number of users.

4 Conclusion

In this chapter, we started introducing the conceptual definition of social networks and their virtual communities. Besides that, we presented some relevant works in the context of social network platforms. Next, we presented the concept of points of interest (PoI) as a prospective source of useful knowledge and information. We showed some relevant approaches, which each one followed a different way to represent and process a PoI. The authors of these works designed scalable data representations in terms of important locations to a user, by considering particular attributes of the geographical space, such as heterogeneity, diversity of characteristics of relationships, and spatio-temporal autocorrelation. Nevertheless, we have to stress the importance of the PoI data model proposed by the W3C Points of Interest Working Group (W3C PoIWG). This data model is used as basis to the model proposed in this thesis, due to its flexible, lightweight and extensible specification, as well as its normative syntax in order to provide best practices for sharing, organizing and serving PoI on our layer of services for Location-Based Social Network (LBSN).

Following this idea, we also showed the LBSN as an important and challenging topic in terms of similarity analysis of spatio-temporal data. Since we presented the concepts of PoI, we showed a comprehensible way to represent users' interests derived from the locations that they have been stayed. Therefore, we introduced some important concepts and related works related to LBSN services and applications.

From the state of the art review, we concluded that although there are several LBSN services for recommending systems, which are relatively well designed, most of them do not consider the relations between friends in social

networks and need the direct interaction of users to discover some of their interests. Finally, we have also observed that LBSN was born as a strategy to facilitate the recommendation of important places to users, which is generally provided by recommendation systems. In this context, in our thesis, we have focused on the designing of a layer of services based on LBSN in order to provide a solution to capture users' daily routines for finding correlated information between users. Our approach is presented in the next chapter.

Part II

Proposition

Chapter V

LIDU - Location-based approach to IDentify similar interests between Users in social networks

Contents

We have observed that people work or live in different places but have trajectory correlations in their daily routines. The users' daily routines, therefore, can be captured by mobile social applications and shared in virtual communities in order to increase social interactions in real communities.

Since we have noted this viability to increase social interactions in real communities and the large widespread of smartphones and social networks, we propose a layer of services based on user's daily routines, called LIDU: Location-based approach to IDentify similar interests between Users in social

networks. The key idea is to increase social interactions by relating daily routines and points of interest based on trajectories of mobile users. For instance, a mobile social application jointly with a social network has to be able to answer the following questions:

1. Which of my friends stop in my preferred bakery in Grenoble at the same period of the day?

2. Do any of my friends pass near my apartment to get from their home to their work?

3. Which of my contacts will be passing into the campus of the University of Grenoble during the week? [1]

While these questions are interesting to obtain information of similarities between users' daily routines, some scientific challenges were considered in the designing of our approach. The challenges are mainly related to traditional and new problems involving social networks, mobile computing technologies and spatial data representation. We point out these challenges as follow:

- Determine the relations between users of social networks.

- Propose integrated software architecture according to the characteristics of mobility scenario, such as limited resources, network and sensors.

- Define the structure of user profiles in order to facilitate the association between trajectory and context data of users.

- Design a robust data model to describe the spatial environment, taking into account different levels of the spatial information. The data model helps the classification of spatial knowledge based on points of interest (e.g., bakery, apartment, campus), spatial relations (e.g., near my apartment) and geographic entities (e.g., Grenoble).

- Extend the data model to represent the relations between spatial and temporal data, which allows the characterization of user's trajectories in multiple context information.

- Consider the aspects of the quality of data (e.g., sensored data) and the sharing of personal users' information (respecting the privacy features).

[1] The user defines the contacts to share his/her daily routine.

- Explore the available knowledge in order to identify spatial and contextual similarities between users, taking into account the performance and robustness of the approach.

- Propose a generic system to provide adaptable services for different types of applications, such as a recommendation system.

With these challenges in mind we decided to propose a layer of services focusing on bridging the gap between applications and low-level constructs [169]. This layer of services has to be able to achieve the requirements of these presented challenges and provide more features that facilitates the extension of our approach, such as scalability, heterogeneity, dynamicity, adaptability, knowledge managing, data association, quality of service and security. In summary, this layer helps developers to create applications that make queries to the layer and get results back in an efficient way [170].

Figure 5.1 illustrates the architecture overview of our approach. As we note, two main input data are acquired, which are trajectory data (jointly with context information) and social connections between users. Social connection data are directly processed by the data-modeling algorithm. Clustering algorithm receives the GPS trajectories to discover the best representative trajectory of each user. After that, the correlation algorithm identifies the similar points of interest between users. Finally, the data-sharing algorithm is able to adapt the information according to the requirements of each application.

Formally, our approach allows the execution of algorithms to capture, store, process and share similarity information derived from users' daily routines. Firstly, we use smartphones and their sensors to capture users' daily routines and context information. Secondly, all information is transferred and stored in a relational database located on a server application, which is used as a plug-in on a social network platform. Besides that, we explore the capabilities provided by clustering algorithms to analyze user trajectories and extract relevant information from a large amount of data. Finally, we use an optimized trajectory correlation algorithm to identify similar routines between friends in social networks.

Although the core of our approach is situated in the middle of the presented architecture, the data acquisition process has to be well defined in order to provide the relevant data to our algorithms. Therefore, we present the data

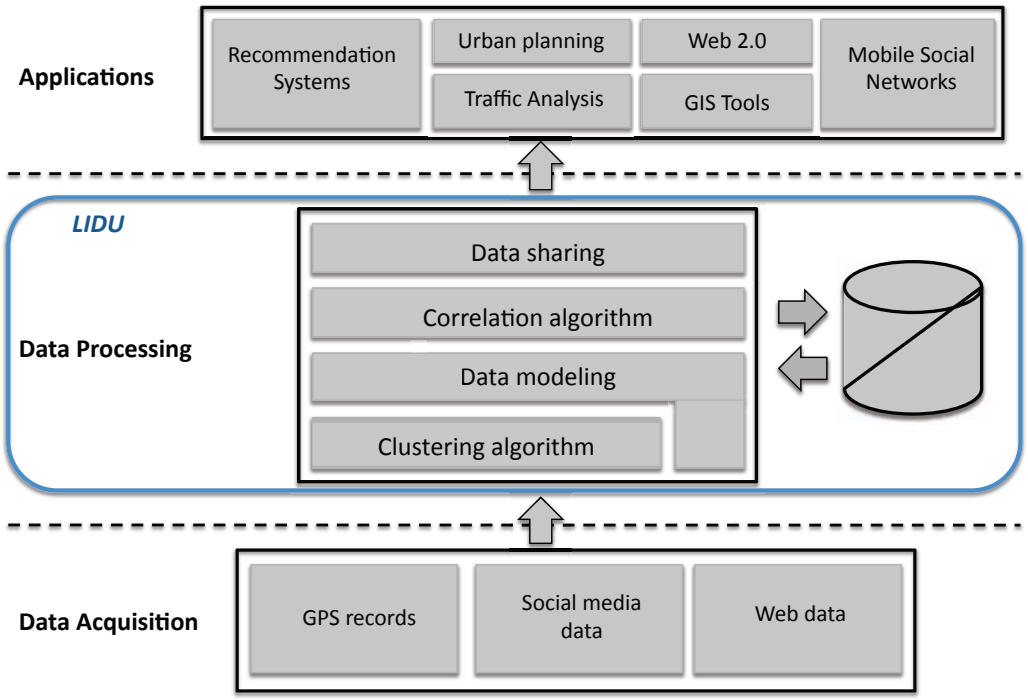


Figure 5.1: Architecture overview.

acquisition process, called profile building entity. The profile building can be denoted as an algorithm to acquire trajectory data and their context information through the use of smartphones. After capturing the profile building component sends the acquired data to the trajectory correlation component, which is the core of our approach. At this moment, the algorithms process the data. These two main entities are presented in Figure 5.2.

Therefore, we start showing the main parts that compose the data acquisition module, which was adapted and implemented to our approach.

1 Profile building

The user profile can be determined taking into account two basic types of data that are used for constructing and enriching the data model. These two basic types are defined as *personal* and *contextual* data. Personal data describes the main features of an entity and the contextual data characterizes the situation. An entity can be a person, place, physical or computational object. For example, in a personal tracking application for mobile users, the personal data

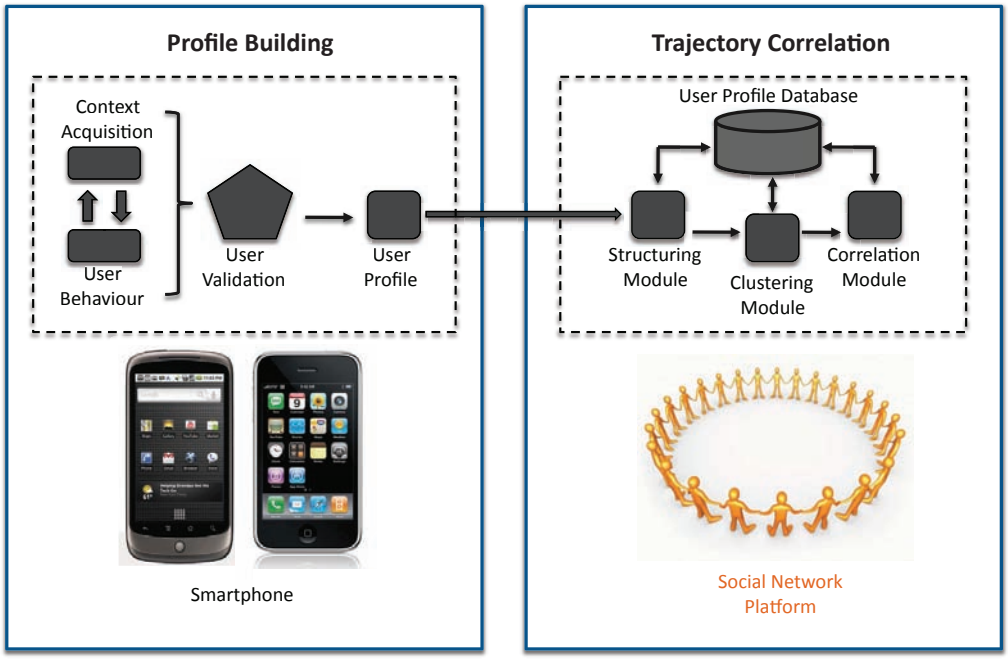


Figure 5.2: Main components of our approach.

would be the information about the user, such as name, birthday, gender, etc. On the other hand, contextual data would be composed of movement records that the user performed over a period of time. A movement record can include such characteristics as the initial point, speed, direction, and time, as well as weather information. We define an entity as a mobile user using a smart phone equipped with GPS, digital camera and Internet connection (e.g. 3G or Edge).

For the contextual data organization, we have followed the concepts and relations of Context Top ontology, introduced by the authors of [7]. Figure 5.3 illustrates this ontology, where *Action* has a *Context* that is composed by some *Context Elements*. The context can also describe the situation of its elements through the property of *describeTheSituationOf*, which *hasContext* is its opposite property.

Based on the Context Top, we divide the context in five main dimensions: social, spatial, temporal, spatio-temporal and computational. The social dimension is related to the features associated with the user, such as user profile and social relations in a social network. The spatial dimension provides the spatial information about the environment where the action is done, for example: geographic coordinates, postal address, etc. The temporal dimension

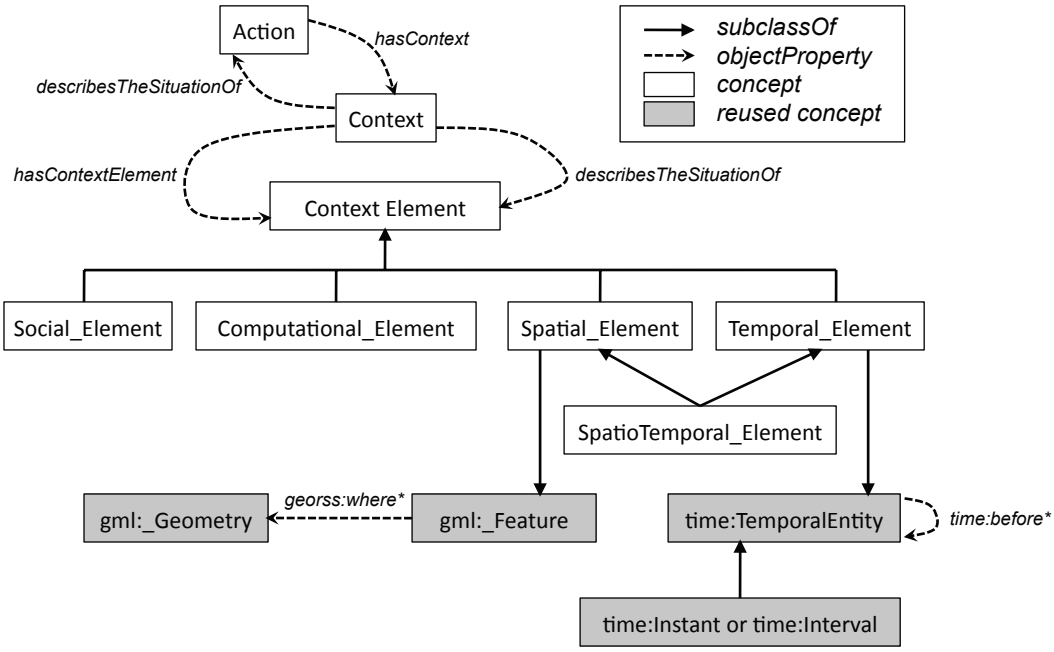


Figure 5.3: Context Top ontology concepts and relations [7].

is composed by the information about time, such as the date, the days in a week, etc. The spatio-temporal dimension has the information derived from the spatial and temporal dimensions (e.g., weather). Finally, the computational dimension offers the facilities provided by the embedded software in the system (e.g., sensors, mobile applications, etc.). Therefore, the features that are used in each dimension are defined by the developer of the context-aware system. We have adopted the same data organization presented in Figure 5.3 for defining the context data generated by our profile building process.

We have also defined a third type of data, named *behavioral* data, which is derived from the association among personal and contextual data. We assume that behavioral data is defined as a user's daily routine that is generated based on the elements that compose a user trajectory and its associated contextual data. In other words, since the personal and contextual data are well acquired and associated, our approach allows the identification of a user's daily routine. We have two ways to identify a user's daily routine, based on a single trajectory or derived from a set of trajectories (e.g., a user that goes from home to work every day). Both ways can be executed by following our profile building process, illustrated on Figure 5.4.

The user can use a mobile application to register a single trajectory that

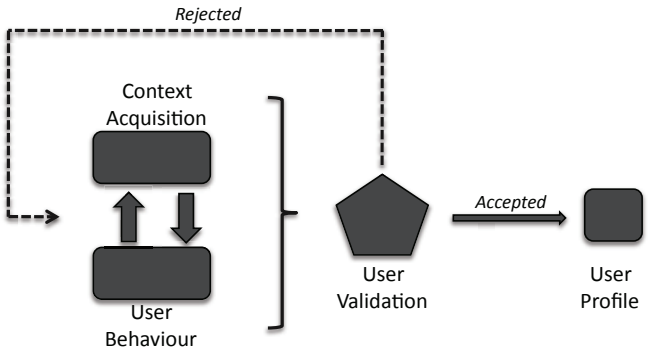


Figure 5.4: The profile building process.

describes his/her trajectory to go from home to work, for example. After visualizing and validating the trajectory that represents his/her daily routine, the user profile is created and the data is sent to the core of our approach. Social connections are already available by some social network platform (e.g., Facebook, LinkedIn, Twitter) on the Internet. Therefore, this single trajectory and its contextual data are used to represent the user's daily routine.

On the other hand, the second way to define a user's daily routine is discovering his/her best representative trajectory from a set of trajectories. Since the user registers more than one trajectory to represent the same daily movement, a clustering algorithm technique can be applied to recognize the best representative trajectory. For example, a user took the similar path to go from home to work for 3 times in a period of 5 days. For the other 2 days, this user decided to change the path due to some incident or problem. For this reason, the 3 similar trajectories could be used to represent the best representative trajectory. Consequently, this best representative trajectory represents the user's daily routine.

In our approach, we provide a method to recognize a user's daily routine from one or multiple trajectories. Following the steps presented in Figure 5.2, the structuring module verifies if there is a previous trajectory for the user. If there is no trajectory, it creates a new user's daily routine. On the other hand, if multiple trajectories are found, clustering and aggregation techniques are used to identify the aggregated trajectory (a best representative of user's daily routine) [171]. As previously mentioned, we apply the OPTICS algorithm to classify user trajectories based on their daily routes.

The clustering and aggregation module provides the best representative

trajectory for each user. This aggregated trajectory from one user is compared to other users by applying our trajectory correlation algorithm (Section 3). This approach enables groups of users to share similar routes to increase geospatial social interaction. The user daily routine then is enriched with additional information about each location in the database. The structuring module then exports the enriched information to update the user profile database.

Assuming that these data are available on the Internet and, consequently, are connected to some social network platform, the social relations can be used to enrich the database. The structuring module requests the social relations for each user who has registered his/her trajectory on the database. Hence, the comparison is performed based on the type of relation between users, for example: best friends, family, colleagues, etc. In our study, we assumed that the comparison of trajectories could use this feature as a filter to avoid security problems, mainly involving privacy.

While the capabilities to capture a sequence of positions, to enrich the database and to discover the best representative trajectory are the starting point of managing movement, designing a approach based on trajectory data requires a structural approach. After obtaining these trajectories, modeling them becomes necessary for important operations, such as: i) to indentify patterns, which will be used for decision making (e.g. registering users trajectories within a city for optimizing traffic of vehicles); ii) to query information about the moving objects (e.g. enriching trajectory data with context information); iii) to optimize intelligent transport systems (e.g. motivating users to use car pooling alternatives in order to reduce the number of vehicles in urban regions).

2 Multi-layer data representation based on user routines

The main motivations to design a suitable data model are related to providing an easy way to manipulate trajectory data, to use structured query languages, to specify profiles through movements, to create and compare profile groups. In parallel, the identification of the scenario is a significant requirement to design a conceptual data model. In this thesis, we take into account the

scenario of an employee that goes from home to work and back everyday within a city, whose the user's daily routine can be represented at different abstraction levels. In addition, we consider a diversity of semantic data that enriches the knowledge on these trajectories. For a user daily trip, we can obtain information about possible user interests based on his/her movements. For example, whenever the user goes from work to home, he usually stops at a specific coffee shop.

Therefore, the conceptual model for trajectories must be able to analyze and manage simple trajectories (direct travels from origin to destination) as well as complex trajectories (where the trajectory is semantically composed of separate segments and/or different abstraction levels). Furthermore, the data model must relate any type of semantic annotation to trajectories, such as attributes of each trajectory and connections between the trajectory and an object stored in the database.

Often, it is important to understand the movement data at multiple abstraction levels for pattern recognition and analyzing movement behaviors as well as to deduce the relationships between users in location-based social networks. In order to create a flexible data model for mobile social application context, we propose a multi-layer data representation of moving objects based on user routines. A specific place as well as a segment or a whole trajectory can denote these user routines.

Several researchers have shown an interest in analyzing and representing spatio-temporal data [15][6][14]. This data is relevant in a number of areas such as social interaction, data mining, medicine and geographical information system. For instance, in the context of social interaction, we pointed out some approaches related to collecting and analyzing daily trajectories of humans, addressing issues such as daily routine, mobility, sport, trips, and social networks. In all these approaches, the amount of data produced is very large and is therefore challenging to interpret.

In parallel, the need for representing information about PoI on the Web has emerged [3] in order to manage and organize context-aware information. Interesting issues include how points or regions can be correlated through multi-layer representation [172] and how user trajectories could be analyzed in terms of their distance to another one [173].

Multi-layer data representation has been of interest for a long time due to its importance for spatial data representation [174][175][176]. In spite of

the large number of issues about multi-layer data representation, there is a lack of multi-layer representation techniques for moving object trajectories. In [177], the authors present a design for multi-layer spatial objects in which both spatial objects and the vertices of their component geometry are labeled with level priority values. Although the data model supporting queries at different abstraction levels is very interesting, it is not intended for representing trajectories and not easily extendable for this context.

In [178], the authors present an interesting Rule-based Location Prediction method (RLP), to guess the user's future location for location-based services. However, they do not consider the partial containment relationship between spatial regions at different spatial levels. In [179] and [180], the authors introduce approaches to consider trajectory patterns between different spatial levels as well as the relation among user, location and trajectory. In particular, GeoLife [179] is a social networking service which increases social connectivity among users taking multiple geospatial scales into account while the work described by [180] focuses on Regions of Interest (ROI) as opposed to multiple abstraction levels. In this thesis we extend the PoI data model proposed by W3C working group and present a multi-layer data representation of correlated trajectories, taking into account the PoI at multiple abstraction levels.

As introduced in Chapter IV, PoI is composed of any number of the following entities:

- **label:** is a human explicit label to name PoI. This entity is important to identify a specific place, which can be used to support the definitions of labels in the different levels of our data model.

- **description:** a human explicit description about the PoI.

- **category:** this entity classifies PoI into a category. For example, it can be a primary attribute (e.g., museum, bar, restaurant), a popularity ranking, or a security rating.

- **time:** time is considered the most common context information, which is generally represented by the time instant that the location was acquired. Time is also used to estimate the duration of an object at a place [24].

- **link:** this entity is a generic manner to represent a relationship from a

PoI to another PoI, or from a web resource to a PoI, both based on the RFC 4087 technique (point-to-point link).

- **metadata:** in this entity, we can insert formal metadata to the PoI (by reference, for example).

Therefore, we have used this definition to construct our data representation, which is presented in the next section.

2.1 Data representation

In our work, we assume that the interests of a user for a specific place, segment or trajectory can represent a user routine. For instance, a user likes to eat at the restaurant X everyday, where this restaurant is a point of interest. In the same way, a user prefers to take a specific street (segment) or a set of different segments (trajectory) to go from home to work. Along this line, a user routine can be defined following a multi-layer representation (see Figure 5.5), where n represent the identifier of each element of the routine, and the links are the relations between these elements at different abstraction levels.

Taking into account the representation presented in Figure 5.5, we classify user routines as Trajectory of Interest (ToI), Segment of Interest (SoI) and PoI at different abstraction levels. Therefore, we define this spatial information to be a multi-layer data representation in order to support the description of the user's daily routine.

According to Figure 5.5, a user routine is presented based on its layer. For instance, the last layer (*Layer 3*) can be represented by the name of the location according to the GPS coordinate (e.g. bakery's name, house number, etc.), based on the PoI data model proposed by W3C working group (with the same entities). Nevertheless, we inserted the entity called *user_id* to identify the owner of the PoI. In parallel, we reused and adapted the PoI data model to define the entities and values of SoI and ToI.

Following our multi-layer data representation and the reference model of W3C, the *Layer 2* is defined as the Segments of interest (SoI) that compose the user trajectory, where each SoI is composed of any number of the following entities:

- **user_ID:** is used to identify the owner of SoI.

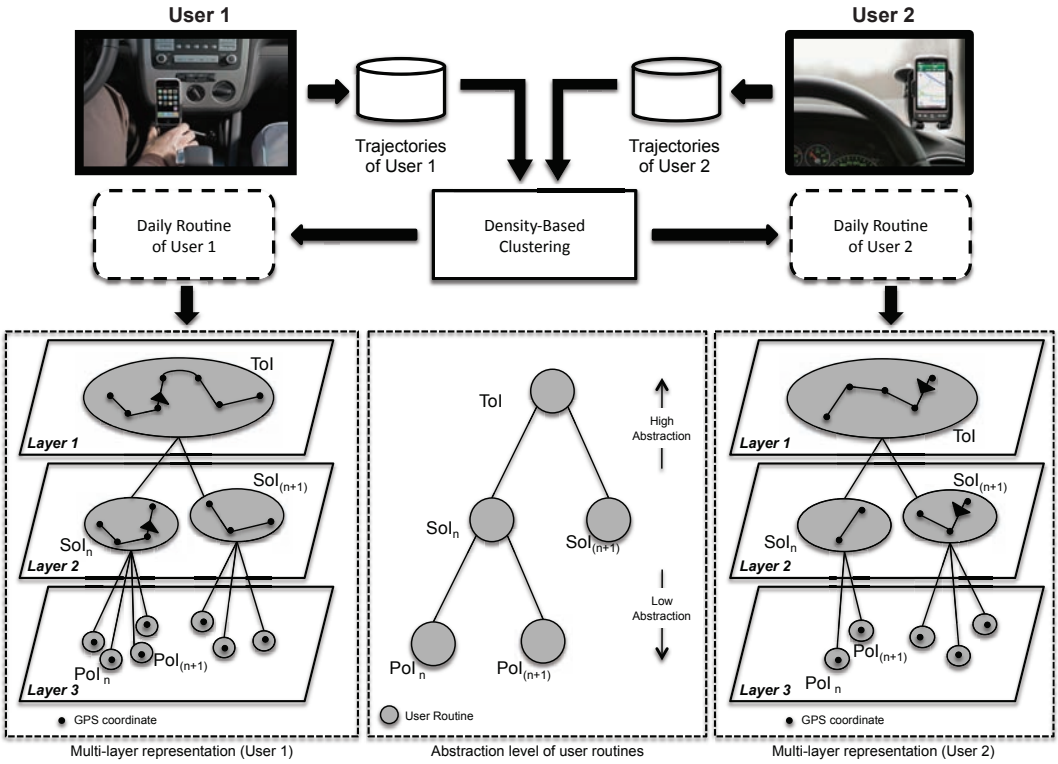


Figure 5.5: Our multi-layer data representation.

- **label:** is a human explicit label to name SoI, which can be generated by using the labels of PoI (e.g., from Work to Bakery).

- **description:** a human explicit description about the SoI.

- **category:** the classification of SoI into a category. Similar to the category of PoI, it can be a primary attribute (e.g., street, avenue, highway), a popularity ranking, or a security property.

- **time:** for this entity, we can have the time interval that the moving object stayed into SoI, based on the initial and final time instants. These time instants are derived from the time instants of the corresponding initial and final PoI's of the segment.

- **link:** similar to the PoI, this entity is a generic manner to represent a relationship from a SoI to another SoI, where the last PoI of the segment has a link with the initial PoI of the next segment.

- **metadata:** is the use of a formal metadata to SoI (by reference, for example).

Finally, ToI in the *Layer 1* could be represented by a whole user trajectory (e.g. to go from home to work). Therefore, we identify the following entities that compose each ToI:

- **user_ID:** is used to identify the owner of ToI.
- **label:** is a human explicit label to name ToI, which can be also generated by using the labels of PoI (e.g., from Work to Home) or by the labels of SoI (e.g., from street X to avenue Y).
- **description:** a human explicit description about the ToI.
- **category:** the classification of ToI into a category. Similar to the category of SoI, it can be a primary attribute (e.g., name of the region that the whole trajectory was registered), a popularity ranking, or a security property. The main characteristic for the security property is related to the access control polices for a user trajectory. Based on the level of the relationship with another user, the user can control the sharing of the whole trajectory (e.g., Public, List_of_Group_Access (specific group of friends in my social network), Private or List of users). While this property can be defined by the user in ToI, it can be also defined in the security properties of SoI and PoI.
- **time:** the time interval that the moving object stayed into ToI, based on the initial and final time instants. Similar to the time entity of SoI, these time instants are derived from the time instants of the corresponding initial and final PoI's of the trajectory. In addition, with this information we can identify the period of the day and the days of the week, for example.
- **metadata:** is the use of a formal metadata to ToI (by reference, for example).

Based on this structure, a user routine is presented as a general interest according to the abstraction level of the user/system. Besides that, a ToI is directly related to a set of SoI's and/or PoI's at low levels. To better

understand this relation, we use a tree structure to show the relation between each information according to the multi-layer data representation.

Based on the illustrated data representation, we design our multi-layer data model for trajectories, taking into account different abstraction levels of user routines. In the following we provide the basic definitions to support our discussion.

1. **Trajectory (T)** is defined as a set of consecutive points captured through a GPS of one trip performed by a user. Each location (L) is composed of a set of information (latitude, longitude, altitude, direction, time stamp for each registered point (t_L) and an approximate speed provided by the GPS). $T = \{L_1, L_2, L_3, ..., L_n\}$, the time interval between two points is computed by the subtraction of $t_{L(k+1)} - t_{L(k)}$, where $(1 \leq k < n)$. This temporal information also allows the recognition of pause instants, according to the proposal of [24]. Although the points are characterized by latitude, longitude and altitude, we focus on points in 2D space (latitude and longitude) to represent the position of each user.

2. **User Routine (UR)** is defined as a human construct to represent a routine of a user based on his/her interest. UR typically denotes a user interest, where a user can identify an entire trajectory, segment of route (e.g. street name) or place (e.g., bakery X), according to the layers presented in Figure 5.5, typically represented by name and characterized by type, which may be used as a reference point or a target in a location based service request (e.g., route destination).

3. **Set of UR (SUR)** is defined as the set of user routines based on the abstraction level of multi-layer representation. The user routine of each abstraction level is defined according to its identifier (ur), such that $traj$, seg and poi represents the ToI, SoI and PoI respectively. Therefore, SUR is formed by a finite set and subset of user routines in different abstraction levels, e.g. $SUR = \{ur_{traj}\{ur_{(seg,1)}, ur_{(seg,2)}, ur_{(seg,n)}\}, ur_{seg}\{ur_{(poi,1)}, ur_{(poi,2)}, ur_{(poi,n)}\}, ... , ur_{(s-1)}\{ur_{(s,1)}, ur_{(s,2)}, ur_{(s,n)}\}\}$, where s represents the abstraction level. For instance, the set to represent a user trajectory in the campus of Joseph Fourier University is

$$SUR_{traj} = \{ \; Chemistry\ Street\{Grenoble\ Informatics\ Laboratory, CERMAV\ Laboratory\}, Piscine\ Street\{ENSIMAG\ Laboratory\}, Library\ Street\{Central\ Library, Mathematics\ Laboratory\}\}$$

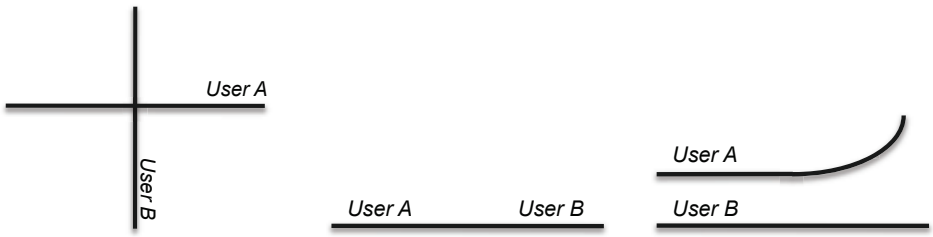
where *traj* can be represented by the user trajectory in the *Layer 1*, *Chemistry Street*, *Piscine Street* and *Piscine Street* are road segments in the *Layer 2*, and *Grenoble Informatics Laboratory*, *CERMAV Laboratory*, *ENSIMAG Laboratory*, *Central Library* and *Mathematics Laboratory* are local places in the *Layer 3*.

The intention to design a conceptual model is to offer basic procedures in order to support designers in the development of mobile social applications. A usual feature in the spatial multi-layer data model is the user routine corresponding to a given abstraction level (trajectory, road segment and local place).

When we consider that a graph of user interests is a tree, we can say that a user interest is associated with ur in different abstraction levels, which allows to indicate that a user interest belongs to the abstraction level s (*traj*, *seg* and *poi*) associated with ur. Since the multi-layer data representation is presented, we take into account the organization of objects for a defined abstraction level. Consequently, a low abstraction level offers the set of PoI's to describe a user trajectory at the highest abstraction level. We observe that for all user routine shown in the data representation, we may have a specific UR available at each abstraction level (s), such that $L \in ur_{poi}$. This representation offers a procedure to understand the set of abstraction levels.

Finally, since two users A and B have a relation in the social network, our data model allows the identification of similar user routines between them, taking into account the different situations, presented in Figure 5.6. We consider the representation of three main situations of social interaction between users.

For the first situation (Figure 5.6(a)), we observe that two users have a point of interaction at the crossing of two UR's (e.g. road segment). Assuming that a user A passes in a specific region (e.g., at campus of University of Grenoble) and the user B also passes at this campus, we cannot affirm that both users are sharing a location L in ur_{poi}. However, our data model provides



(a) Crossing. (b) Sharing a UI. (c) Nearing.

Figure 5.6: Three main representations of situations that we consider as similarities between two users.

a manner to identify this crossing in different abstraction levels. Since we identify common regions between both users, we can identify similar segments and points of interest, allowing the identification of similar routines in different abstraction levels.

In Figure 5.6(b) the point of interaction could be the complete set of PoI (e.g. all road segment or a part of it). For this example, when we identify that both users are sharing a street, it is not evident that they are sharing the same part of this segment. However, while the similar segment is identified, our algorithms verify if the locations represented in the PoI layer corresponds to the same part of the shared segment.

Finally, in Figure 5.6(c), the most important information is the proximity between users. Hence, this proximity can be determined according to each layer in our model. The user could define this proximity. Consequently, the users can consider a possible social interaction due to the proximity of their trajectories, segments or points of interest.

2.1.1 Multi-layer representation of correlated trajectories

As one or a set of user interests may describe a user routine, we need to consider every information of each abstraction level (*ToI*, *SoI* and *PoI*). We then define a user trajectory as a sequence of UR's, where the set of segments crosses between different abstraction levels in the required order. The following example presents a multi-layer representation in order to illustrate our approach.

- $set_{ToI} = \{ur_{traj}\}$

- $set_{SoI} = \{ur_{(seg,1)}, ur_{(seg,2)}, ur_{(seg,3)}\}$
- $set_{PoI} = \{ur_{(poi,1)}, ur_{(poi,2)}, ur_{(poi,3)}, ur_{(poi,4)}, ur_{(poi,5)}, ur_{(poi,6)}, ur_{(poi,7)}\}$

For instance, we can construct the following sets of UR (SUR):

- $SUR_1 = \{ur_{traj}\{ur_{(seg,1)}\{ur_{(poi,1)}, ur_{(poi,2)}\}\}\}$
- $SUR_2 = \{ur_{traj}\{ur_{(seg,2)}\{ur_{(poi,3)}, ur_{(poi,4)}, ur_{(poi,5)}\}\}\}$
- $SUR_3 = \{ur_{traj}\{ur_{(seg,3)}\{ur_{(poi,6)}, ur_{(poi,7)}\}\}\}$

The Figure 5.7 illustrates these sets of UR's related to each abstraction level. In the next definition, the user routine descriptor (D) contains the sequence of the determined user routines. For instance, we determine two different trajectory descriptors for user 1 (D_1) and user 2 (D_2):

- $D_1 = < ur_{(seg,1)}, ur_{(seg,2)}, ur_{(poi,7)} >$
- $D_2 = < ur_{(poi,1)}, ur_{(poi,3)}, ur_{(poi,5)} >$

We note that the descriptors can be composed by UR's at different abstraction levels due to multiple location names, which can be obtained from reverse geocoding services. Therefore, our data representation is also able to find a similarity although these UR's are at different abstraction levels. The concept of multi-layer representation is an important step to understand the relations and similarities between UR's, grouped in different user descriptors. For instance, if we consider D_1, the user describes a trajectory from a departure ur_{seg} (at the second abstraction level) to a destination in a ur_{poi} (at the third abstraction level). In case of D_2, the user describes his/her routine at the same level.

A multi-layer data representation should be able to identify the abstraction level of each UR. This data representation becomes an important element for providing the accurate information to identify the similarities between user routines. If we observe the previous trajectory descriptors and the three situations presented in Figure 5.6, we see some challenges to develop a data model at different abstraction levels. For instance, if we observe D_1 and D_2, we observe that the first user is passing in $ur_{(seg,2)}$ (at the second level) and the other user is passing in $ur_{(poi,3)}$ (at the third abstraction level). Therefore, our approach allows the identification of similar routines between users who are sharing UR's in different abstraction levels.

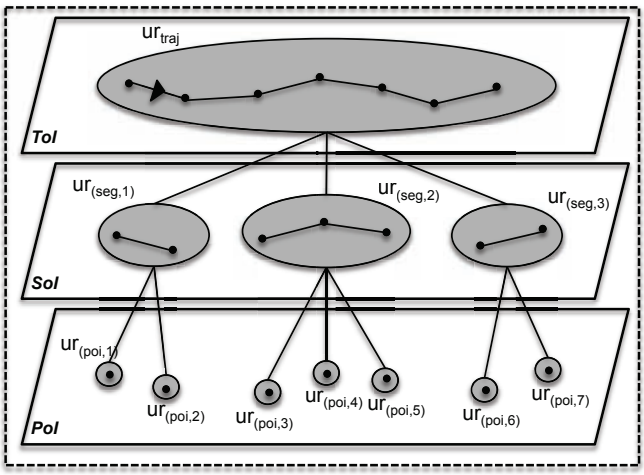


Figure 5.7: Example of multi-layer data representation.

2.1.2 Representation of temporal data

While the clustering algorithm processes the spatial information in order to identify the best representative trajectory for each user, the temporal information becomes relevant contextual information to enrich the services that are provided by our approach. Hence, we designed a data representation of temporal information, which is detailed as follows.

Our approach follows the temporal representation presented in [148], where the time is processed after identifying the spatial similarities. Making use of the best representative trajectories, we obtain multiple information of time for each position in the user's trajectory. Figure 5.8 illustrates an example of a best representative trajectory with intermediary locations.

Assuming that this best representative trajectory was obtained by a dataset of *10* trajectories of a user to go everyday from home to work. Consequently, we have *10* working days for this example. The clustering algorithm then discovers that the user recorded *7* similar trajectories, by passing at the same streets and near to specific locations. Intuitively, we note that this user registered different time instants by location (illustrated by the points in the trajectory). We can see these different time instants in Table 5.1.

In Table 5.1, we may deduce that the user have traveled for three different trajectories in three working days to go from home to work. These days are *Day 2*, *Day 6* and *Day 10*. In contrast, we have seven trajectories that were recognized to construct the best representative trajectory. Given the

Figure 5.8: Example of a best representative trajectory with multiple locations between a departure (Home) and a destination (Work).

Locations	Day 1	Day 3	Day 4	Day 5	Day 7	Day 8	Day 9
Home	08:00	08:10	08:05	08:07	08:15	08:12	08:17
Bakery	08:07	08:18	08:12	08:15	08:23	08:20	08:25
Supermarket	08:15	08:26	08:20	08:23	08:30	08:28	08:32
Restaurant	08:25	08:36	08:29	08:31	08:37	08:35	08:39
Post office	08:32	08:43	08:36	08:38	08:42	08:41	08:45
Work	08:40	08:50	08:43	08:45	08:50	08:47	08:52

Table 5.1: Time instants by location from a best representative trajectory of a user.

Supermarket as the location, we see that the user passed close to it at *08:15* in the first day, at *08:26* in the third day and at different time instants in the other 5 days.

Taking into account this example, we designed our representation of temporal data, where the key idea is to store all the time instants by location and represent them in a time interval. The time interval specifies all the time instants that the user passed close to each specific location. Finally, this data can be used to enrich the information that will be provided by our approach.

3 The trajectory correlation algorithm to identify similar interests between users based on user's daily routines

Taking into account the idea to analyze user's daily routines in order to increase the number of social interactions between users, we propose an opti-

mized algorithm based on Minimum Bounding Rectangles (MBR) [181] and the Hausdorff distance [182].

The Hausdorff distance is often used to determine the similarity of two shapes [183] and to measure errors for approximating a surface in generating a triangular mesh [184]. In our approach, we are interested to use Hausdorff distance computation in two different cases. Basically, the first case is applied when the algorithm finds a correlated area between two MBR's. It uses Hausdorff distance to compute the distance between the points that are in the correlated area. On the other hand, if there is no correlated area, the Hausdorff distance computation is used to compute the distance of near points between two MBR's. When the distance of two MBR's is found, the algorithm allows the expansion of both MBR's in order to find one or more points of social interactions, taking into account a threshold (D_{max}) for the expansion.

Firstly, we identify four extreme points of each trajectory (the northernmost, the southernmost, the westernmost and the easternmost). With these points, we create the MBR for the users' trajectories. Figures 5.9 illustrates the MBR for a specific trajectory.

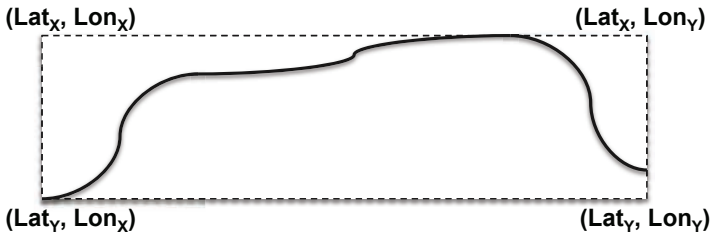


Figure 5.9: An example of MBR.

For instance, we consider two users A and B and the existence of MBR's for their respective trajectories. The four points to represent the rectangle of the user A are:

$$(Lat_{max(A)}, Lon_{min(A)}), (Lat_{max(A)}, Lon_{max(A)}),$$
$$(Lat_{min(A)}, Lon_{min(A)}), (Lat_{min(A)}, Lon_{max(A)}).$$

The points for the user B are:

$$(Lat_{max(B)}, Lon_{min(B)}), (Lat_{max(B)}, Lon_{max(B)}),$$
$$(Lat_{min(B)}, Lon_{min(B)}), (Lat_{min(B)}, Lon_{max(B)}).$$

Furthermore, we execute the trajectory correlation process according the algorithm as follows.

Algorithm 2 Main algorithm.

Input: two trajectories of users A and B with the points containing their respective coordinates.

Comment: *It is verified if the two MBR's does not have a correlated area.*

if $(Lat_{max(A)} < Lat_{min(B)})$ **or** $(Lat_{max(B)} < Lat_{min(A)})$ **or** $(Lon_{max(A)} < Lon_{min(B)})$ **or** $(Lon_{max(B)} < Lon_{min(A)})$ **then**
  Execute **HausDist** of MBR(A) and MBR(B);
  if HausDist $< D_{max}$ **then**
    Expand MBRs;
    Restart main algorithm;
  else
    There is no correlated area;
    Stop main algorithm;
  end if
end if

Comment: *Otherwise, we select the correlated area and execute the HausDist algorithm.*
Select correlated area (Alg. 3);
Execute **HausDist** (Alg. 4);

Output: The points in the correlated area and the distances between the points of A in relation to the points of B.

As we can observe in the main algorithm, when there is no correlation between two MBR's, we execute an algorithm to compute the Hausdorff distance between two MBR's. The main reason to carry out this algorithm is related to the problem involving extreme points in the MBR faces. For example, we have a point in the right face of the MBR(A) and another point in the left face of the MBR(B). Although the MBR(A) is close to the left face of MBR(B), there might be no intersection, as presented in Figure 5.10. Then, we might

have a problem, because two near points are not present in the correlated area.

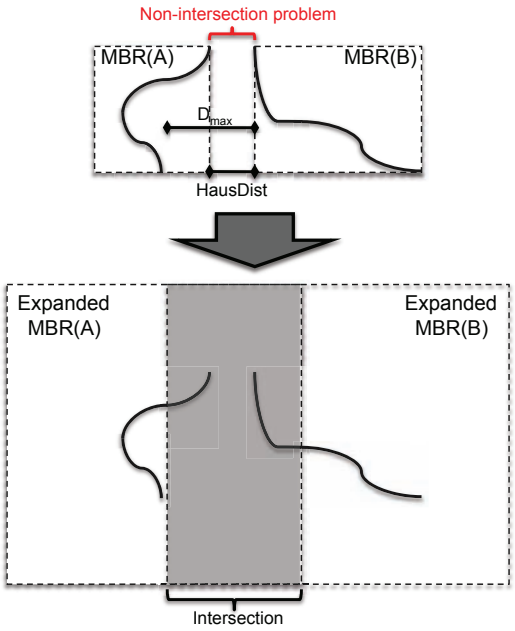


Figure 5.10: MBR Expansion for the non-intersection problem.

To solve this problem, we propose a MBR expansion algorithm, which computes the Hausdorff distance of two MBR's in order to verify if the expansion is possible or not according to the threshold D_{max}. The Hausdorff distance from the MBR(A) to the MBR(B) can be determined by exploiting the characteristic for each MBR area, there has to be at least one object that touches it. Therefore, we identify the area in MBR(A) closest to a face in MBR(B). After that, the algorithm computes the Hausdorff distance (HausDist) of these two faces and compare the result with D_{max}. If HausDist is less than D_{max}, then both MBR's expands their related areas from the current distance to the result of D_{max}. Figure 5.10 shows the MBR expansion process for the no intersection problem.

On the other hand, if there is an intersection of MBR's, the algorithm 3 is executed in order to determine the correlated area.

Since the correlated area of MBR's is found, the main algorithm executes the Hausdorff distance computation of the points. Assuming that a and b are points of sets A and B respectively and that they are in the correlated area, then the Algorithm 4 is executed.

Algorithm 3 Selection process

Input: $Lat_{min(A)}$, $Lat_{max(A)}$, $Lat_{min(B)}$, $Lat_{max(B)}$
if $Lat_{max(A)} > Lat_{max(B)}$ **then**
  Select $Lat_{max(B)}$
else
  Select $Lat_{max(A)}$
end if
if $Lat_{min(A)} > Lat_{min(B)}$ **then**
  Select $Lat_{min(A)}$
else
  Select $Lat_{min(B)}$
end if
if $Lon_{max(A)} > Lon_{max(B)}$ **then**
  Select $Lon_{max(B)}$
else
  Select $Lon_{max(A)}$
end if
if $Lon_{min(A)} > Lon_{min(B)}$ **then**
  Select $Lon_{min(A)}$
else
  Select $Lon_{max(B)}$
end if
Output: correlated area

Algorithm 4 Hausdorff distance algorithm

Input: points of trajectories A (a_i such as $i = 1$ to n) and B (b_j such as $j = 1$ to m), where n and m are the total of points in the trajectories A and B respectively.

HausDist = 0
for all point a_i of A **do**
  shortest = Inf ;
  for all point b_j of B **do**
    $distance_{ij}$ = distance (a_i , b_j)
    if $distance_{ij}$ < shortest **then**
      shortest = $distance_{ij}$
    end if
  end for
  if shortest > HausDist **then**
    HausDist = shortest
  end if
end for

Output: the shortest distance of a point in the trajectory A and another point in the trajectory B.

While the similar user routines are identified between two best representative trajectories, the algorithm starts the comparison between temporal information. To compare the temporal similarities between users, we consider all the similar locations identified. We have adopted the Parzen-window method [185] to identify temporal similarities by location. Parzen-window has been used in a large number of research areas, such as pattern recognition, data classification, image processing and tracking. We decided to use the Parzen window due to the well representation of each time instant at the time interval, where the density of the points can be easily recognized and visualized in the graph.

By definition, the Parzen-window is a density-based estimation that considers the data-interpolation technique [186]. Assuming that we have a random variable (x), then this technique computes the probability density function (PDF) in which the random variable was derived. In summary, it superposes kernel functions at each observation (x_i). Hence, the PDF $(f(x))$ of the Parzen-window is computed by

$$f(x) = \frac{1}{n}\sum_{n=1}^{n}\frac{1}{h_n^{dim}}K\left(\frac{x-x_i}{h_n}\right), \tag{5.1}$$

where $K()$ is the kernel function, dim is the dimensional space and h_n is the window width. Based on this equation, we are able to compute the value of $f(x)$ at a certain location (point). Along this line, we can determine a window function at x and define the total of observations xi that are close to the window.

For our approach, we determined the Gaussian PDF as the kernel function for Parzen-window density computation. Thus, the PDF $f(x)$ with the Gaussian function becomes

$$f(x) = \begin{cases} \frac{1}{n}\sum_{k=1}^{n}\frac{1}{(h\sqrt{2\pi})^{dim}}e^{\left(-\frac{1}{2}\left(\frac{x-x_k}{h}\right)^2\right)} & \text{if } t_b < x < t_e \\ 0 & \text{otherwise,} \end{cases} \tag{5.2}$$

such as t_b is the initial time instant and t_e is the final time instant within the time interval for each location. As we are analyzing a point in comparison to another points in the time interval, the value of $dim = 1$. An important element related to the use of Parzen-window is the value of the window size (h). According to [187] and [188], when the Gaussian kernel is being used, the

optimal value of h is defined by

$$h = \left(\frac{4\sigma^5}{3n}\right)^{\frac{1}{5}}, \tag{5.3}$$

where n is the number of time instants in the time interval and σ is the standard deviation of the samples.

Therefore, we can obtain the frequency that the user is near to a certain location. Since we have identified a similar routines between two users, we can compare the temporal graphs to know the probability of rendezvous between them at a certain period of time. Taking into account the example of the supermarket (Table 5.1) for a user A, we construct a time interval between 08 : 15 and 08 : 32, with the time instants [*08:15, 08:20, 08:23, 08:26, 08:28, 08:30, 08:32*]. Then, the PDF of the Parzen function then generates the graph presented in Figure 5.11 in order to represent all the time instants that user A passed near to supermarket in these seven days.

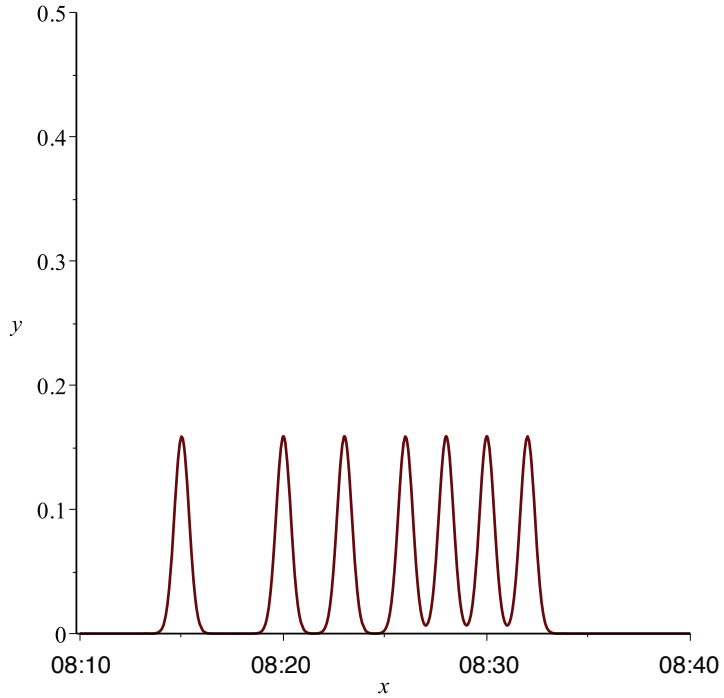


Figure 5.11: Time instants that user A have passed near to supermarket in the 7 days.

As we observe, the graph shows the probability of each time instant that user A was near to supermarket in the interval. Next, we assume that another

user B (who is friend of A) have also passed near to the same supermarket in other ten days. Given a time interval of user B between 08 : 10 and 08 : 50, with the time instants [*08:10, 08:15, 08:16, 08:16, 08:20, 08:21, 08:30, 08:40, 08:42, 08:50*], the PDF of the Parzen function generates the graph presented in Figure 5.12.

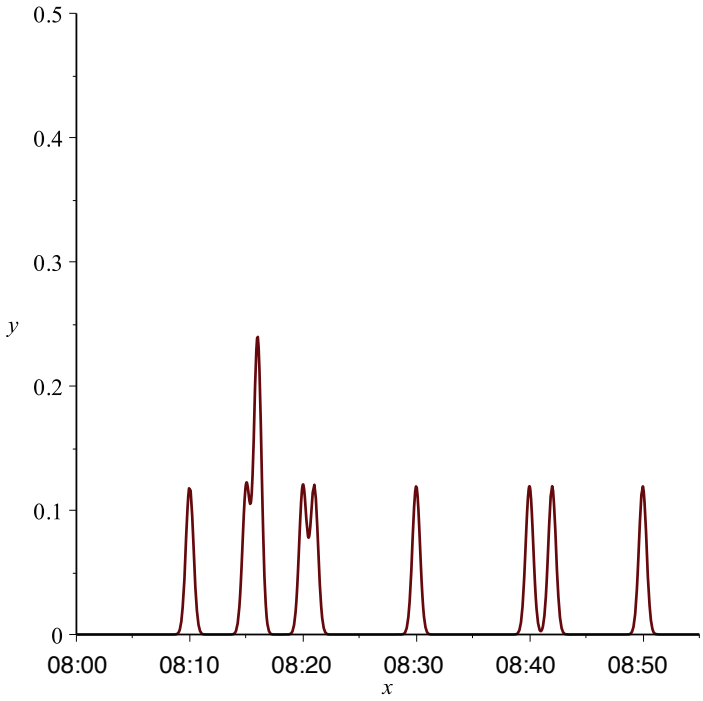


Figure 5.12: Time instants that user B have passed near to supermarket in the 10 days.

Intuitively, we observe that these graphs can represent all time instants in which both users have passed near to each location. Since we discover similar routines between users A and B, we can use these graphs to estimate the rendezvous between them, considering the temporal similarity. One way to compare these graphs is through the superposition, by observing the common areas. Another manner is to compute the probability from the highest value of time instant to the lowest value.

4 Sharing routines between users

A well-known solution of Web applications that involves sharing and estimation of user interests is called recommendation system. In general, recommendation systems are classified in two groups, which are content-based and collaborative filtering systems [165]. In terms of content-based systems, a recommendation is performed based on the user preferences in relation to a specific content. For example, if a user prefers to listen country music than the other genres, the system recommends new songs having the "country" genre as the preference for that user. On the other hand, collaborative filtering systems recommend some information based on similar features of users and/or data. This kind of system is commonly used to recommend information that is preferred by a group of similar users.

Taking into account the characteristics of our approach, we have considered the collaborative filtering system as the best method to share the routines between users. These routines are represented by the similar user interests, which are identified by the trajectory correlation algorithm. For example, the recommending system is able to answer the question about a friend who is passing into the campus of the University of Grenoble during the week. Therefore, the collaborative filtering system verifies the user routines (in terms of spatio-temporal information) of a group of users to identify similar interests between them

Following the steps of our approach the data sharing algorithm can send a message to the user alerting that a friend passes in front of a specific number of the street X all the weekdays between 10:00 AM and 10:30 AM. This message can also contain accurate information about distance, which is acquired by the Hausdorff distance algorithm.

The final part of our approach is the data-sharing algorithm, which enables the generation of an enriched information based on the processed data. It reads all the fields related to a correlated point in order to automatically create the message that will be sent to one or both users. Figure 5.13 shows the creation of a message by using context information, which will be sent to the user B about a possible point of social interaction with the user A.

The data-sharing algorithm can be applied to several types of applications, for example: mobile social applications, social networks, SMS, and others. Besides that, our proposal allows the inclusion of a color-based scheme for

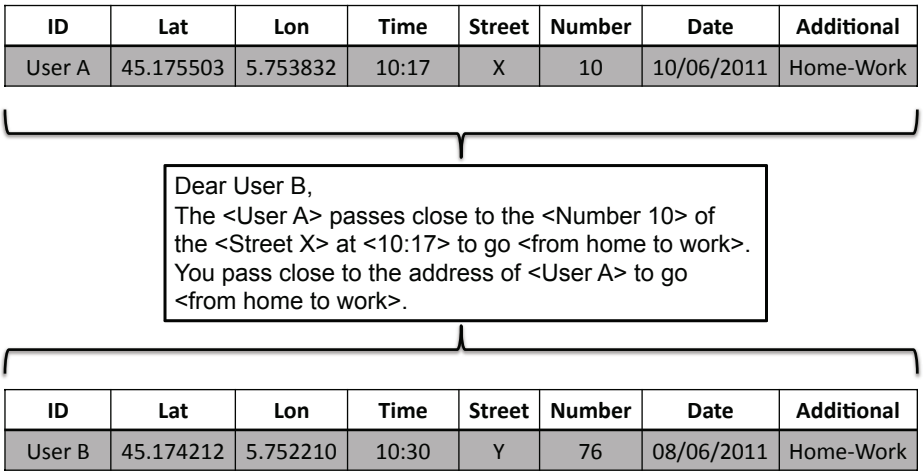


Figure 5.13: The context information of a correlated point in the database of the user B about the user A.

the visualization of potential points of interaction, taking into account the probability of interaction among users. Finally, in the next chapter, we present the evaluation of our approach, taking into account different scenarios.

5 Conclusion

Virtual community platforms provide solutions to social connectivity, giving people the capability to share interests, opinions, and personal information with other users. Nevertheless, we argue that the absence of context-aware mechanisms in virtual communities could be one of the main reasons that social interactions are frequently missed. The users' daily routines, therefore, can be captured by mobile social applications and shared in virtual communities in order to improve the social connections in real communities.

In this chapter, we introduced our location-based approach to identify similar interests between users in social networks (LIDU). The key idea is to provide a layer of services to acquire daily routines in order to find near points and, consequently, increase social interactions in real communities.

We presented a flexible multi-layer data model for mobile social application context based on user routines. We designed a conceptual view to be adaptable and acceptable to a set of generic features as well as to assist developers in designing solutions with the inherent complexity of trajectory semantics

(spatio-temporal data). Besides that, we discussed how our data model could offer mobile social applications with direct support for trajectories. Next, we presented an algorithm to execute the trajectory correlation process based on Minimum Bounding Rectangles (MBRs) and the Hausdorff distance (Haus-Dist) for finding spatial similarities. Furthermore, we used Parzen-window technique to identify similarities of temporal data.

To validate our Approach, we implemented and tested a mobile social application for tracking daily routines. Additionally, we developed a plug-in on a virtual community platform to receive the user profiles and to execute the trajectory correlation algorithm. Our results are presented in the next chapter.

Chapter VI

Evaluation of our approach

Contents

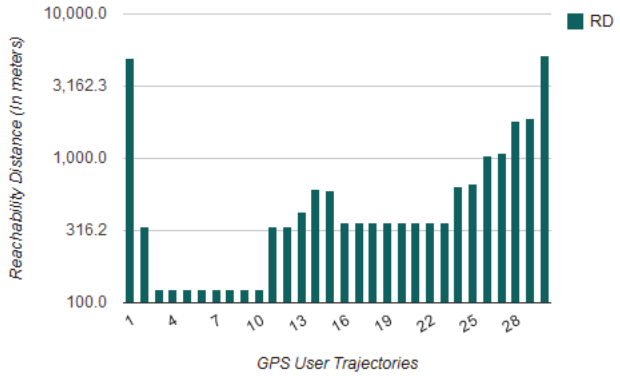
In this chapter, we present the results obtained by the evaluations of our approach in different scenarios. These evaluations were divided in three parts, which are: trajectory data acquisition, clustering algorithm and trajectory correlation algorithm. Since the main scientific contributions of this thesis are related to the clustering and trajectory correlation algorithms, we start presenting these algorithms. After that, we present the mobile application that was developed to perform the trajectory data acquisition process. In the following sections we present these parts and discuss the results obtained in each evaluation.

1 Clustering algorithm

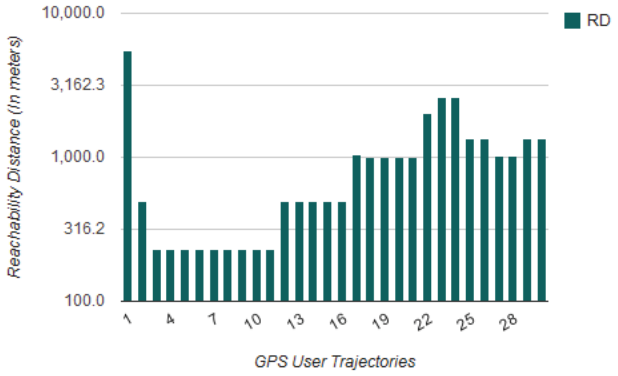
To demonstrate the efficiency of the clustering algorithm we have applied our approach to two separate users, based on their registered trajectories in Dublin, Ireland. The overall approach can be summarized in three steps. First of all clustering is applied to individual user trajectories over a period of one month. A user's daily routine is a trajectory from home to work. After obtaining distinct groups an aggregated trajectory has to be chosen.

With the help of visualization and aggregation techniques, a best representative trajectory for each user is obtained. This aggregated trajectory

obtained from several user trajectories is then compared to other users by applying our trajectory correlation algorithm. This will enable groups of users to share similar routes to increase geospatial social interaction. We now explain the different input parameters we have used in order to verify the results.



(a) User 1 ($\epsilon = 1000$ & minNbs $= 3$).

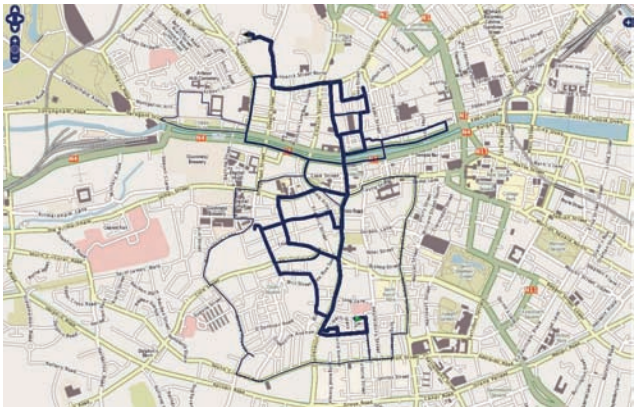


(b) User 2 ($\epsilon = 1000$ & minNbs $= 3$).

Figure 6.1: Reachability plots showing clustering structure.

OPTICS clustering algorithm requires two input parameters: distance threshold (ϵ) and minimum neighbors *(minNbs)*. The authors of OPTICS [1] suggest that the value of these two parameters have to be large enough to yield good results. We structured our experiment in a way that we choose a range of distance threshold values as well as minimum neighbors. For our scenario, we defined the distance threshold between 1000 meters and 15000 meters $\Rightarrow$ *(1000* $\leq \epsilon \leq$ *15000)*. Similarly, for minimum neighbors we selected a value of 1 up to 10 $\Rightarrow$ *(1* $\leq$ *minNbs* $\leq$ *10)*.

The experiment was run with a combination of values for both parameters.



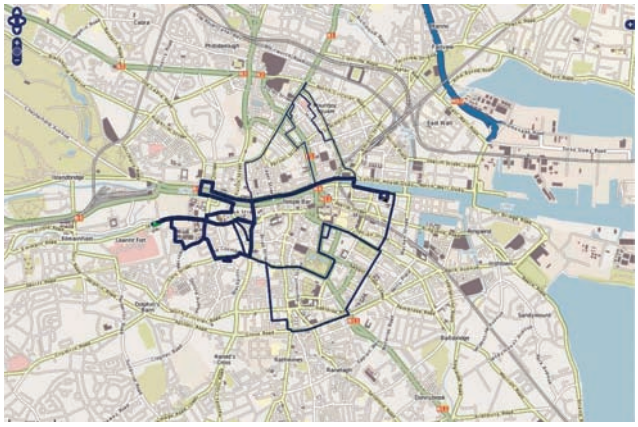
(a) Three clusters showing distinct routes of User 1 (overlay on map).

(b) Three clusters showing distinct routes of User 1 (without overlay).

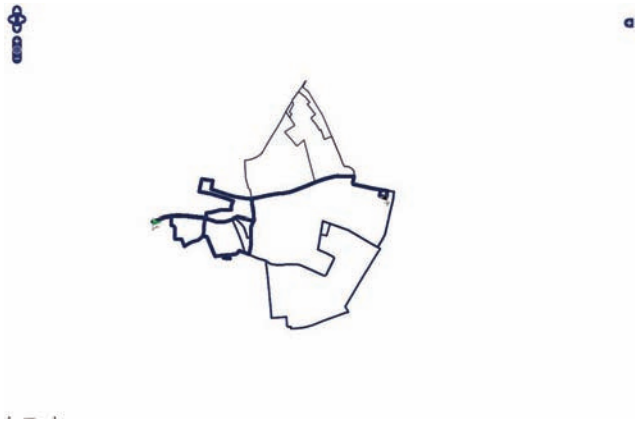
Figure 6.2: Clusters of user 1.

Based on the statistics and a range of reachability plots we obtained, we found the best combination of values $\Rightarrow$ *(ϵ = 1000 & minNbs = 3)*. This condition revealed a satisfactory result in terms of the clustering structure from the reachability plots.

The reachability plots obtained are illustrated in Figures 6.1(a) and 6.1(b). The plots show re-ordering of objects (trajectories in the dataset) on x-axis while y-axis demonstrates the reachability distances between trajectories. Automatic cluster extraction techniques from a graph were presented in [1][189]. This data independent visualization provides analysts a high-level understanding of clustering structure. From these graphs clusters can be identified based on Gaussian-bumps or valleys. As a general rule the cluster starts from a



(a) Three clusters showing distinct routes of User 2 (overlay on map).



(b) Three clusters showing distinct routes of User 2 (without overlay).

Figure 6.3: Clusters of user 2.

steep-down area and ends at a steep-up area.

Based on the first plot in Figure 6.1(a), we can clearly see that there are two dominant clusters in user trajectories (trajectory 2 to 13 and trajectory 14 to 25) shown by the valleys in the plot. The other cluster is a group of trajectories, which does not specifically form a valley however they are grouped together into one cluster. The second graph (see Figure 6.1(b)) also shows three clusters with varying cardinalities (trajectory 2 to 16, 17 to 22 and 23 to 30). In both the graphs, the first trajectory is considered as noise (see OPTICS algorithm [1]).

In Figures 6.2(a), 6.2(b), 6.3(a) and 6.4(b), the three clusters (from both

graphs) are drawn in different styles. The representative routes for each cluster are drawn with different thickness for visualization purposes.

(a) Best representative aggregated user trajectories (user 1).

(b) Best representative aggregated user trajectories (user 2).

Figure 6.4: Best representative trajectories of users 1 and 2.

The clusters show three distinct routes both users adopted over a period of one month to travel from home to work. On average each user trajectory contains almost 100 points. The clustering structure also forms distinct groups based on a specific route on a specific day of the month. For example in Figures 6.2(a) and 6.2(b), cluster 2 holds trajectories starting from trajectory *14* to trajectory *25* that include *11* days routes. For this specific case we can acquire knowledge about the patterns related with a particular day of a week or a month. For example, if we observe the order in which the trajectories were recorded in case of cluster 2 we obtain *(1,2,3,4,7,8,9,12,13,14,15)*. We can apply heuristics and visualization techniques such as heat maps in order to gain more insights into user behaviors. As apparent from the above sequence user 1 always follows a similar or close route during at least three consecutive days of a month such as *(1,2,3)*, *(7,8,9)* and *(13,14,15)*.

After analyzing the clustering structure the next step is to find an aggregated trajectory or a best representative of a particular user route. For this purpose we have applied a simple yet interesting visualization technique. When all three clusters from both users are visualized using a single grey scale color scheme, it reveals the most frequent route adopted. The color has to

be selected in a way that it must be transparent enough to visualize these changes. The phenomenon is illustrated in Figures 6.4(a) and 6.4, where user 1 and user 2 best representatives can be visualized and extracted respectively for further analysis.

2 Trajectory correlation algorithm

Since the clustering algorithm recognizes the best representative trajectory for each user, the trajectory correlation algorithm is executed. For this example, the algorithm firstly generates the MBRs for each best representative user trajectory and identifies the correlation between both MBRs. After that, it computes the Hausdorff distance of the points in the correlated area.

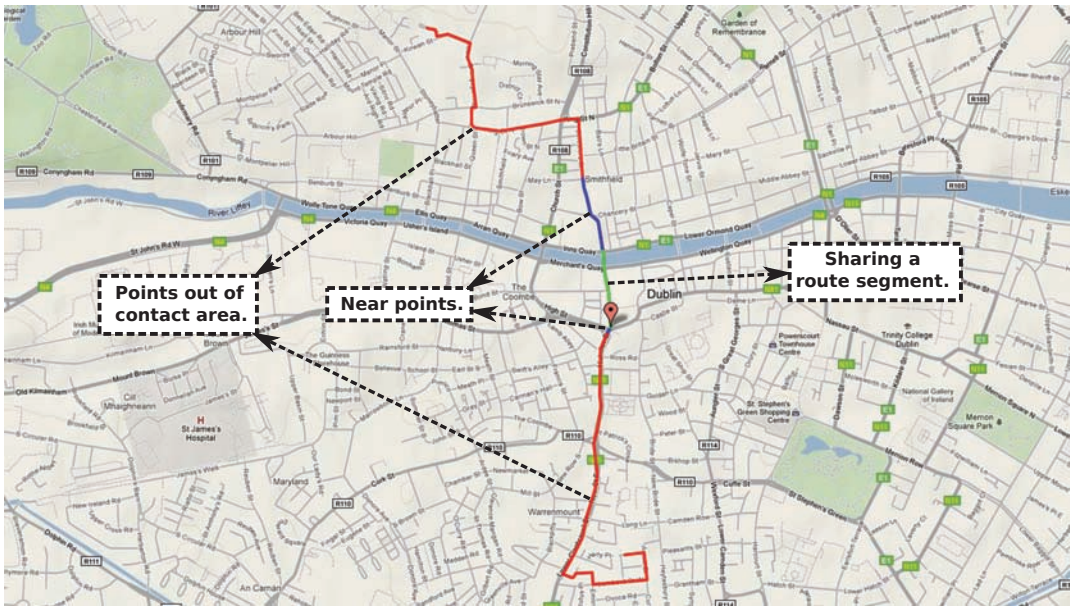


Figure 6.5: Best representative trajectory of user A in comparison to user B.

In order to present the accuracy and efficiency of our system we used a color-based scheme to represent the points in the same road segment, the near points and the points out of the correlated area. Figures 6.5 and 6.7 show the trajectory of the users A and B respectively with the colors representing the near points between them. The green color represents the same segment that is used by both users for their daily routines. The blue color denotes the possible points of interaction, which is in the correlated area among the

MBRs. Finally, the red color indicates the points that are out of the correlated area. Additionally, the system allows the generation of messages making use of the context information.

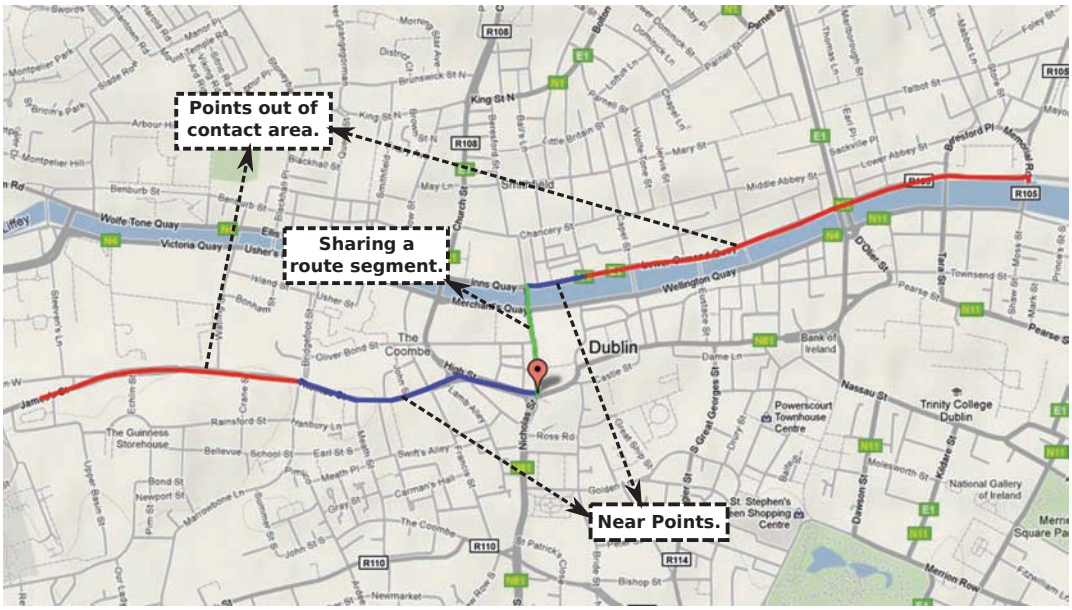


Figure 6.6: Best representative trajectory of user B in comparison to user A.

Based on the results, we observe that both analyzed users have common interests and our algorithm was able to identify the similar routines between them. These similarities are presented according to the situations described in the last chapter. Taking into account the different abstraction levels of our data model, these results illustrate the common segments of interest (SoI) between two users. This is possible due to the use of enriched information that is associated with each location in the database. In other words, each coordinate is registered in the database with its associated context information (e.g., postal address, time, speed of the moving object, weather, etc). Therefore, this enriched information facilitates the identification of similar segments and comes as an additional feature to increase the accuracy of the final result.

These results of our correlated trajectory algorithm are associated with two trajectories containing user routines at the same abstraction level of our multi-layer data model (see Figure 5.5). However, our algorithm also allows to identify similar user routines in different abstraction levels. That is possible due to our top down processing to find the similar interests between

two trajectories. Firstly, we compare the highest abstraction levels of both users, taking into account the region around each trajectory. Since we find the correlated regions of both trajectories, we perform the comparison in the next layer for finding similar road segments between users' trajectories, which allows to obtain more details about the type of similarity (e.g. near, sharing). Finally, we carry out the comparison at the lowest abstraction level in order to find similarities between local places, such as: bakery X, hospital Y, and others.

Figure 6.7 illustrates the same comparison, but at a different (less detailed) abstraction level. The routine of user **B** is *Grenoble*, since his/her whole trajectory is within Grenoble (Level 1 of our data model). On the other hand, the routine of user A is represented by road segments (Level 2 of our data model). Based on that, the trajectory algorithm finds the similarities between the routines of user B (at the level of Trajectory of Interest (ToI)) in comparison to the routine of user **A** (at the level of Segments of Interest (SoI)). As the routine of user **A** is a subset of the set of the ToI represented by *Grenoble*, the map is shown with a green dot over *Grenoble*. Figure 6.7 presents an example of how a multi-layer data model can provide information at different abstraction levels.

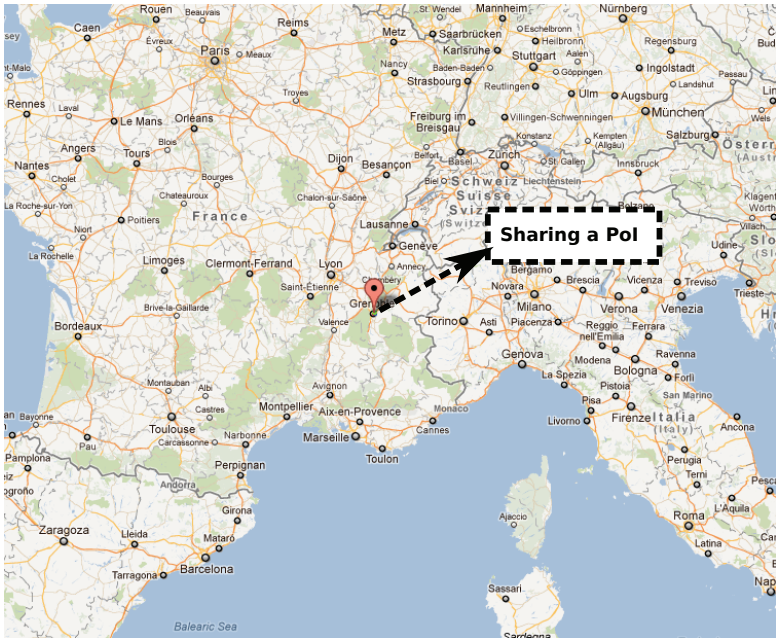


Figure 6.7: Best representative PoI (Grenoble) of user B in comparison to user A at a different abstraction level.

Since the similar user routines are identified, we can process the similarity analysis in the temporal data, comparing the time intervals in which the users have passed in a specific location (as presented in Chapter V). With the final results, we can provide complete information about users' similarities to the applications.

3 Trajectory data acquisition

To evaluate the efficiency of our trajectory data acquisition in a real situation, we implemented our proposal for the ZeroCO2 project [190]. We designed our system to be a digital logbook during a boat expedition around the Mediterranean Sea. The logbook, which was created as a book to record readings from the ship log [191], is an essential instrument to the navigation and has to be used daily. In general, the crew uses paper-based logbooks to register all information and, frequently, the information is collected from distinct equipments. Hence, we concluded that our system was able to create a complete logbook for this boat expedition. In addition, the challenging scenario of the sea added some problems involving the recurrent absence of Internet connection and the lack of battery charging.

Our system was responsible to track the trajectory followed by the boat, adding all context information to each registered coordinates. Although our system proposes the use of audio, video and photo as data, we used only photos for this first experiment in the project ZeroCO2. Taking into account this scenario, we face new challenges that have motivated us to improve the context-aware system proposed in the previous section.

The mobile application interface is shown in Figure 6.8. As we can observe, there are two main functions: the tracking mechanism and the digital camera. The tracking mechanism is responsible for registering the geographic coordinates to construct the trajectory. The interface shows the position, speed, date and, if Internet connection is available, wind speed and humidity. The digital camera takes a picture and, automatically, adds the context information to it. There is also the option "Tag" with which you can add the information manually.

An important result discovered during our tests is related to the use of metadata following some standard, such as Web Ontology Language OWL

(a) Tracking mechanism. (b) Digital camera.

Figure 6.8: Tracking mechanism and digital camera.

[192]. Several solutions adopt this language to obtain inferred information about a context. However, it needs to add a large number of information in the metadata file to perform this task. Consequently, the mobile application generates several unused information into the metadata file, causing some problems of memory overflow in the mobile application. Therefore, we optimized the content of our metadata files registering only the relevant information. Besides that, we developed our own local parser to get the information of each tag and to infer about context information using the HTML parser.

The first evaluation was done during a travel around the Marseille coast. We ran this first test to calibrate the distance filter option and to execute the performance evaluation in the mobile phone. This option is responsible to define the detail level of the trajectory. We assigned the value fifty meters to the distance filter, which means that a position will be registered if it is higher than fifty meters in comparison with the last position registered. With the first results, we refined our system to the second test: a travel from Marseille to Ajaccio (Corsica Island).

Figure 1.8 shows the performance evaluation of our application in the mo-

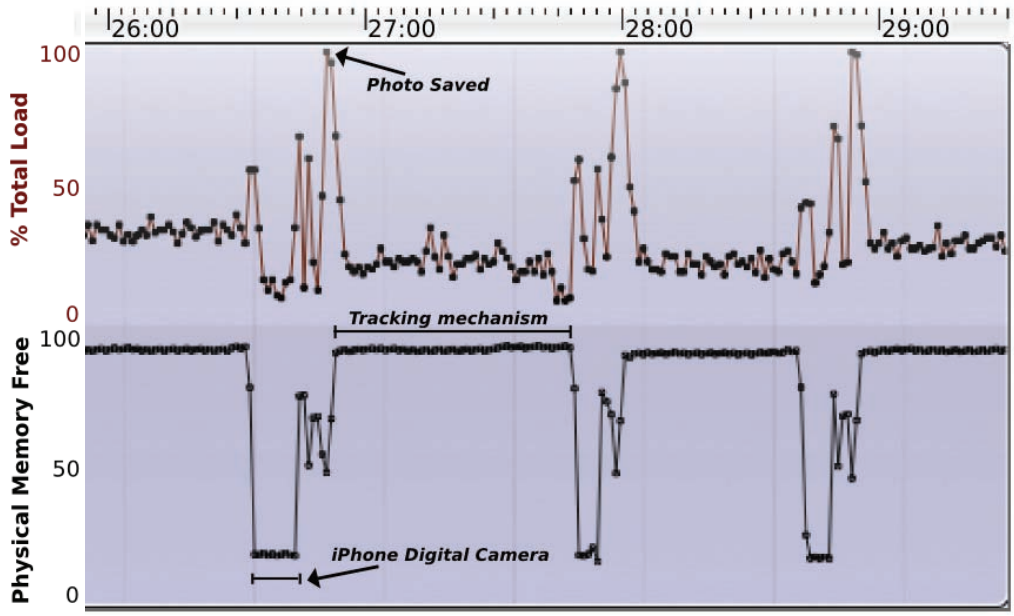
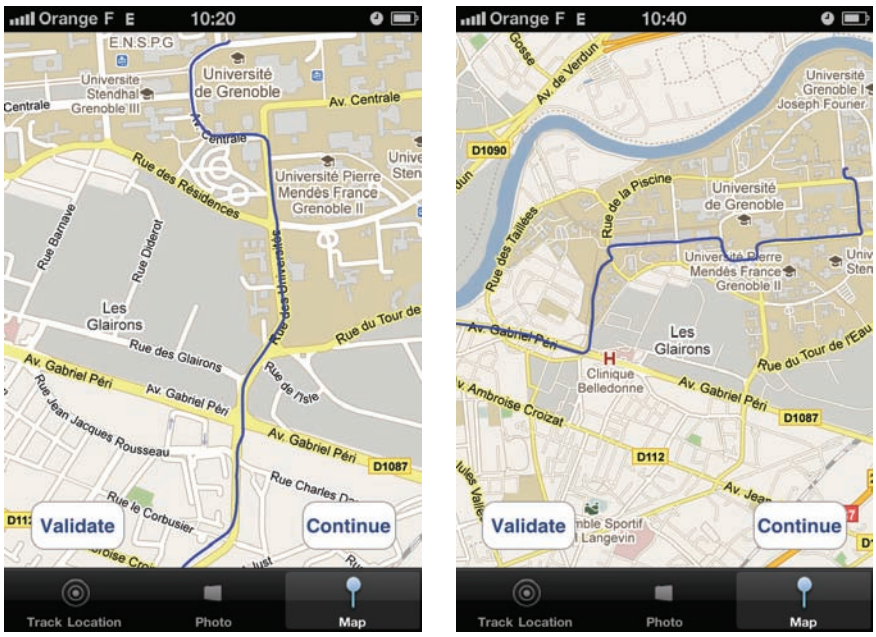


Figure 6.9: Physical Memory Free and Total Load in the iPhone.

bile phone during the interval from 26 to 29 minutes. The evaluation was conducted during the first tests, using the XCode Instruments [193] version 2.7. We observed that the *Total Load* (i.e., System and User) and the *Physical Memory Free* followed the same behavior while the mobile application functions were in operation. According to the results, the tracking mechanism requires approximately 10% of the memory and 25% of the processing to capture and register the positions. Likewise, while the iPhone digital camera is working, the memory used is approximately equal to 80% and the total load did not change. After taking the photo, the function *Save Photo* can be selected. When the *Save Photo* function is activated, the maximum load is used to associate and register all data and context information in the hard disk. Finally, the memory is cleaned when all data and information are associated and saved and the total load returns to follow the tracking mechanism. These results were important to guarantee that the user can use the application for a long time without stopping it due to memory or processing overhead problems.

Other important results are related to the mobile phone battery consumption during the trajectory registration. In the first test, when the distance filter had been configured to register each movement of the user, the iPhone battery level was down to 10% after 2 hours. After setting the distance filter to



(a) Daily trajectory of the user X. (b) Daily trajectory of the user Y.

Figure 6.10: Mobile Social Application.

fifty meters, the iPhone battery level was down to 10% after 3 hours. Another factor that can affect this result is the frequency that photos are captured.

The mobile application was one of the three modules developed in this project. Hence, more details about the usability, the desktop application and the server solution can be consulted in Appendix A.

This results achieved by the tests in a real scenario were important to observe the stability of processing and the memory use during the data acquisition process. Therefore, our solution can be used to efficiently obtain the necessary data (trajectory and context information) to our approach.

Based on these results, we adapted our mobile application to capture trajectories and context information about each location in urban centers. Figure 6.10 shows the interface of our mobile social application, containing the trajectories of two users who registered their daily trajectories from home to work.

As we can observe in the example presented in Figure 6.10, both users live and work in different places. The user X registers his/her daily trajectory that represents his/her daily routine to go from home to work. The user Y does the same registering process. Then, each user visualizes the trajectory

on the map. If this trajectory represents a good trajectory, the user validates it. Otherwise, the trajectory is rejected.

In terms of social networks, as previous mentioned, we enriched the database with the relations between users who register and share their trajectories. This data can be used to define different levels of relationships, such as: best friends, family, colleagues, friends, others. Along this line, we can define some controls to share personal trajectories only with users that have a certain level of relation with us. Therefore, in this evaluation, we used Facebook Developer Platform [194] to capture the relations between two best friends and assumed that they share their trajectories.

Analyzing the presented results and taking into account the use of context information to describe user routines, we conclude that our approach can be applied to a large number of applications, for instance: to offer a system that increases social interactions in real communities based on virtual communities (relations between friends in social network platforms); to develop a system that encourages rides among friends (car pooling); and others. Therefore, the data-sharing algorithm provides the information according to the requirements required by the application.

4 Conclusion

In this chapter, we presented some elements related to the evaluation of our approach. We started showing the implementation of our mobile application, which acquires trajectory data and context information of users. To validate the application, we tested it in a challenging scenario of a yacht traveling in the sea. After that, we adapted the application to register trajectories in urban canters, capturing all context information of each place by using reverse-geocoding techniques. Additionally, we developed a plug-in on a virtual community platform to receive the data containing user relations in social networks.

Next, we presented the evaluation of the OPTICS algorithm to discover the best representative trajectory of each user, which determines the user's daily routine. We explored the capabilities provided by this clustering algorithm to analyze user trajectories and extract relevant information from them. We focused on clustering and aggregating multiples trajectories generated by the

same user in order to identify habits or preferences. The results showed that this clustering algorithm is efficient to the requirements of our approach.

Finally, we introduced the results of our trajectory correlation algorithm, which finds similarities between multiple user trajectories based on each user preference and PoI. The results demonstrated that the similar routines between two or more users could be identified. Therefore, we conclude that our research provided interesting avenues for exploring Location-based Social Network (LBSN) applications. These avenues and the conclusion of this thesis are presented in the next chapter.

Chapter VII

Conclusion

Contents

This thesis has been realized in the context of Location-Based Social Network (LBSN). The key idea is to use the extensive knowledge about users' interests derived from social network platforms and their behaviors based on trajectory data in order to provide enriched information through a layer of service. While mobile phone provides the embedded features to register, store and publish personal information, the social network becomes an important platform for relating, enriching and sharing user interests.

Along this line, the context of our approach was classified into four main topics, which can be summarized by the following questions:

1. How can we use the user's information available in social network platforms (virtual communities) to increase the number of users' interactions in the physical world (real communities)?

2. How to make reasoning about similar user interests taking into account several data captured from different sources of knowledge? In addition, how to extract the relevant information on the acquired data to make reasoning from them?

3. What is the optimal restriction to share user interests based on their connecting strength, which are derived from relationship connections on social network platforms?

4. How can we represent PoI in multiple abstraction levels, taking into account the different descriptions of users about PoI?

Based on the challenges involving these questions, we proposed a layer of services, called LIDU (a Location-based approach to IDentify similar routines between Users in social networks). The main objective of this approach is to provide a service layer that allows to capture, store and process users' daily routines in order to find similarities between multiple user trajectories and near interests between two or more users. Finally, the enriched information can be used to provide a large number of services in different applications.

1 Summary of the contributions

A contribution of this thesis is the review of state of the art in which deals with the research topics considered in this thesis. Firstly, we presented how the most important related works define movement representation and its main features. After that, we introduced a conceptual view on moving object trajectories in order to help the understanding and analysis of spatio-temporal data. We also pointed out some works related to similarity analysis of moving object trajectories, in the context of spatial, temporal and spatio-temporal resemblance. Finally, we close the first chapter by showing an overview of the main challenges related to frequent problems in analyzing dissimilar trajectories as well as some solutions to solve these problems.

Next, we addressed our attention on spatio-temporal clustering methods to find trajectory patterns of moving objects in geographic spaces. Besides that, we showed the different ways to identify moving object patterns derived from the trajectories. Eventually, we finish the state of the art by introducing the conceptual definitions of social networks and their virtual communities. Next, we presented the main definitions of points of interests according to the representation specified by W3C PoI working group [3] and other approaches. Finally, we pointed out the current works in this area as well as we show the main concepts and definitions related to LBSN's.

The next two chapters are related to our approach, where the first one detailed the algorithms designed to our approach and the second one presented the results obtained in the evaluations. As we observed in the evaluation, our approach achieved interesting results. Based on the thesis results, we can conclude that our approach is able to provide a reasonable and useful source of knowledge and information to system designers and developers. This became

possible due to the use of robust algorithms and the designing of a scalable data representation in terms of user interests, while always respecting the particular features of the geographical space, such as heterogeneity, diversity of characteristics of relationships, and spatio-temporal autocorrelation.

Besides the review of the state of the art, we also presented the contributions of our approach. The thesis contributions aim to provide a solution to answer the previous questions.

Starting at the most specific contribution, the correlation algorithm achieved significant results as a solution to process user interests, trajectories, social relations and temporal information to increase the number of social interactions in real communities. This contribution is directly related to the use of social relations derived from virtual communities and the trajectory data obtained by mobile devices. Therefore, the two first questions deal with this first contribution.

According to the intermediary contribution, we concluded that the clustering algorithm was efficient to extract relevant information from trajectory data. This efficiency was validated by the recognition and generation of best representative trajectories of users, which can be used as relevant source of data for different algorithms. Hence, this contribution is associated with the extracting and optimization of the acquired data that is used to make reasoning, which answers the second question

Finally, the general contribution is the whole approach, which provided enriched information about user's similarities to a large number of applications. This was possible due to the designing of a flexible multi-layer data model based on different ways to represent user routines. Consequently, the general contribution answers all the questions related to this thesis, offering a reasonable and optimal source of knowledge to system designers and developers.

2 Perspectives

The contributions presented in this thesis bring up some perspectives for the continuation of our approach. The perspectives can be categorized into five parts, three parts related to the optimization of our approach and two parts associated with the extension of our approach. The three perspectives related

to the optimization of our approach are derived from trajectory correlation algorithm, clustering algorithm and data modeling.

- We explained the MBR expansion process for solving the problem of near points of interest that are not identified in the correlated area. However, we did not evaluate it in a real scenario. Hence, we intend to evaluate this process and compare it with other solutions, such as caching-based methods. Additionally, we aim to reuse our proposal in different types of scenarios and to develop a mobile social application based on the enriched information provided by the algorithm. In terms of applications, several applications can be developed making use of this enriched data, such as: a variety of recommendation systems, urban planning, traffic analysis, Web 2.0 based solutions, GIS tools and other types of mobile social applications.

- We have addressed the clustering algorithm on a static dataset, which the order of processing chain (e.g., acquisition, modeling, handling) is respected. In summary, the acquired data are modeled and integrated to the database and then handled by another dedicated software. With these aspects in mind, we would like to know what are the impacts that dynamic geo-referenced data could have on the database and/or the software that handle these continuously updated data. It is important to emphasize that studies with micro-clustering algorithms showed that these algorithms could achieve interesting performance results when applied in dynamic datasets, where the data is frequently updated [124] [121]. In addition, it is essential to perform a comparison between different clustering techniques in order to find the best approach for covering the requirements of several scenarios/applications.

- In the context of the data model, we also intend to investigate the impact of dynamic datasets in our multi-layer data model. We aim to evaluate the data model in the context of trajectory data flow. Additionally, we would like to perform a study of different aspects related to the knowledge, which can reveal strengths and weaknesses derived from user's trajectory data and the relationships in virtual communities. Thus, it is essential to investigate the different levels of relationships in order to automatically relate users with similar interests to recommend some

information. Besides that, it is important to study the performance of abstraction level adaptation in terms of multilayer indexing of movement data, as the evaluation presented in [195].

- In this thesis, we have used only one source of contextual data. However, we can use multiple sources, which can affect the results related to the data reasoning. The crossing between multiple data can enrich the acquired information and increase the data reasoning process. Nevertheless, the well definition of a procedure to manage a large number of data becomes necessary.

- In the context of privacy, it is possible to design different policies to determine the access control in the sharing of user's interests. For example, the relationship level (best friend, family, others) is one way to do that, however, the similarity ranking (based on the users who have similar routines) could be another method. Therefore, the user has to be comfortable to share his/her information between trusted-users. While the Role Based Access Control (RBAC) is widely adopted in several types of computer networks, we observed that it does not support the dynamic characteristics of pervasive environments. Hence, we aim to follow the approaches of [196] and [197] to apply RBAC-based access control in the shared data.

Besides the perspectives in terms of our approach, some extensions can be detailed, such as quality of information and streaming databases.

In terms of the quality of information, we know that the geo-referenced information is not completely accurate. For example, if we use a reverse geocoding to obtain the number of a house located in a specific street based on the latitude and longitude, we can obtain different information. Therefore, this kind of problem has to be considered in the case of solutions that need to capture accurate information about a specific location.

Finally, we would like to stress that our approach was designed to be adaptable for adding new solutions, which explore the trajectory data and are based on some kind of connections between users.

Part III

Appendix

Appendix A

A Context-Aware Web Content Generator Based on Personal Tracking

This paper was published on the 11^{th} International Symposium on Web and Wireless Geographical Information Systems (W2GIS)

Abstract *Context-awareness has been successfully included in the mobile phone applications due mainly to the presence of numerous sensors and the access to several communication networks. Therefore, we present a Context-Aware Web Content Generator Based on Personal Tracking, which uses the user context information obtained by mobile devices to generate content for a large number of web applications. While registering the trajectory followed by the mobile device, it allows users to create multimedia documents (e.g. photo, audio, video), which are connected to an enriched description of the user context (e.g. weather, location, date). Finally, all this data and documents are combined to produce a new content, which is published on the Web. We also show results of tests performed in a real scenario and describe our strategy to avoid battery overconsumption and memory overflow in mobile phones. Moreover, a user evaluation is presented in order to measure the system performance, in terms of precision and system overall usability.*

1 Introduction

Mobile phones, nowadays, are not simple call-making devices anymore. They have already become real information centers. With all the embedded features like GPS, accelerometer, Internet connection, digital camera, among others, a user easily creates and publishes personal multimedia content. For instance, any user can quickly take a picture and put it in his/her web-based photo

album. In addition, multimedia content can be enriched and organized with context information collected by smartphones, such as date, geographical position and current weather.

There are several applications that use context information to enrich and organize multimedia documents. This information might be proximity of people or objects in the photo, current temperature, date, etc. This type of metadata can be obtained from sensors of mobile devices or from the web. With this information associated, context-aware applications can better organize the multimedia, providing user-friendly visualization of the content, and suggesting annotations for document indexation [198][199][200].

In this paper, we go a step forward proposing the use of context information to generate new multimedia content. First of all, the user trajectory is registered by using the GPS sensor of the device. While registering the trajectory, the user can produce multimedia documents, such as: photos, audio or video. Likewise, context information can be associated with each multimedia created, as geographic position, date, and temperature. These data will be easily shared to the Internet, presented as a microblog, for example. In short, our system works in three steps: i) collecting context and user-added data; ii) processing and organizing them; iii) publishing the composed content on a web-based application (e.g., blogs/microblogs, web albums).

It is also important to mention that context-aware systems have some dependencies that may not be satisfied in some situations. The Internet connection, for example, can be limited or even not available at certain moments. Another problem is related to the mobile device battery. For example, all these features (GPS, Bluetooth, Internet access, etc.) are necessary to the context data acquisition, but they spend too much electric power. In order to minimize these dependencies, we propose some design decisions that have impact in trajectory and context gathering mechanisms.

In the interest of evaluate the system usability and the performance of our gathering mechanisms, we apply it in a challenging scenario. It was used by three of the crewmembers of a boat as a digital logbook. The system registered the boat trajectory, allowed the insertion of photos, suggested annotations using context information and published the content in the blog of the project ZeroCO2[1].

[1] www.zeroco2sailing.com/blog/

The organization of this paper is presented as follows. Section 2 presents related works and introduces an overview about context-awareness. Section 3 presents our proposed system. Section 4 discusses a case study of our system tested in a real situation. Section 5 presents results of the system performance and user evaluation. Finally, Section 6 concludes this work and gives some perspectives.

2 Context-awareness is more than system adaptation

Several research areas use the notion of context with distinct meanings.

In the field of information systems, the concept of context refers primarily to the user status and the surrounding environment at the moment he/she is accessing a system. Frequently, the knowledge of the user location is a prerequisite for the success of this kind of system. According to Dey *et al.* [201], the context is constructed from all information elements that can be used to characterize the situation of an entity. An entity is defined as any person, place or thing (including users and the own applications) considered relevant to the interaction between the user and the application. Consequently, the term "context-aware" is associated with systems that guide their behavior according to their context of use. Most authors in this field consider context awareness as the ability to perceive the situation of the user in several aspects, and adapt the system behavior accordingly [202].

On the other hand, in the multimedia domain, the notion of context, and mainly, its exploitation is slightly different. Context-awareness is more than simple adaptation mechanisms. This distinction is studied in some works, such as Naaman et al. [203], which presented the behavior of users to organize and find photos. In fact, most of the information referred by people about their image memories consists of aspects related to their context at the moment the photo is taken (when, where, with who, etc.). These authors argue that the information about the context creation of a photo facilitates the search of a specific photo in a set of multimedia documents.

The popularization of mobile devices equipped with location sensors and GIS (Geographical Information Systems) have provided the technology and data necessary to develop multimedia systems able to gather the desired con-

text information. Nowadays, we can categorize these context-aware multimedia systems in three groups: multimedia organization and annotation tools; multimedia sharing systems; and context sharing systems.

2.1 Multimedia organization and annotation tools

Following the aforementioned concepts in Naaman *et al.* [203], some research projects and commercial applications propose automatic photo annotation by using context metadata. In fact, nowadays, the use of photo geotagging is not unusual for mobile users since most of the smartphones contain geotagging applications. For example, in Kennedy *et al.* [204], the authors identified local markers from 110,000 Flickr images of the San Francisco Bay Area. Most of the photos were taken from mobile phones and were georeferenced. Hence, image data with views that best represent a marker according to visual similarity were retrieved by means of a marker or location search.

Research projects, such as PhotoGeo [200], PhotoCopain [205], MediAssist [199], and PhotoMap [206] gather a larger set of contextual metadata, which includes user location, identity of nearby objects and people, date, season and temperature. They exploit these contextual metadata for photo organization, publication and visualization. For instance, PhotoMap provides automatic annotation about spatial, temporal and social contexts of a photo (i.e., where, when, and who was nearby). PhotoMap also offers a Web interface for spatial and temporal navigation in photo collections. The system exploits spatial Web 2.0 services to show where a user took the photos and the itinerary followed when taking them.

2.2 Multimedia sharing systems

The modern capabilities of mobile devices and the success of Web 2.0 sites stimulate a new kind of multimedia phenomenon: the create-to-share behavior [207]. Mobile users create multimedia using their devices and with the purpose of sharing the information almost instantly.

Some context-aware systems try to exploit context metadata to increase this experience of multimedia sharing [207] [208]. For instance, Zonetag projects [207] use the position information to suggest the photo annotation before sharing it. Other approaches aim to refine the multimedia content taking

into account the user context. For example, the Aware project [208] replaces the MMS application in Nokia mobile phones by a context-aware application, which adds automatically the position information to each MMS sent by the user, such as an address derived from the combination of a GSM Cell-ID and an address database.

2.3 Context sharing systems

A large number of messages shared on social networks, such as FourSquare and Twitter microblogs, refers to the information of user context. Hence, this information can be derived automatically by mobile phones equipped with sensors [209] and published on these Web sites. For instance, ContextWatcher [210] is a mobile application to capture and share the most common context information. The main objective is to acquire and describe accurately the current status of the user. The context information of a user is composed of position (e.g., geographic coordinates, altitude and address), speed, humor, heartbeats and weather. All this information is combined and published over a map-based site that shows the current context of all users.

Other approaches, such as Melog [211] and SnapToTell [212], propose the generation of more complex multimedia documents from a set of pictures created by users and the context information associated with them. For instance, Melog tries to recognize events by using clustering techniques. The identified events are used to structure a micro-blog about the user travels.

3 Our Approach

Taking into account the classification of context-aware groups, our proposal can be classified into groups one and three. Figure 1.1 presents an overview of our system, which is divided in three main parts: Data Acquisition, Data Processing and Publishing. Data Acquisition concerns the sensor application, note writing, data capturing and every other data collection process. After that, all acquired data will be processed in the Data Processing. In this part, the system uses the raw data in order to capture inferred information and to suggest a textual annotation. When the user context is properly collected and inferred, the Publishing part initiates its process. Finally, a new content is formerly produced and can be shared in the Internet, taking into account the

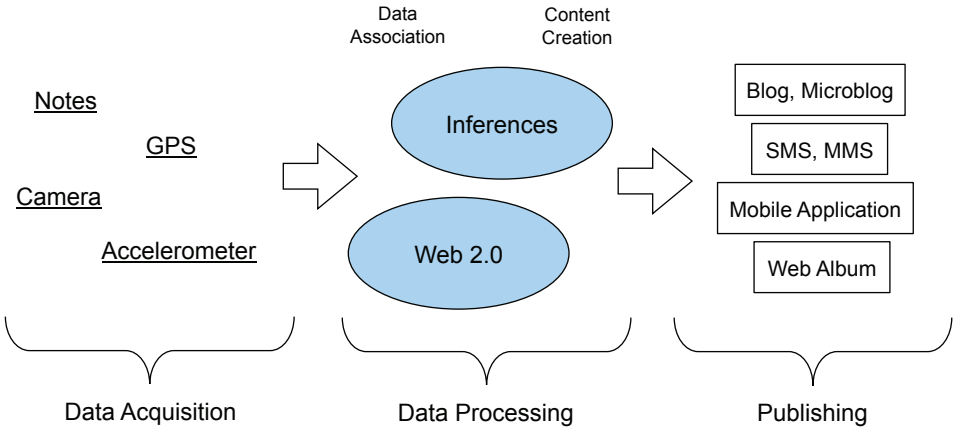


Figure 1.1: System Overview.

association of each context information.

For instance, a user is registering the trajectory of his/her boat trip using our system. While he/she is arriving in the harbor, he/she decides to take a photo of another boat. At this moment, besides the photo, the sensors acquires context information, such as position, direction, speed and weather[2]. The collected data is manipulated by the second part in order to acquire inferred information and to suggest textual annotations to the user. After validating the annotations and association with the photos, the user can visualize his/her augmented trajectory and publish the content on the web.

Nowadays, one of the main features of context-aware systems is the location tracking. Our system also relies in this feature. It gets the mobile device position periodically by GPS and derives the trajectory followed by the mobile. In addition, GPS collects the geographic location of a taken picture to add this information in the metadata. This action is important to help the content generation as well as to acquire new information (e.g., weather) of a photo that was previously taken. These three parts of our system are detailed in the next sections.

3.1 Data Acquisition

One of the most important parts of our system is the data acquisition. It uses the sensors in the mobile device to get information about localization, device

[2]weather will be acquired if an Internet connection is available.

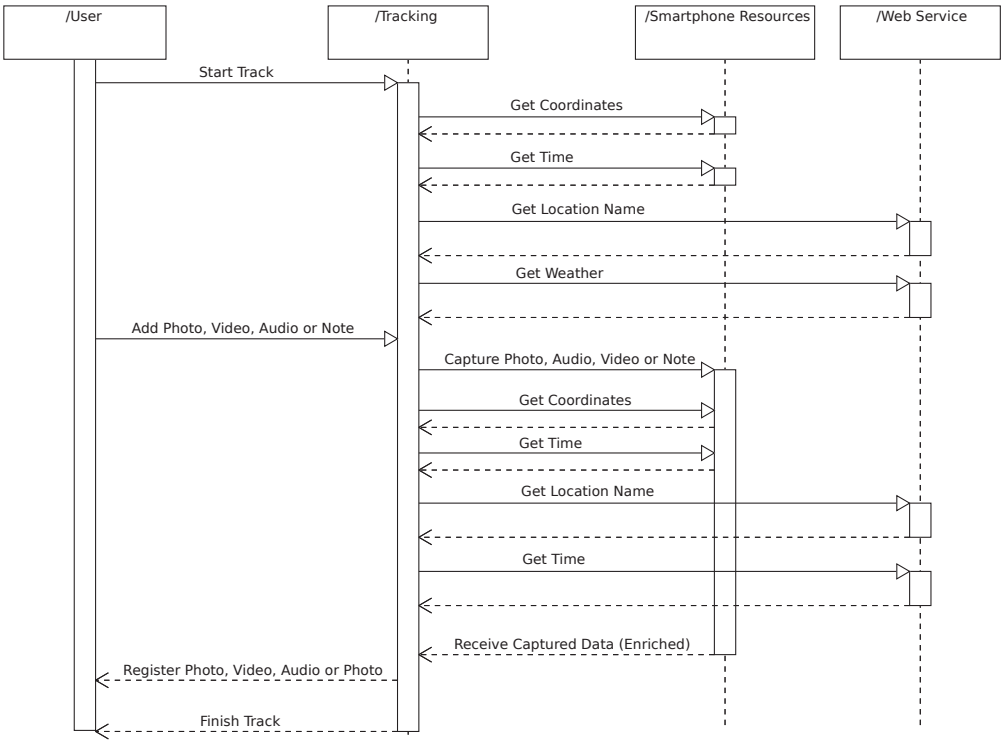


Figure 1.2: Sequence Diagram of Data Acquisition.

orientation, speed, time, etc. In addition, some initial notes made by the user are also considered as *Data Acquisition*.

During the data acquisition process, we have to do the relations between each information collected, as presented in Figure 1.2. According to the Figure, the user starts the data acquisition process in the mobile device. The tracking mechanism, then, begins to register the user trajectory. While the tracking mechanism is running, the user decides to take a photo, creating a new event. At the moment, the parameters of the digital camera are defined. Besides that, the context information is gathered using sensors. Other kind of information can be obtained if an Internet connection is available, such as the location name and weather conditions, both using the position information acquired by the GPS.

Since we are working with mobile devices, the efficiency of our system is directly related with the battery consumption. To reduce the overconsumption of battery, we propose the insertion of a distance filter in the tracking mechanism. The key idea is to avoid registering coordinates for short distances. Consequently, we have to observe what is the best distance filter value to reg-

ister the coordinates. For example, we define the distance filter equal to 10 meters, the mobile device will register the current position in the metadata if it is higher than 10 meters. Otherwise, it will be dropped. The distance filter is an important feature of our system because it is responsible for the relation between battery consumption and trajectory construction.

In order to improve the data processing step, it is important to organize the acquired data into the metadata. Therefore, we used *tags* to arrange each information in the metadata.

3.2 Data Processing

The *Data Processing* is the real core of the system. It is responsible to increase the robustness of our system by offering more than a context-aware data collector, as follows. It associates, suggests and organizes the information in order to provide a comprehensive structure to be published.

Making use of the acquired data organized by tags, the data processing part is started. It has to provide an interface for users to facilitate the content generation. The key idea is to use the context information of a data to suggest the text that will be published. Our system provides an initial recommended text based on the information acquired by the mobile application. For example, if a user is registering his/her trajectory and takes a photo in a specific position, the system will generate a new photo with the name *IMG0001* and will register the coordinates *45°10'0"N, 5°43'0"E* at *15:00* on *03/02/2010*. Besides that, the user adds a note describing some characteristics of the photo. According to Figure 1.3, a text is suggested for each acquired data. Following the previously example, if the user selects the photo $IMG0001$ in our system interface, then a new text might be suggested: *"The photo IMG0001 was taken on the location < location_name > at 15:00 on 03/02/2010 and the weather was < weather_status >. < additional_note >".*

If the mobile device due to an absence of connection does not acquire the information of location name and weather, our system interface has to be able to obtain this information based on context information. Nevertheless, the specialized web services provide the weather status for present and future times. To solve this problem, we propose a mechanism to capture this information using a HTML parser in order to get the location name and weather status for the past time. This parser reads the web page *Daily History* of the

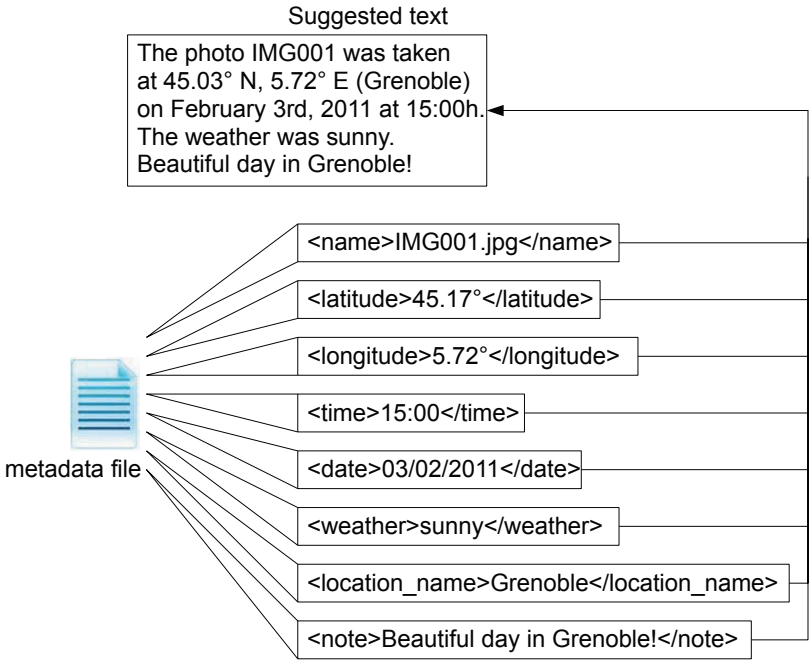


Figure 1.3: Data Processing.

Weather Underground and obtains the weather status related to the context information provided by the acquired data. When the user generates the content to describe all events registered during his/her trajectory, the third step of our system can be started.

3.3 Publishing

The last part of the system is responsible for publishing the content produced. In Figure 1.1, we have proposed some applications to publish the content, such as microblog, SMS, mobile application, etc. In spite of the existence of a large number of applications to publish the content generated by our system, we choose the web content publication on blogs/microblogs because of their natural manner to publish the web content. Their structure, based on individual posts, is perfect to publish a data with context information. We can use the natural content organization to sort the posts in terms of the context information. Moreover, the user could view the content organized by

day or by place, for example.

In addition, we propose a map-based interface and pop up windows in order to present the content (annotations, photos, audio, video and context information related to a position) in the trajectory. We intend to use map-based interfaces taking into account the usability studies presented in the literature [213][199]. These works show that map interfaces demonstrate more interactivity advantages than browsing information with hierarchical links. Moreover, with a map-based interface, we can easily illustrate the trajectories generated by the mobile users together with the context information.

4 Using the proposed system in a real situation

To evaluate the efficiency of our system in a real situation, we implemented our proposal for the ZeroCO2 project [190]. We designed our system to be a digital logbook during a boat expedition around the Mediterranean Sea. The logbook, which was created as a book to record readings from the ship log [191], is an essential instrument to the navigation and has to be used daily. In general, the crew uses paper-based logbooks to register all information and, frequently, the information is collected from distinct equipments. Hence, we concluded that our system was able to create a complete logbook for this boat expedition. In addition, the challenging scenario of the sea added some problems involving the recurrent absence of Internet connection and the lack of battery charging.

Our system was responsible to track the trajectory followed by the boat, adding all context information to each registered coordinates. Although our system proposes the use of audio, video and photo as data, we used only photos for this first experiment in the project ZeroCO2. Taking into account this scenario, we face new challenges that have motivated us to improve the context-aware system proposed in the previous section.

4.1 Challenges and System Improvements

When using any context-aware application in the sea, we have to handle new challenges in order to avoid problems in the application and information loss.

The main difficulties that we consider in this work are detailed as follows.

- **Lack of continuous Internet connection**. In a sea expedition, frequently, there is an absence of Internet connection on mobile phones. 3G or 4G signal is perceived only when the ship is near the coast. Some high-level context information, like a location address, can be computed in a future moment (i.e., when an Internet connection is available). However, some other data cannot be easily recovered. For instance, Weather Forecast services only provide real-time information. For this reason, a special context "cache" system should be designed for providing past context information.

- **Robustness:** the absence of Internet connection and the movable nature of a ship expedition make the remote repair of the mobile application impossible or in situ. Thus, the mobile application has to be reliable. Previously, our research team has also developed context-aware multimedia systems following the architectures of PhotoMap [206] and CoMMediA (Context-Aware Mobile Multimedia Architecture) [198]. Therefore, we tested both projects that adopt Java Mobile Edition as mobile platform. In the user tests of these systems, some memory overflow incidences occurred caused by simultaneous access to the GPS sensor and the camera phone. This problem occurs even when using synchronized threads, and, sometimes, requires redeployment of the mobile application.

- **Energy limitation to recharge mobile devices**. Another critical problem found in the PhotoMap and CoMMediA projects was heavy energy consumption during the use of the mobile application. For instance, in forty minutes, the battery of a Nokia N95 was fully discharged since GPS, photo camera and Bluetooth sensors are greedy in energy consumption. In some ship expeditions, electric energy restrictions are present and the mobile application should be designed to overcome this issue.

Based on the system overview presented in Figure 1.1, we decided to divide the digital logbook in three parts: a mobile application to register the boat trajectory; a desktop application to receive the acquired data and generate the web content; and a blog to publish the content.

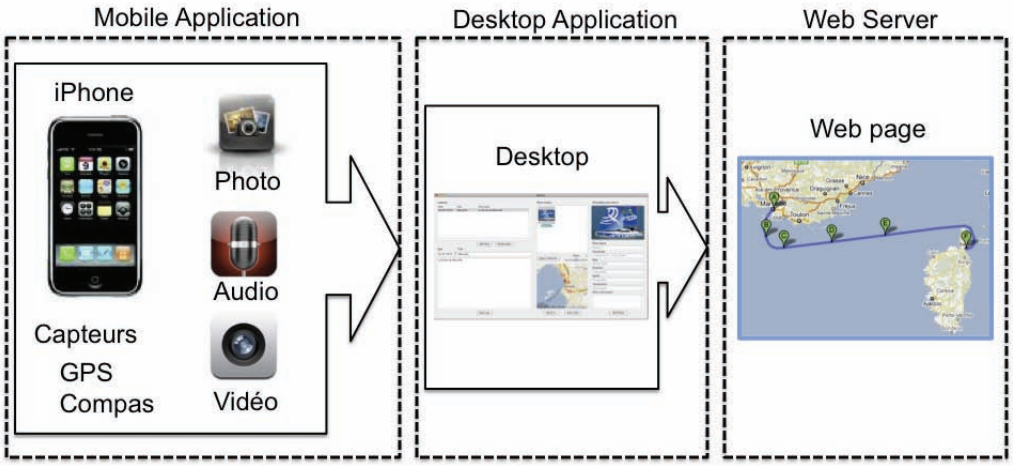


Figure 1.4: Digital logbook parts.

An overview of the digital logbook is presented in Figure 1.4. The *Data Acquisition* process was developed in the iOS platform, since the user-friendly interaction is well known in this mobile platform. The mobile application performs the boat tracking, take the photos, and carries out the relation among each data and its context information. It is important to note that some of *Data Processing* features were also implemented in the mobile phone, such as the inference mechanism to get the weather status and location name.

The *Data Processing* step was implemented as a desktop application to offer an interface of creation and publication of web content. It implements the module to get the context information that was not acquired by the mobile application, using the HTML parser. Another important feature in the desktop application is the function of text suggestion for each photo. We tried to develop a robustness and intuitive interface user interface to improve the usability.

The *Publishing* step was developed using an open source blog solution, which is available on the web page of the ZeroCO2 project[3]. It shows the complete digital logbook, containing the content generated by the crewmembers and the map with the boat trajectory.

[3] www.zeroco2sailing.com/blog/

(a) Tracking mechanism. (b) Digital camera.

Figure 1.5: Tracking mechanism and digital camera.

4.2 Mobile Application

The mobile application interface is shown in Figure 1.5. As we can observe, there are two main functions: the tracking mechanism and the digital camera. The tracking mechanism is responsible for registering the geographic coordinates to construct the trajectory. The interface shows the position, speed, date and, if Internet connection is available, wind speed and humidity. The digital camera takes a picture and, automatically, adds the context information to it. There is also the option "Tag" with which you can add the information manually.

An important result discovered during our tests is related to the use of metadata following some standard, such as Web Ontology Language OWL [192]. Several solutions adopt this language to obtain inferred information about a context. However, it needs to add a large number of information in the metadata file to perform this task. Consequently, the mobile application generates several unused information into the metadata file, causing some problems of memory overflow in the mobile application. Therefore, we optimized the content of our metadata files registering only the relevant infor-

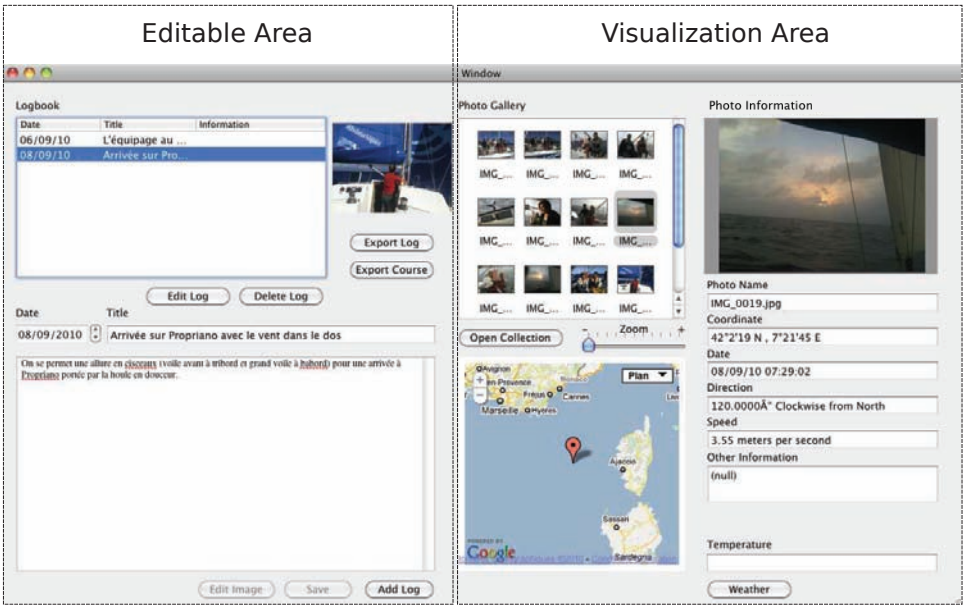


Figure 1.6: Desktop Application Interface.

mation. Besides that, we developed our own local parser to get the information of each tag and to infer about context information using the HTML parser.

4.3 Desktop Application

The desktop application interface is shown in Figure 1.6. As stated, the desktop application has two segments: the editing area and the visualization area. In the editing area, the user can add an annotation based on the text suggestion function of our system (Figure 1.3). Besides that, the user is able to edit previously annotations. In the visualization area, the application shows the photo album jointly with all context information about each selected photo. A small map shows the position where the selected photo was taken. In addition, this interface permits that the user captures weather status of a photo, in case this information was not captured at the moment the photo was taken, due to an absence of Internet connection. This is only possible due to our proposed HTML parser (Section 3.2).

4.4 Web Application

The web application is a microblog solution, in which each annotation created by the crewmember is posted. Some parts of the blog are shown in Figure

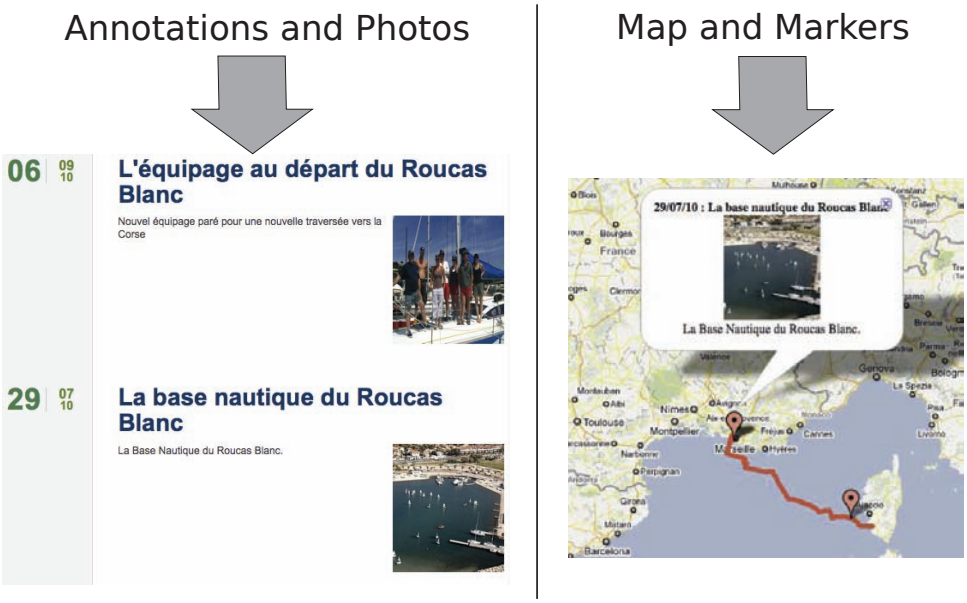


Figure 1.7: Web Application.

1.7. All content generated by the user in the desktop application is stored in a MySQL database and consulted by Java and PHP scripts. In addition to the illustrated posts, it also presents a map showing the trajectory of the boat. This map was developed with the Google Maps API [214]. We developed our map-based interface taking into account the usability studies presented in the literature [199] [213].

The web application also performs an indexation process to improve browsing and interaction procedures. The amount of multimedia and context information increments quickly in our system. Then, to avoid future performance difficulties related to the large number of access, spatial and temporal indexes are associated with each annotation in the MySQL database.

5 Results

In this section we describe the performance and user evaluation of our system.

5.1 Performance Evaluation

The first evaluation was done during a travel around the Marseille coast. We ran this first test to calibrate the distance filter option and to execute the

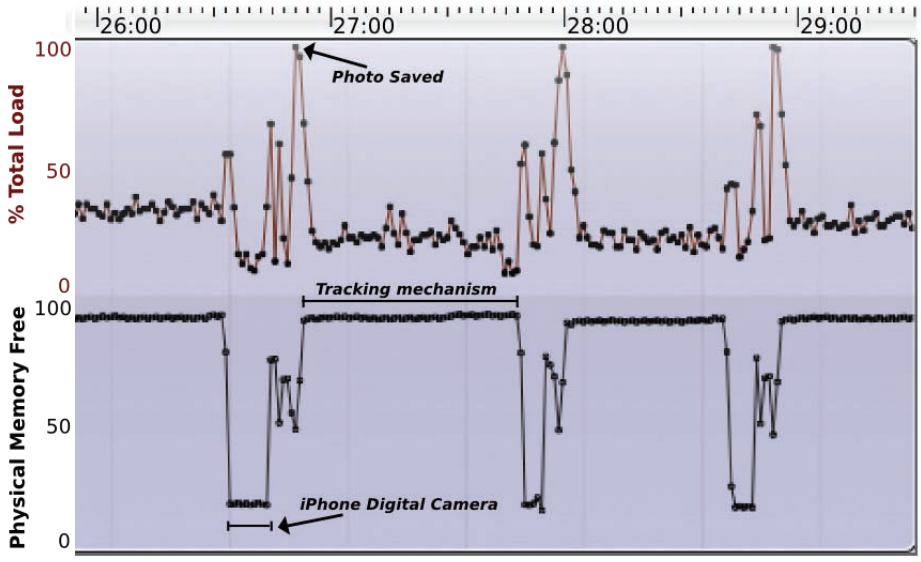


Figure 1.8: Physical Memory Free and Total Load in the iPhone.

performance evaluation in the mobile phone. As explained before, this option is responsible to define the detail level of the trajectory. We assigned the value fifty meters to the distance filter, which means that a position will be registered if it is higher than fifty meters in comparison with the last position registered. With the first results, we refined our system to the second test: a travel from Marseille to Ajaccio (Corsica Island).

Figure 1.8 shows the performance evaluation of our application in the mobile phone during the interval from 26 to 29 minutes. The evaluation was conducted during the first tests, using the XCode Instruments [193] version 2.7. We observed that the *Total Load* (i.e., System and User) and the *Physical Memory Free* followed the same behavior while the mobile application functions were in operation. According to the results, the tracking mechanism requires approximately 10% of the memory and 25% of the processing to capture and register the positions. Likewise, while the iPhone digital camera is working, the memory used is approximately equal to 80% and the total load did not change. After taking the photo, the function *Save Photo* can be selected. When the *Save Photo* function is activated, the maximum load is used to associate and register all data and context information in the hard disk. Finally, the memory is cleaned when all data and information are associated and saved and the total load returns to follow the tracking mechanism. These results were important to guarantee that the user can use the application for a

long time without stopping it due to memory or processing overhead problems.

Other important results are related to the mobile phone battery consumption during the trajectory registration. In the first test, when the distance filter had been configured to register each movement of the user, the iPhone battery level was down to 10% after 2 hours. After setting the distance filter to fifty meters, the iPhone battery level was down to 10% after 3 hours. Another factor that can affect this result is the frequency that photos are captured.

5.2 User Evaluation

Once the first distance filter adjustments were performed, our system was used by three ZeroCO2 crewmembers. After a one-week expedition, the users filled in a general usability survey. Despite the small number of users, we have tried, with this questionnaire, to measure the main benefits and issues of our context-aware annotation proposal. We also wanted to know if using a mobile phone in an "adverse environment" could disturb the real ZeroCO2 missions. The following survey questions were asked:

- Rate how easy it was to create a digital logbook without and with our digital logbook.
- Rate how fast it was to create a digital logbook without and with our digital logbook.
- How do you qualify the accuracy of the digital logbook generated annotations?
- Could you describe the main digital logbook advantages and shortcomings?

For the first three questions, a five-scale graph was provided in which the number one corresponds to a very bad rate, and five corresponds to very a good rate. For instance, for question 1, the number one corresponds to very difficult, and five corresponds to very easy. Figures 1.9 and 1.10 show the experiments results for the first two questions.

Without our system, the crewmembers have to synchronize all information collected by a digital camera with a desktop application (e.g., a word processor) in order to create a digital logbook. Additionally, another step has to

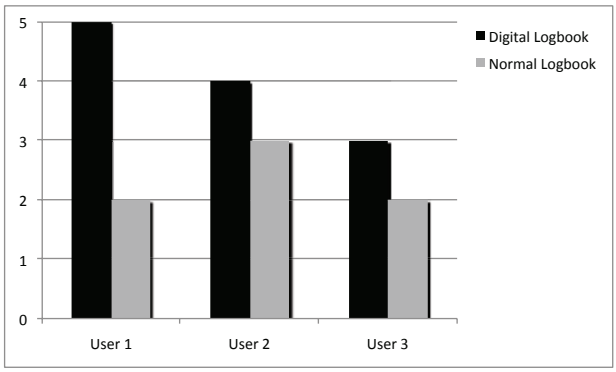


Figure 1.9: Easiness comparison between our digital logbook and normal logbook tools.

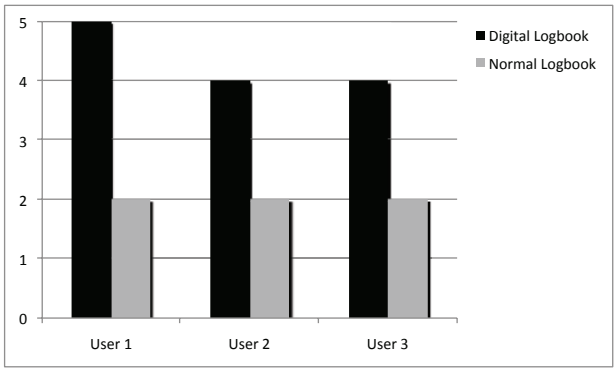


Figure 1.10: Annotation time comparison between our digital logbook and normal logbook tools.

be performed for publishing the logbook information on the web (e.g., using a blog authoring tool). With a mobile device and the digital logbook, most of the processes of logbook creation, edition, and publishing are automated by our proposal. The survey results presented in Figures 1.9 and 1.10 reflect the differences between these two approaches. Interestingly, two users have given a greater difference in scores concerning the time question (Figure 1.10), which shows how fast it is to publish information with our digital logbook.

Regarding the accuracy question, two users have scored "precise", and the other one has scored "very precise". Despite the use of distance filter option, one can see that the generated annotation is still very satisfactory for the users.

For question 4, the users have highlighted the advantages of intuitive interface on the iPhone application, and the simplicity and speed for logbook

creation. None of the users have mentioned disruption on their daily missions. However, the synchronization between the iPhone and the Mac book was pointed out as the main drawback. Two users have even suggested skipping this step by editing the information on the iPhone and publishing them directly on the Web.

With these results in mind, the generation of context-aware annotation is, as we expected, a useful way to automate multimedia edition and publishing even in an "adverse environment".

6 Conclusion

In this paper, we presented a new context-aware web content generator based on personal tracking. It is a context-aware system for the creation, annotation and sharing of multimedia content. We showed that our solution was used as a wizard editor for the generation of a real digital logbook. By designing a practical and efficient strategy, our system provided a user-friendly interface and offered a mechanism for context acquisition that avoids battery overconsumption and memory overflow.

Usability and performance tests were also performed in collaboration with ZeroCO2 project. The user evaluation results demonstrated that the crewmembers were comfortable using our system and found it an excellent tool to accurately publish context information according to the geographical position. Beyond our approach for context-aware systems, another important contribution is associated with the development of a context-aware photo management tool on smartphones.

As future work, we aim to offer a framework for the development of context-aware systems. This framework will provide a collection of procedures able to acquire, store, increase and infer contextual metadata related to multimedia document. The key idea is to reuse our proposal in several types of scenarios, for example: tracking an excursion in forests and mountains; studying the behavior pattern of a vehicle based on its speed, course, and position; mapping the course of runners and other athletes; and applying that for mobile learning lectures such as Geology courses that are usually taken in the field.

Bibliography

[1] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering Points to Identify the Clustering Structure. *ACM SIGMOD Record*, 28:49–60, June 1999. 3, 56, 57, 58, 128, 129, 130

[2] Natalia Andrienko and Gennady Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):205–219, February 2011. 3, 61, 62, 63

[3] W3C PoI. W3c points of interest working group charter, Jun 2012. 3, 23, 73, 76, 79, 80, 81, 105, 142

[4] Yu Zheng. Location-based social networks: Users. In *Computing with Spatial Trajectories. Yu Zheng and Xiaofang Zhou (editors)*, pages 243–276. Springer New York, 2011. 4, 13, 83, 84

[5] Yu Zheng and Xing Xie. Location-based social networks: Locations. In *Computing with Spatial Trajectories. Yu Zheng and Xiaofang Zhou (editors)*, pages 277–308. 2011. 4, 85, 88, 89, 91, 92

[6] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. Smart itinerary recommendation based on user-generated gps trajectories. In *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing*, UIC'10, pages 19–34, Berlin, Heidelberg, 2010. Springer-Verlag. 4, 18, 85, 86, 89, 90, 105

[7] Windson Viana, Jose B. Filho, Jerome Gensel, Marlene Villanova-Oliver, and Herve Martin. Photomap: from location and time to context-aware photo annotations. *Journal of Location Based Services*, 2(3):211–235, September 2008. 4, 101, 102

[8] Eija Kaasinen. User needs for location-aware mobile services. *Personal Ubiquitous Computing*, 7:70–79, May 2003. 11

[9] Sarfraz Khokhar and Arne A. Nilsson. Introduction to mobile trajectory based services: A new direction in mobile location based services. In *Proceedings of the 4th International Conference on Wireless Algorithms, Systems, and Applications*, WASA '09, pages 398–407, Berlin, Heidelberg, 2009. Springer-Verlag. 11

[10] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, EuroSys '09, pages 205–218, New York, NY, USA, 2009. ACM. 12

[11] Yu Zheng and Xizofang Zhou. *Computing with spatial trajectories*. Springer-Verlag New York Inc., 2011. 12, 14, 17, 25, 42, 48, 87

[12] Facebook. http://www.facebook.com/, Jun 2011. 14, 83

[13] Linkedin . http://www.linkedin.com/, June 2011. 14, 83

[14] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 791–800, New York, NY, USA, 2009. ACM. 17, 18, 86, 105

[15] Reinaldo B. Braga, Socrates de M. M. da Costa, and Herve Martin. A trajectory correlation algorithm based on users' daily routines. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 501–504, New York, NY, USA, 2011. ACM. 17, 76, 105

[16] Yu Zheng, Lizhu Zhang, Zhengxin Ma, Xing Xie, and Wei-Ying Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web*, 5:5:1–5:44, February 2011. 17, 85, 87

[17] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM - Special issue on information filtering.*, 35(12):61–70, December 1992. 17, 91

[18] Atsuyoshi Nakamura and Naoki Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 395–403, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 17, 91

[19] Yu Zheng and Xing Xie. Learning location correlation from gps trajectories. In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*, MDM '10, pages 27–32, Washington, DC, USA, 2010. IEEE Computer Society. 18, 85

[20] Yu Zheng and Xing Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology*, 2(1):2:1–2:29, January 2011. 18, 87, 90

[21] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing*, 16(5):469–484, 2012. 18, 85, 86

[22] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1029–1038, New York, NY, USA, 2010. ACM. 18, 85

[23] Curtin University of Technology. Are virtual communities a good thing socially?. Department of Internet Studies, Mai 2010. 19

[24] Stefano Spaccapietra, Christine Parent, Maria Luisa Damiani, Jose Antonio de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. *Journal of Data & Knowledge Engineering*, 65:126–146, April 2008. 19, 26, 27, 30, 31, 80, 106, 110

[25] Christine Parent, Stefano Spaccapietra, and Esteban Zimányi. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 26

[26] Fosca Giannotti and Dino Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008. 26

[27] Hechen Liu and Markus Schneider. Querying moving objects with uncertainty in spatio-temporal databases. In *Proceedings of the 16th international conference on Database systems for advanced applications - Volume Part I*, DASFAA'11, pages 357–371, Berlin, Heidelberg, 2011. Springer-Verlag. 27

[28] Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25(1):1–42, March 2000. 27

[29] Martin Erwig and Markus Schneider. Spatio-temporal predicates. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):881–901, July 2002. 27

[30] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3):240–252, June 2008. 28, 67, 69, 70, 71, 72

[31] National Imagery and Mapping Agency. Department of defense world geodetic system 1984: its definition and relationships with local geodetic systems. Technical Report TR8350.2, National Imagery and Mapping Agency, St. Louis, MO, USA, January 2000. 28

[32] May Yuan and Kathleen Hornsby. *Computation and visualization for understanding dynamics in geographic domains - a research agenda*. CRC Press, 2008. 28

[33] Biadgilgn Demissie. *Geo-Visualization of Movements: Moving Objects In Static Maps, Animation And The Space-Time Cube*. VDM Verlag, 2010. 28

[34] Peter Turchin. *Quantitative Analysis of Movement: Measuring and Modeling Population Redistribution in Animals and Plants*. Sinauer, 1998. 28

[35] Somayeh Dodge, Robert Weibel, and Patrick Laube. Exploring movement-similarity analysis of moving objects. *SIGSPATIAL Special*, 1(3):11–16, November 2009. 28, 33

[36] Natalia V. Andrienko and Gennady L. Andrienko. Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42(2):117–138, 2007. 29, 33, 35, 70, 71

[37] Jason A. Dykes and David M. Mountain. Seeking structure in records of spatio-temporal behaviour: Visualization issues, efforts and applications. *Computational Statistics and Data Analysis*, 43(4):581–603, 2003. 30

[38] Patrick Laube, Todd Dennis, Pip Forer, and Mike Walker. Movement beyond the snapshot-dynamic analysis of geospatial lifelines. *Computers, Environment and Urban Systems*, 31(5):481–501, 2007. 30

[39] Fosca Giannotti and Dino Pedreschi. *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer Publishing Company, Incorporated, 2008. 30

[40] Torsten Hägerstrand. What about people in regional science ? *Papers in Regional Science*, 24(1):6–21, 1970. 30, 31

[41] Carola Eschenbach, Christopher Habel, and Lars Kulik. Representing simple trajectories as oriented curves. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pages 431–436. AAAI Press, 1999. 30

[42] Donna J. Peuquet. Making space for time: Issues in space-time data representation. *Geoinformatica*, 5(1):11–32, March 2001. 31

[43] Kathleen Hornsby and Max J. Egenhofer. Modeling moving objects over multiple granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1):177–194, 2002. 31

[44] Marius Thériault, Christophe Claramunt, Anne-Marie Séguin, and Paul Villeneuve. Temporal gis and statistical modelling of personal lifelines. In *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, page 433, 2002. 31

[45] Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. Locally adaptive dimensionality reduction for indexing large

time series databases. *ACM Transactions on Database Systems*, 27(2):188–228, June 2002. 32

[46] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. *LNCS, Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, 730:69–84, 1993. 32, 60

[47] Helmut Alt and Leonidas J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In *Handbook of Computational Geometry*, pages 121 – 153. North-Holland, 2000. 32, 37

[48] Waldo R. Tobler. Experiments in migration mapping by computer. *Cartography and Geographic Information Science*, 14(2):155–163, 1987. 33

[49] Natalia V. Andrienko and Gennady L. Andrienko. *Exploratory analysis of spatial and temporal data - a systematic approach.* Springer, 2006. 33

[50] Robert B. McMaster. A statistical analysis of mathematical measures for linear simplification. *Cartography and Geographic Information Science*, 13(2):103–116, 1986. 35

[51] Laurent Etienne. *Motifs spatio-temporels de trajectoires d'objets mobiles, de l'extraction à la détection de comportements inhabituels. Application au traffic maritime.* PhD thesis, École Doctorale des Sciences de la Mer, Université de Bretagne Occidentale, 2012. 36, 43

[52] Felix Hausdorff. Dimension und äußeres maß. *Mathematische Annalen*, 79(1):157–179, 1918. 36

[53] Sarana Nutanong, Edwin H. Jacox, and Hanan Samet. An incremental hausdorff distance calculation algorithm. *Journal of the VLDB Endowment*, 4(8):506–517, May 2011. 36, 49, 50, 51

[54] Helmut Alt and Michael Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computing Geometry and Applications*, 5:75–91, 1995. 37, 39

[55] Boris Aronov, Sariel Har-Peled, Christian Knauer, Yusu Wang, and Carola Wenk. Fréchet distance for curves, revisited. *Algorithms–ESA 2006*, pages 52–63, 2006. 37

[56] Helmut Alt. The computational geometry of comparing shapes. In *Efficient Algorithms*, volume 5760, pages 235–248. Springer Berlin - Heidelberg, 2009. 38

[57] Maurice Fréchet. Sur l'écart de deux courbes et sur les courbes limites. *Transactions of the American Mathematical Society*, 6(4):435–449, 1905. 38

[58] Maurice Fréchet. Relations entre les notions de limite et de distance. *Transactions of the American Mathematical Society*, 19(1):53–65, 1918. 38

[59] Maurice Fréchet. L'expression la plus generale de la "distance" sur une droite. *American Journal of Mathematics*, 47(1):1–10, 1925. 38

[60] Maurice Fréchet. Definition of the probable deviation. *The Annals of Mathematical Statistics*, 18(2):288–290, 1947. 38

[61] Alon Efrat, Leonidas J. Guibas, Sariel Har-Peled, Joseph S. B. Mitchell, and T. M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, 28(4):535–569, 2002. 39

[62] Helmut Alt and Michael Godau. Measuring the resemblance of polygonal curves. In *Proceedings of the eighth annual symposium on Computational geometry*, SCG '92, pages 102–109, New York, NY, USA, 1992. ACM. 39

[63] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical report, 1994. 39, 40

[64] Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the fréchet distance. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 645–654, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics. 39

[65] Ariane Mascret, Thomas Devogele, Iwan Le Berre, and Alain Hénaff. Coastline matching process based on the discrete fréchet distance. *Progress in Spatial Data Handling*, pages 383–400, 2006. 40, 41

[66] Thomas Devogele. A new merging process for data integration based on the discrete fréchet distance. In *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, pages 167–181, 2002. 41

[67] Michail Vlachos, Dimitrios Gunopoulos, and George Kollios. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 673–, Washington, DC, USA, 2002. IEEE Computer Society. 42, 48, 60

[68] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, pages 491–502, New York, NY, USA, 2005. ACM. 42

[69] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, pages 419–429. ACM, 1994. 42

[70] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008. 42, 48, 51

[71] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 678–684, New York, NY, USA, 2005. ACM. 42, 92

[72] Hioraki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978. 42, 44

[73] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. 43

[74] Sang-Wook Kim, Sanghyun Park, and Wesley W. Chu. Efficient processing of similarity search under time warping in sequence databases:

An index-based approach. *Information Systems*, 29(5):405–420, 2004. 44

[75] Eamonn Keogh and Chotirat A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005. 44

[76] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, page 707, 1966. 46

[77] Maxime Crochemore and Wojciech Rytter. *Text algorithms*. Oxford University Press, USA, 1994. 47

[78] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, 2003. 47

[79] Tolga Bozkaya, Nasser Yazdani, and Meral Özsoyoglu. Matching and indexing sequences of different lengths. In *Proceedings of the sixth international conference on Information and knowledge management*, pages 128–135, 1997. 47

[80] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803, 2004. 47

[81] Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney, and Kyuseok Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, 1995. 47

[82] Yuhan Cai. Indexing spatiotemporal trajectories with chebyshev polynomials. Master's thesis, The University of British Columbia, 2002. 47

[83] Zaiben Chen, Heng Tao Shen, Xiaofang Zhou, Yu Zheng, and Xing Xie. Searching trajectories by locations: an efficiency study. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management*

of data, SIGMOD '10, pages 255–266, New York, NY, USA, 2010. ACM. 49

[84] Helmut Alt, Bernd Behrends, and Johannes Blömer. Approximate matching of polygonal shapes (extended abstract). In *Proceedings of the seventh annual symposium on Computational geometry*, SCG '91, pages 186–193, New York, NY, USA, 1991. ACM. 49

[85] Hui Ding, Goce Trajcevski, and Peter Scheuermann. Efficient similarity join of large sets of moving object trajectories. In *Proceedings of the 2008 15th International Symposium on Temporal Representation and Reasoning*, pages 79–87. IEEE Computer Society, 2008. 51

[86] Gaurav Sinha and David M. Mark. Measuring similarity between geospatial lifelines in studies of environmental health. *Journal of Geographical Systems*, 7(1):115–136, 2005. 51

[87] Marc Van Kreveld and Jun Luo. Trajectory similarity of moving objects. In *GI Days - Young Reasearchers Forum*, 2007. 51

[88] Goce Trajcevski, Hui Ding, Peter Scheuermann, Roberto Tamassia, and Dennis Vaccaro. Dynamics-aware similarity of moving objects trajectories. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 11, 2007. 51

[89] Slava Kisilevich, Florian Mansmann, Mirco Nanni, and Salvatore Rinzivillo. Spatio-temporal clustering. In *Data Mining and Knowledge Discovery Handbook*, pages 855–874. 2010. 54

[90] Oded Maimon and Lior Rokach, editors. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* Springer, 2010. 54, 59

[91] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd International Conference on Knowledge Discovery and*, pages 226–231, 1996. 55

[92] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbscan

and its applications. *Data Mining Knowledge Discovering*, 2(2):169–194, June 1998. 55, 56

[93] Mirco Nanni and Dino Pedreschi. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems*, 27(3):267–289, November 2006. 55

[94] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *The 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967. 56

[95] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, pages 426–435, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. 58

[96] Elias Frentzos, Kostas Gratsias, and Yannis Theodoridis. Index-based most similar trajectory search. In Rada Chirkova, Asuman Dogac, M. Tamer Özsu, and Timos K. Sellis, editors, *IEEE 23rd International Conference on Data Engineering, ICDE 2007*, pages 816–825. IEEE, 2007. 58

[97] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 593–604, New York, NY, USA, 2007. ACM. 59

[98] Mirco Nanni. *Clustering methods for stapio-temporal data.* PhD thesis, Dipartamento di Informatica, Università degli Studi di Pisa, 2002. 60

[99] Bart Kuijpers, Bart Moelans, Walied Othman, and Alejandro Vaisman. Analyzing trajectories using uncertainty and background information. In *Proceedings of the 11th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '09, pages 135–152, Berlin, Heidelberg, 2009. Springer-Verlag. 60

[100] Gennady Andrienko, Natalia Andrienko, and Stefan Wrobel. Visual analytics tools for analysis of movement data. *SIGKDD Explorations*

Newsletter - Special issue on visual analytics, 9:38–46, December 2007. 61

[101] Gennady L. Andrienko, Natalia V. Andrienko, Salvatore Rinzivillo, Mirco Nanni, Dino Pedreschi, and Fosca Giannotti. Interactive visual clustering of large collections of trajectories. In *IEEE Symposium on Visual Analytics Science and Technology, VAST*, pages 3–10, 2009. 61, 62

[102] Tobias Schreck, Jürgen Bernard, Tatiana Von Landesberger, and Jörn Kohlhammer. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization*, 8(1):14–29, January 2009. 62

[103] Teuvo Kohonen, Manfred R. Schroeder, and Thomas S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition, 2001. 62

[104] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, and Gennady Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7:225–239, June 2008. 62

[105] Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. *J. Mach. Learn. Res.*, 4:1001–1037, December 2003. 63

[106] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 64

[107] Shivakumar Vaithyanathan and Byron Dom. Model-based hierarchical clustering. In *In Proceedings 16th Conference on Uncertainty in Artificial Intelligence*, pages 599–608. UAI, 2000. 64

[108] Marina Meilă and David Heckerman. An experimental comparison of model-based clustering methods. *Journal on Machine Learning*, 42(1-2):9–29, January 2001. 64

[109] Brian S. Everitt and D. J. Hand. *Mixture Models: Inference and Applications to Clustering*. Chapman and Hall, London, 1981. 64

[110] Shi Zhong. *Probabilistic model-based clustering of complex data*. PhD thesis, 2003. 64

[111] Volodymyr Melnykov. Efficient estimation in model-based clustering of gaussian regression time series. *Journal on Statistical Analysis and Data Mining*, 5(2):95–99, April 2012. 64

[112] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 63–72, New York, NY, USA, 1999. ACM. 64

[113] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, December 2005. 64

[114] Evangelos Dermatas and George Kokkinakis. Algorithm for clustering continuous density hmm by recognition error. *IEEE Transactions on Speech and Audio Processing*, 4(3):231 –234, may 1996. 64

[115] Marco Ramoni, Paola Sebastiani, and Paul Cohen. Bayesian clustering by dynamics. *Journal on Machine Learning*, 47(1):91–121, April 2002. 64

[116] Carole Beal, Sinjini Mitra, and Paul R. Cohen. Modeling learning patterns of students with a tutoring system using hidden markov models. In *Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 238–245, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press. 64

[117] Darya Chudova, Scott Gaffney, Eric Mjolsness, and Padhraic Smyth. Translation-invariant mixture models for curve clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 79–88, New York, NY, USA, 2003. ACM. 64

[118] Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering clusters in motion time-series data. In *Proceedings of the*

2003 IEEE computer society conference on Computer vision and pattern recognition, CVPR'03, pages 375–381, Washington, DC, USA, 2003. IEEE Computer Society. 64

[119] Marc Benkert, Joachim Gudmundsson, Florian Hübner, and Thomas Wolle. Reporting flock patterns. *Computational Geometry: Theory and Applications*, 41(3):111–125, November 2008. 64, 65

[120] Hoyoung Jeung, Heng Tao Shen, and Xiaofang Zhou. Convoy queries in spatio-temporal databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 1457–1459, Washington, DC, USA, 2008. IEEE Computer Society. 64, 65

[121] Zhenhui Li. Incremental clustering for trajectories. Master's thesis, University of Illinois at Urbana-Champaign, 2010. 66, 67, 144

[122] Yifan Li, Jiawei Han, and Jiong Yang. Clustering moving objects. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 617–622, New York, NY, USA, 2004. ACM. 66

[123] San-Yih Hwang, Ying-Han Liu, Jeng-Kuen Chiu, and Ee-Peng Lim. Mining mobile group patterns: a trajectory-based approach. In *Proceedings of the 9th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'05, pages 713–718, Berlin, Heidelberg, 2005. Springer-Verlag. 66

[124] Tao Li and Sarabjot S. Anand. Hirel: An incremental clustering algorithm for relational datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, ICDM '08, pages 887–892, Washington, DC, USA, 2008. IEEE Computer Society. 67, 144

[125] Hoyoung Jeung, Man Lung Yiu, and Christian S. Jensen. Trajectory pattern mining. In *Computing with Spatial Trajectories. Yu Zheng and Xiaofang Zhou (editors)*, pages 143–177. 2011. 67, 69

[126] Yida Wang, Ee-Peng Lim, and San-Yih Hwang. Efficient mining of group patterns from user movement data. *Journal of Data & Knowledge Engineering*, 57(3):240–282, June 2006. 70

[127] Patrick Laube and Stephan Imfeld. Analyzing relative motion within groups of trackable moving point objects. In *Proceedings of the Second International Conference on Geographic Information Science*, GIScience '02, pages 132–144, London, UK, UK, 2002. Springer-Verlag. 70, 71, 72

[128] Rita De Caluwe, Guy de Tré, and Gloria Bordogna. *Spatio-Temporal Databases: Flexible Querying and Reasoning*. Springer, 2010. 70

[129] Jun Wook Lee, Ok Hyun Paek, and Keun Ho Ryu. Temporal moving pattern mining for location-based service. *Journal of Systems and Software*, 73(3):481–490, November 2004. 70

[130] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA, 1995. IEEE Computer Society. 70

[131] Joachim Gudmundsson and Marc van Kreveld. Computing longest duration flocks in trajectory data. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, GIS '06, pages 35–42, New York, NY, USA, 2006. ACM. 70

[132] Joachim Gudmundsson, Patrick Laube, and Thomas Wolle. Movement patterns in spatio-temporal data. In *Encyclopedia of GIS*, pages 726–732. 2008. 70

[133] Joachim Gudmundsson, Marc van Kreveld, and Bettina Speckmann. Efficient detection of motion patterns in spatio-temporal data sets. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, GIS '04, pages 250–257, New York, NY, USA, 2004. ACM. 71, 72

[134] Patrick Laube. *Analysing point motion - spatio-temporal data mining of geospatial lifelines*. PhD thesis, 2005. 71, 72

[135] Philip W. Blythe, Geoffrey F. Miller, and Peter M. Todd. Human simulation of adaptive behavior: Interactive studies of pursuit, evasion, courtship, fighting, and play. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 13–22. MIT Press/Bradford Books, 1996. 72

[136] Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM. 72

[137] Vania Bogorny and Monica Wachowicz. A framework for context-aware trajectory. *Data Mining for Business Applications*, pages 225–239, 2009. 75

[138] *Community and Society*. Dover Publications, November 2011. 76

[139] Nicholas C. Mullins and Carolyn J. Mullins. *Theories and theory groups in contemporary American sociology*. Harper & Row, 1973. 76

[140] Sajid S. Shaikh. Computations in social networks. Master's thesis, Kent State University, 2007. 77

[141] Peter Kollock. *The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace*, chapter 9, pages 220–239. Routledge, 11 New Fetter Lane, London EC4P 4EE, 1999. 77

[142] Barry Wellman. Computer networks as social networks. *Science*, 293(5537):2031–2034, 2001. 77

[143] Eric Chuk, Rama Hoetzlein, David Kim, and Julia Panko. Creating socially networked knowledge through interdisciplinary collaboration. *Arts and Humanities in Higher Education*, 11(1-2):93–108, FEB/APR 2012. 77

[144] Andrea Kavanaugh. The impact of computer networking on community: A social network analysis approach. 1999. 77

[145] J. David Johnson. *Managing Knowledge Networks*. Cambridge University Press, New York, NY, USA, 1st edition, 2009. 77

[146] Yutaka Matsuo. Social network and spatial semantics for real-world information service. In Toru Ishida, Les Gasser, and Hideyuki Nakashima, editors, *Massively Multi-Agent Systems*, volume 3446 of *Lecture Notes in Computer Science*, pages 573–573. Springer Berlin / Heidelberg, 2005. 78

[147] Zeqian Shen and Kwan-Liu Ma. Mobivis: A visualization system for exploring mobile data. In *PacificVis*, pages 175–182, 2008. 78

[148] Yannis Manolopoulos, Jaroslav Pokorny, and Timos K. Sellis, editors. *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006, Thessaloniki, Greece, September 3-7, 2006, Proceedings*, volume 4152 of *Lecture Notes in Computer Science*. Springer, 2006. 81, 114

[149] Geoffrey Benjamin Zenger. Trajectory-based point of interest recommendation. Master's thesis, School of Computing Science, Simon Fraser University, 2009. 82

[150] Yu Zheng, Yukun Chen, Xing Xie, and Wei-Ying Ma. Geolife2.0: A location-based social networking service. In *Mobile Data Management*, pages 357–358, 2009. 83, 87

[151] Ramaswamy Hariharan and Kentaro Toyama. Project lachesis: Parsing and modeling location histories. In *GIScience*, pages 106–124, 2004. 83

[152] Lars Backstrom and Jure Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM. 83

[153] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, GIS '08, pages 34:1–34:10, New York, NY, USA, 2008. ACM. 84

[154] Xin Cao, Gao Cong, and Christian S. Jensen. Mining significant semantic locations from gps data. *VLDB Endowment*, 3(1-2):1009–1020, September 2010. 84, 87

[155] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 442–445, New York, NY, USA, 2010. ACM. 84

[156] Vincent W. Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Conference on Artificial Intelligence, AAAI*, 2010. 85

[157] Yuki Arase, Xing Xie, Takahiro Hara, and Shojiro Nishio. Mining people's trips from large scale geo-tagged photos. In *Proceedings of the international conference on Multimedia*, MM '10, pages 133–142, New York, NY, USA, 2010. ACM. 87

[158] Xin Lu, Changhu Wang, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the international conference on Multimedia*, MM '10, pages 143–152, New York, NY, USA, 2010. ACM. 87

[159] Liliana Ardissono, Anna Goy, Giovanna Petrone, and Marino Segnan. A multi-agent infrastructure for developing personalized web-based systems. *ACM Transactions on Internet Technologies*, 5(1):47–69, February 2005. 89

[160] Simon Dunstall, Mark E. T. Horn, Philip Kilby, Mohan Krishnamoorthy, Bowie Owens, David Sier, and Sylvie Thiébaux. An automated itinerary planning system for holiday travel. *Journal of IT and Tourism*, 6(3):195–210, 2003. 89

[161] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 35–44, New York, NY, USA, 2010. ACM. 90

[162] P. Kumar, V. Singh, and D. Reddy. Advanced traveler information system for hyderabad city. *Transactions on Intelligent Transportation Systems*, 6(1):26–37, March 2005. 90

[163] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. 91, 92

[164] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 259–266, New York, NY, USA, 2003. ACM. 91

[165] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005. 92, 124

[166] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM. 92

[167] Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining, SDM*, 2005. 92

[168] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003. 92

[169] Salem Hadim and Nader Mohamed. Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE Distributed Systems Online*, 7(3):1–, March 2006. 99

[170] Kristian Ellebaek Kjaer. A survey of context-aware middleware. In *Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering*, SE'07, pages 148–155, Anaheim, CA, USA, 2007. ACTA Press. 99

[171] Reinaldo B. Braga, Ali Tahir, Michela Bertolotto, and Hervé Martin. Clustering user trajectories to find patterns for social interaction applications. In *11th International Conference on Web and Wireless Geographical Information Systems (W2GIS'12)*, W2GIS '12, Naples, IT, 2012. Lecture Notes in Computer Science - Springer. 103

[172] Cedric du Mouza and Philippe Rigaux. Multi-scale classification of moving objects trajectories. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, pages 307–, Washington, DC, USA, 2004. IEEE Computer Society. 105

[173] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S. Tseng. Mining user similarity from semantic trajectories. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, LBSN '10, pages 19–26, New York, NY, USA, 2010. ACM. 105

[174] Jean-Claude Muller, Jean-Philippe Lagrange, and Robert Weibel. *GIS And Generalisation: Methodology And Practice (Gisdata, No 1)*. CRC Press, 1 edition, apr 1995. 105

[175] Nirvana Meratnia and Rolf A. de By. Aggregation and comparison of trajectories. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, GIS '02, pages 49–54, New York, USA, 2002. ACM. 105

[176] Changxiu Cheng, Feng Lu, and Jun Cai. A quantitative scale-setting approach for building multi-scale spatial databases. *Journal of Computers and Geosciences*, 35:2204–2209, November 2009. 105

[177] Sheng Zhou and Christopher B. Jones. Design and implementation of multi-scale databases. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '01, pages 365–386, London, UK, UK, 2001. Springer-Verlag. 106

[178] Thi Hong Nhan Vu, Keun Ho Ryu, and Namkyu Park. A method for predicting future location of mobile user for location-based services system. *Comput. Ind. Eng.*, 57:91–105, August 2009. 106

[179] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data(base) Engineering Bulletin*, 33(2):32–39, 2010. 106

[180] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD in-*

ternational conference on Knowledge discovery and data mining, KDD '07, pages 330–339, New York, NY, USA, 2007. ACM. 106

[181] Dimitris Papadias, Timos Sellis, Yannis Theodoridis, and Max J. Egenhofer. Topological relations in the world of minimum bounding rectangles: a study with r-trees. *ACM SIGMOD International Conference on Management of Data*, 24:92–103, May 1995. 116

[182] Mikhail J. Atallah. A linear time algorithm for the hausdorff distance between convex polygons. *Informatics Processing Letters*, 17(4):207–209, 1983. 116

[183] Edwin H. Jacox and Hanan Samet. Metric space similarity joins. *ACM Transaction on Database Systems*, 33:7:1–7:38, June 2008. 116

[184] Stephan Bischoff, Darko Pavic, and Leif Kobbelt. Automatic restoration of polygon models. *ACM Transactions on Graphics*, 24:1332–1352, October 2005. 116

[185] Emmanuel Parzen. On the estimation of a probability density function and the mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, September 1962. 121

[186] P. Wand and C. Jones. *Kernel Smoothing*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1995. 121

[187] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1986. 121

[188] W. Martinez. *Computational statistics handbook with MATLAB*. Chapman & Hall/CRC, 2001. 121

[189] Stefan Brecheisen, Hans-Peter Kriegel, Peer Kröger, and Martin Pfeifle. Visually mining through cluster hierarchies. In *International Conference on Data Mining, Orlando, FL*. Citeseer, 2004. 129

[190] Joseph Fourier University, Atomic Energie Commission (CEA), Floraris and RM Fora Marine. Zeroco2 project, September 2010. 135, 158

[191] William E. May and Leonard. Holder. *A history of marine navigation*. Foulis, Henley on Thames,, 1973. 135, 158

[192] Mike Dean and Guus Schreiber. OWL web ontology language reference. W3C recommendation, W3C, February 2004. 136, 161

[193] Apple Inc. Instruments for Performance Analysis - Version 2.7, September 2010. 137, 164

[194] Facebook. Facebook developer platform, Jun 2011. 139

[195] Marco D. Adelfio, Sarana Nutanong, and Hanan Samet. Similarity search on a large collection of point sets. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 132–141, New York, NY, USA, 2011. ACM. 145

[196] José Bringel Filho. *CxtBAC : Une Famille de Modeles de Controle d'acces Sensible au Contexte pour les Environnements Pervasifs*. PhD thesis, Université Joseph Fourier, Grenoble, 2010. 145

[197] Maria Luisa Damiani, Elisa Bertino, Barbara Catania, and Paolo Perlasca. Geo-rbac: A spatially aware rbac. *ACM Trans. Inf. Syst. Secur.*, 10(1), February 2007. 145

[198] Windson Viana, Alina Miron, Bogdan Moisuc, Jerome Gensel, Marlene Villanova-Oliver, and Hervé Martin. Towards the semantic and context-aware management of mobile multimedia. *Multimedia Tools and Applications*, pages 1–39, 2010. 10.1007/s11042-010-0502-6. 150, 159

[199] Neil O'Hare and Alan F. Smeaton. Context-aware person identification in personal photo collections. *Transactions on Multimedia*, 11(2):220–228, 2009. 150, 152, 158, 163

[200] Hugo de Figueiredo, Yuri Lacerda, Anselmo de Paiva, Marco Casanova, and Claudio de Souza Baptista. Photogeo: a photo digital library with spatial-temporal support and self-annotation. *Multimedia Tools and Applications*, pages 1–27, 2011. 10.1007/s11042-011-0745-x. 150, 152

[201] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *HUC*, pages 304–307, 1999. 151

[202] Fabiana G. Marinho, Fabrício Lima, João B. F. Filho, Lincoln S. Rocha, Marcio E. F. Maia, Saulo B. de Aguiar, Valéria L. L. Dantas, Windson Viana, Rossana M. C. Andrade, and Eldânae Teixeira. A software product line for the mobile and context-aware applications domain. In *SPLC*, pages 346–360, 2010. 151

[203] Mor Naaman, Susumu Harada, QianYing Wang, Hector Garcia-Molina, and Andreas Paepcke. Context data in geo-referenced digital photo collections. In *ACM Multimedia*, pages 196–203, 2004. 151, 152

[204] Lyndon S. Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 297–306, New York, NY, USA, 2008. ACM. 152

[205] Mischa M. Tuffield, Stephen Harris, Christopher Brewster, Nicholas Gibbins, Fabio Ciravegna, Derek Sleeman, Nigel R. Shadbolt, and Yorick Wilks. Image annotation with photocopain. In *Proceedings of Semantic Web Annotation of Multimedia (SWAMM-06) Workshop at the World Wide Web Conference 06. WWW*, pages 22–26, 2006. 152

[206] Windson Viana, José Bringel Filho, Jérôme Gensel, Marlène Villanova Oliver, and Hervé Martin. Photomap - automatic spatiotemporal annotation for mobile photos. In *W2GIS'07: Proceedings of the 7th international conference on Web and wireless geographical information systems*, pages 187–201, Berlin, Heidelberg, 2007. Springer-Verlag. 152, 159

[207] Morgan Ames. Why we tag: motivations for annotation in mobile and online media. In *In CHI ?07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 971–980. ACM Press, 2007. 152

[208] Mika Raento, Antti Oulasvirta, Renaud Petit, and Hannu Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4:51–59, 2005. 152, 153

[209] Louise Barkhuus, Barry Brown, Marek Bell, Scott Sherwood, Malcolm Hall, and Matthew Chalmers. From awareness to repartee: sharing location within social groups. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 497–506, New York, NY, USA, 2008. ACM. 153

[210] Johan Koolwaaij, Anthony Tarlano, Marko Luther, Petteri Nurmi, Bernd Mrohs, Agathe Battestini, and Raju Vaidya. Context Watcher-Sharing context information in everyday life. Calgary, Canada, July 2006. 153

[211] Hongzhi Li and Xian-Sheng Hua. Melog: mobile experience sharing through automatic multimedia blogging. In *Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing*, MCMC '10, pages 19–24, New York, NY, USA, 2010. ACM. 153

[212] Pujianto Cemerlang, Joo-Hwee Lim, Yilun You, Jun Zhang, and Jean-Pierre Chevallet. Towards automatic mobile blogging. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 2033 –2036, july 2006. 153

[213] Dong-Sung Ryu, Woo-Keun Chung, and Hwan-Gue Cho. Photoland: a new image layout system using spatio-temporal information in digital photos. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1884–1891, New York, NY, USA, 2010. ACM. 158, 163

[214] Google Inc. The Google Maps/Google Earth APIs, September 2010. 163

LIDU: Location-based approach to IDentify similar interests between Users in social networks

Abstract: Sharing of user data has substantially increased over the past few years facilitated by sophisticated Web and mobile applications, including social networks. For instance, users can easily register their trajectories over time based on their daily trips captured with GPS receivers as well as share and relate them with trajectories of other users. Analyzing user trajectories over time can reveal habits and preferences. This information can be used to recommend content to single users or to group users together based on similar trajectories and/or preferences. Recording GPS tracks generates very large amounts of data. Therefore clustering algorithms are required to efficiently analyze such data. In this thesis, we focus on investigating ways of efficiently analyzing user trajectories, extracting user preferences from them and identifying similar interests between users. We demonstrate an algorithm for clustering user GPS trajectories. In addition, we propose an algorithm to correlate trajectories based on near points between two or more users. The final results provided interesting avenues for exploring Location-based Social Network (LBSN) applications.

Keywords: Location-Based Social Networks (LBSN), Geographic Information Systems (GIS), social networks, similarity analysis, clustering algorithms, Points of Interest (PoI).

LIDU: Une approche basée sur la localisation pour l'identification de similarités d'intérêts entre utilisateurs dans les réseaux sociaux.

Résumé: Grâce aux technologies web et mobiles, le partage de données entre utilisateurs a considérablement augmenté au cours des dernières années. Par exemple, les utilisateurs peuvent facilement enregistrer leurs trajectoires durant leurs déplacements quotidiens avec l'utilisation de récepteurs GPS et les mettre en relation avec les trajectoires d'autres utilisateurs. L'analyse des trajectoires des utilisateurs au fil du temps peut révéler des habitudes et préférences. Cette information peut être utilisée pour recommander des contenus à des utilisateurs individuels ou à des groupes d'utilisateurs avec des trajectoires ou préférences similaires. En revanche, l'enregistrement de points GPS génère de grandes quantités de données. Par conséquent, les algorithmes de clustering sont nécessaires pour analyser efficacement ces données. Dans cette thèse, nous nous concentrons sur l'étude des différentes solutions pour analyser les trajectoires, extraire les préférences et identifier les intérêts similaires entre les utilisateurs. Nous proposons un algorithme de clustering de trajectoires GPS. En outre, nous proposons un algorithme de corrélation basée sur les trajectoires des points proches entre deux ou plusieurs utilisateurs. Les résultats finaux ouvrent des perspectives intéressantes pour explorer les applications des réseaux sociaux basés sur la localisation.

Mots-clés: Réseaux Sociaux Basés sur la Localisation (LBSN), Systeme d'Information Geographique (SIG), Réseaux sociaux, Analyse de similarités, Algorithmes de clustérisation, Points d'Intérêt (PoI).