



Sixth Framework Programme
Information Society Technologies
Future Emerging Technologies



Project Acronym:

GeoPKDD

Project full title:

Geographic Privacy-aware Knowledge Discovery and Delivery

Project Number: FP6-014915

Instrument: Specific Target Research Project

Project Deliverable D1.3

Design of the Trajectory Warehouse Architecture

Date of preparation: 08/12/2006

Revision: final

Operative commencement date of Contract: 01/12/2005

Project Coordinator: Fosca Giannotti

Project institute coordinator: Knowledge Discovery and Delivery-LAB / ISTI-CNR

Programme Name:IST
Project Number:.....014915
Project Title:GEOPKDD

Document Number:D1.3
Work-Package:.....WP1
Document Type:Deliverable
Contractual Date of Delivery:01/12/2006
Actual Date of Delivery:08/12/2006
Title of Document:Design of the Trajectory Warehouse Architecture
Author(s):M. L. Damiani, C. Vangenot, EPFL and Univ. Milano E. Frentzos, G. Marketos, I. Ntoutsis, N. Pelekis, Y. Theodoridis, V. Verykios, A. Raffaeta

Dissemination LevelPublic

GeoPKDD consortium participants

Partic. Role	Partic. N.	Participant name	Short name	Country
CO	1	KDD Lab. joint research group of ISTI-CNR , Istituto di Scienza e Tecnologie dell'Informazione, Pisa. http://www-kdd.isti.cnr.it/ and Univ. Pisa , Dept. of Computer Science http://www.di.unipi.it	KDDLAB	I
CR	2	Univ. of Hasselt , Theoretical Computer Science Group. http://alpha.uhasselt.be/research/groups/theocomp/	HASSELT	B
CR	3	EPFL - Ecole Polytechnique Fédérale de Lausanne , Lab. DB, Lausanne. http://lbdwww.epfl.ch/e/ and University of Milan - Computer Science and Communication Department http://www.dico.unimi.it/	EPFL	CH
CR	4	Fraunhofer Institute for Autonomous Intelligent Systems , Sankt Augustin. http://www.ais.fraunhofer.de/	FAIS	D
CR	5	Wageningen UR , Centre for GeoInformation. http://cgi.girs.wageningen-ur.nl/	WUR	NL
CR	6	Research Academic Computer Technology Institute , Research and Development Division. http://www.cti.gr/ and Univ. Piraeus, Dept. of Informatics http://www.unipi.gr	CTI	GR
CR	7	Sabanci University , Faculty of Engineering and Natural Sciences. http://www.sabanciuniv.edu/	UNISAB	TK
CR	8	WIND Telecomunicazioni SpA , Direzione Reti Wind Progetti Finanziati & Technology Scouting. http://www.wind.it	WIND	I

DOCUMENT CONTENTS

1	INTRODUCTION.....	1
1.1	ENVISIONED ARCHITECTURE.....	3
1.2	STRUCTURE OF THE DELIVERABLE.....	3
2	BACKGROUND ISSUES	3
2.1	TRAJECTORY DB MODELING	3
2.2	GEOPKDD DATA REQUIREMENTS	3
2.3	THE GEOPKDD DATA SYNTHESIZER.....	3
2.4	THE GEOPKDD TRAJECTORY STREAM MANAGER	3
2.5	TRAJECTORY DATABASE MANAGEMENT.....	3
3	RELATED WORK	3
3.1	MODELING	3
3.1.1	<i>Spatial dimensions</i>	3
3.1.2	<i>Spatial measures</i>	3
3.2	AGGREGATION FUNCTIONS AND THEIR IMPLEMENTATION	3
3.3	OTHER PROPOSALS.....	3
4	TDW FOR GEOPKDD PURPOSES.....	3
4.1	TDW MODELING	3
4.1.1	<i>Thematic, spatial and temporal measures</i>	3
4.1.2	<i>Thematic, spatial and temporal dimensions</i>	3
4.1.3	<i>Hierarchies on dimensions</i>	3
4.2	OLAP ISSUES	3
4.3	FEEDING AND OPERATING THE TDW.....	3
4.3.1	<i>ETL process</i>	3
4.3.2	<i>OLAP Servers</i>	3
5	FURTHER ISSUES	3
5.1	MULTIPLE SPATIAL TOPOLOGIES.....	3
5.2	MULTIPLE REPRESENTATION OF TRAJECTORIES	3
5.3	TRAJECTORY COMPRESSION.....	3
5.4	UNCERTAINTY ISSUES	3
5.5	SECURITY AND PRIVACY IN A MOBILE CONTEXT.....	3
5.5.1	<i>The GEO-RBAC model: an overview</i>	3
5.5.2	<i>The K-Anonymity Model for Spatiotemporal Data</i>	3
6	CONCLUSIONS	3
7	REFERENCES	3

1 Introduction

The GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery) project aims at extracting user-consumable forms of knowledge from large amounts of

raw geographic data referenced in space and in time, also taking into account privacy issues. Figure 1 illustrates the GeoPKDD concept at a glance.

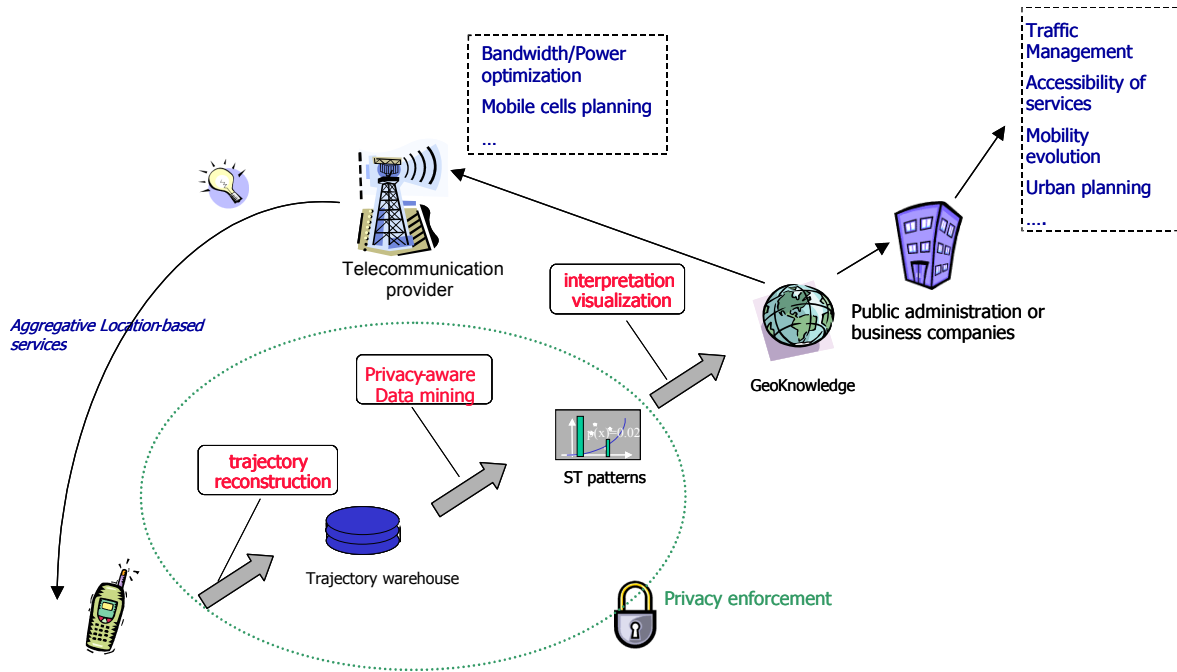


Figure 1: The GeoPKDD concept

Specifically, WP1 focuses on the design and development of an efficient and effective privacy-aware Trajectory Data Warehouse (TDW); efficient by means that it considers complexity issues and effective by means that it provides useful information to the end users.

DWs have received considerable attention of the database community as a technology for integrating all sorts of transactional data, dispersed within organizations whose applications utilize either legacy (non-relational) or advanced relational database systems. DWs form a technological framework for supporting decision-making processes by providing informational data. A Data Warehouse (DW) is defined as a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management of decision making process [Inm96].

In a DW, data are organized and manipulated in accordance with the concepts and operators provided by a multidimensional data model which views data in the form of a data cube [AAD+96]. A data cube allows data to be modeled and viewed in multiple dimensions, where each dimension represents some business perspective, and is typically implemented by adopting a star (or snowflake) schema model. According to this model, the DW consists of a fact table (schematically, at the centre of the star) surrounded by a set of dimensional tables related with the fact table, which contains keys to the dimensional tables and measures.

Dimensions represent the analysis axes, while measures are the variables being analyzed over the different dimensions. For example, in the marketing domain a kind of measure is the amount of sales and dimensions may be time, location and product. Under these example assumptions, the DW stores the amount of sales for a given product in a given

region and over a given period of time. Each dimension is organized as a hierarchy (or even a set of hierarchies) of dimension levels, each level corresponding to a different granularity for the dimension. For example, year is one level of the time dimension, while the sequence *<day, month, year>* defines a simple hierarchy of increasing granularity for the time dimension. Finally, the members of a certain dimension level (e.g. the different months for the time dimension) can be aggregated to constitute the members of the next higher level (e.g. the different years). The measures are also aggregated following this hierarchy by means of an aggregation.

DWs are optimized for On-Line Analytical Processing (OLAP) operations. Typical OLAP operations include the aggregation or de-aggregation of information (called roll-up and drill-down, respectively) along a dimension, the selection of specific parts of a cube (slicing and dicing) and the reorientation of the multidimensional view of the data on the screen (pivoting) [VS99]. DWs following the paradigm of multidimensional data modeling have been widely investigated for conventional, non-spatial data. There is some initial research on spatial data warehousing where dimensions are categorized in three different types: descriptive (or thematic), temporal and spatial [BMH01], but spatio-temporal data warehousing is still in its infancy.

The motivation behind a Trajectory Data Warehouse (TDW) is to transform raw trajectories to valuable information that can be utilized for decision making purposes in ubiquitous applications, such as Location-Based Services (LBS), traffic control management, etc. Intuitively, the high volume of raw data produced by sensing and positioning technologies, the complex nature of data stored in trajectory databases, as well as the intricate and specialized query processing demands even for simple user queries, make extracting valuable information from them a hard task. As such, the idea is to extend traditional aggregation techniques so as to produce summarized trajectory information and provide OLAP style analysis.

One could mention an abundance of applications that would benefit from such a framework. Let us consider the application domain of supervision systems monitoring the road traffic (or else the movements of users) in a city and providing specialized location-aware services to their subscribers. Analysts, decision makers in the field as well as end users would rave an advantage of a prompt response to queries like “which is the total number of users moving inside a district covered by a particular set of cells at a given temporal interval?”, (here the important issue is to count the number of users, rather than getting their ids), or “which road has the highest traffic within a distance of 1Km from each hospital?” or, given an emergency call, “which is the nearest police station taking into account the current traffic condition?”, or “is there a substantial difference in the average speed of vehicles visiting downtown during weekends?”.

Each of the above mentioned queries can be answered by legacy systems; however, the computation cost and, as such, the response time is prohibitive for either real-time services or for proper analysis of the application domain. Existing approaches covered try to provide working solutions to the problem adopting ideas coming from the paradigm of spatial and spatio-temporal DWs. However, TDWs require a different approach for reasons that are described in the following paragraphs.

1.1 Envisioned architecture

The architecture envisioned so far is illustrated in Figure 2. More specifically, mobile devices are transmitting periodically the latest part of their trajectory, according to some user-defined parameters. This vast amount of data collected by all subscribed users is forwarded to a stream-based module (trajectory stream manager), whose purpose is to perform some basic trajectory preprocessing. This may include parameterized trajectory compression (so as to discard unnecessary details and concurrently keep informative abstractions of the portions of the trajectories transmitted so far), as well as techniques to handle missing/erroneous values. These trajectories are stored to a Moving Object Database (MOD) wherein appropriate querying and Extract-Transform-Load (ETL) processes are applied (possibly taking into account various types of infrastructural geodata) so as to derive information about trajectories (e.g. trajectory content in different granularities, aggregations, motional metadata etc.) to feed in the TDW. Storage of raw trajectories into the MOD and processed trajectories into the TDW are materialized through mapping constructs of the corresponding models.

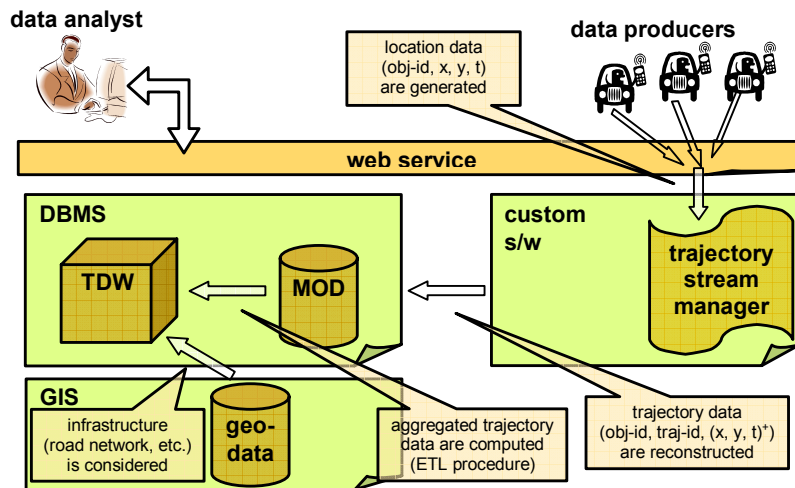


Figure 2: Trajectory warehouse architecture

A TDW serves two core needs: to provide the appropriate infrastructure for advanced reporting capabilities and to facilitate the application of trajectory mining algorithms on the aggregated data. According to end users needs, they could have access either to basic reports or OLAP-style analysis. What-if scenarios and multidimensional analysis are typical examples of analytics that could served by a TDW.

Additionally, incorporating GIS layers could result in a richer conceptual model providing thus more advanced analysis capabilities. Combining trajectory data with thematic layers (such as geographic, topographic and demographic layers) could enhance the analytics capabilities of potential applications.

Trajectory mining regards the application of data mining techniques on TDW. This task produces trajectory patterns that describe the behavior of trajectories. For instance, sequential and frequent patterns may be discovered using traditional or ad hoc pattern extraction methods.

1.2 Structure of the deliverable

In the rest of the report, we first present all the data-related modeling and management assumptions made in the context of the GeoPKDD project (Section 2), and then we describe related legacy DW approaches already proposed in the literature (Section 3). Subsequently, we present the design decisions taken so far towards setting the roadmap for the development of efficient TDW (Section 4), as well as further issues identified but still not addressed and respective privacy and security issues involved (Section 5).

2 Background issues

Before investigating specific management issues regarding trajectories of moving objects, we present the basic notion of trajectory. So, a trajectory is the description of the movement of something. Movement implies a temporal dimension as we can only perceive movement through comparison at two different instants. Therefore, a trajectory can also be equivalently defined as the record of a time-varying phenomenon. A strict definition of "movement" relates it to change in physical position. Physical movement implies an object and a reference system within which one can assess positions. Most frequently, the reference system is geographical space and we speak about objects moving in space, therefore about trajectories of objects in space. As geographical space per se is continuous, physical movement is described by a continuous change of position, i.e. a function from time to geographical space.

2.1 Trajectory DB modeling

A trajectory in the real world is the description of the movement of some object, which may be a point, a line, an area or a volume. Movement in the real world is continuous or defined over a continuum. So the modeling of a trajectory should be as a function. Indeed, an object trajectory can be seen as a triple of functions (f_x, f_y, f_a) from time t to x , y and a respectively, where a denotes the altitude. Each object has an identity, and is characterized by alphanumerical attributes and the geometric description of its trajectory. Moreover the geometric description of the trajectory is particular: it is based on four variables (time t , position in x and y , altitude a), but three out of the four variables, namely (x, y, a) , can be obtained as a continuous function of the fourth, t . Then, a finite representation of the trajectory can be supported by a set of positions $P_i(x,y,a,t)$. Figure 3 illustrates the trajectory of a moving object (P_0, P_1, P_2, P_3) that is a function of time. In the GeoPKDD context we omit the altitude a , while the in-between samples positions are approximated through linear interpolation.

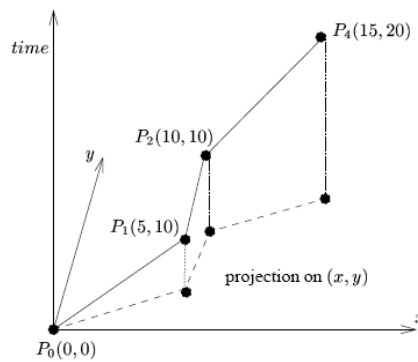


Figure 3: Trajectory representation

Once we have a function of time representing a moving object trajectory, we need to decide how to represent it into a computer system. This necessity was studied in the MOD field of research, which appeared in the research area in 90's aiming at keeping track of object locations and at supporting location-aware queries. In other words, the MOD technology has allowed one to model the movements of entities (i.e., trajectories) in a database and to pose queries about such movements.

More specifically, the database must describe not only the current state of the spatial data but also the whole history of this development. Thus it should allow to go back in time to any particular instant, to retrieve the state of the database at that time, to understand the evolution, to analyze when certain relationships were fulfilled, etc. To achieve this, we adopt an approach similar to the one initially proposed by Güting and colleagues [EGS+99, GBE+00, FGN+00, LFG+03]. In this approach, moving points are viewed as 3D objects (2D space + 1D time) whose structure and behavior is captured by modeling them as Abstract Data Types (ADT). Such types can be integrated as base (attribute) data types into relational, object-oriented, or other DBMS data models; they can be implemented as data blades, cartridges, etc. for extensible DBMSs.

2.2 GeoPKDD data requirements

The availability of real data is important for the GeoPKDD project since real world specifications could be extracted from them, thus improving the applicability of our work to real life applications. In the opposite case, the transformation of real data based on our envisioned architecture is required. To deal with either case, we have devised both a generic and extensible model for our system and adapt real or synthetic data into this model as part of some pre-processing task.

The nature of both the realistic as well as the synthetic available trajectories' datasets imply that there are two basic abstractions that one may finally adopt to conceptualize the notion of a trajectory; that of a set of exact space-time locations (e.g. $\langle (x_1, y_1, t_1), \dots, (x_n, y_n, t_n) \rangle$) or that of a set of cells (e.g. $\langle (cell_1, t_1), \dots, (cell_n, t_n) \rangle$). Both levels of abstraction should be supported with the former being more informative and the latter being closer to today's users (since GPS-equipped mobile devices are not ubiquitous yet). What is more, the model should provide mechanisms to switch between the two abstractions. CENTRE [GMP+05] is our initial development towards this direction.

Switching between these levels of abstraction is also an approach to support location privacy.

obj-id	northings	Eastings	Altitude	timestamp	other (signal strength, etc.)
123456	37.94	23.65	72.23	01/04/2006 12:32:44	...
...

obj-id	cell-id	datetime	other (signal strength, etc.)
123456	A-2001	01/04/2006 12:32:44	...
...

Figure 4: Realistic alternative raw trajectory data formats

2.3 *The GeoPKDD data synthesizer*

The objective of the GeoPKDD data synthesizer task (Task 1.2) is to produce an integration of such different approaches in generating trajectory data. The level of the integration can vary from loose to tight. A loose integration means to design a set of interface specifications to make these tools producing a common output format (standard trajectories format). A tight integration, on the other hand, means to integrate the tools in a unified software architecture and a unified user interface.

Given the current state of the art, where the tools available in the literature have been developed by different organization, for different purposes, using different languages (from Java, to C, etc), the objective of this year in the project was to design a loose integration between tools. This means to design an architecture where interfaces are defined to produce a common output format and where a set of guidelines are drawn to direct the user to the suitable tool based on his/her requirements.

The common output format is a text file in the comma separated format (CSV), where each row is a space-time position of an object. Indeed, each row is a $\langle obj-id, x, y, t \rangle$ tuple where $obj-id$ is the identifier of the object and (x, y) is the location of the object at timestamp t .

2.4 *The GeoPKDD trajectory stream manager*

The streaming model is completely suitable for moving object data since they encounter frequent updates, their volume is unexpectedly varied, and they are being processed under real-time conditions with several continuous spatiotemporal queries. Thus, MODs perfectly fit with the concept of feeding the TDW using a streaming procedure.

The main role of the *trajectory stream manager* developed in the context of the GeoPKDD project, is to deal with the following problem of trajectory reconstruction: raw location data which are reported from positioning devices in the form of $(obj-id, x, y, t)$ need to be transformed into trajectory data containing also a trajectory identifier

(*obj-id*, *traj-id*, *x*, *y*, *t*). In other words, extracting trajectories from raw location data is the goal of the stream manager developed. Determining, therefore, the trajectory identifier (which also determines the set of consecutively sampled positions of the same object forming a trajectory) is a task which requires the utilization of several – reasonable – assumptions in order to deal with real-world requirements. For example, there are scenarios where a raw location dataset including timestamps of several days must be given a single *traj-id* (e.g. emigration of birds in autumn) while in other cases it must be split into several trajectories (e.g. home-to-office movement of humans). Moreover, under certain circumstances, two consecutive - perhaps remote - locations should not be assigned to a single *traj-id*.

$\Delta t(t_i - t_{i-1})$ dist ($p_i - p_{i-1}$)	$< \Delta t_{min}$	between Δt_{min} and Δt_{max}	$> \Delta t_{max}$
$< dist_{min}$	// Noisy (too slow) movement 1. Discard p_{i-1} 2. Maintain p_i as the object's current location		// New trajectory 1. Append p_{i-1} in (the tail of) object's current trajectory
between $dist_{min}$ and $dist_{max}$	IF $dist/\Delta t > v_{max}$ THEN // Noisy (too fast) movement 1 ... 2 ... (see above) ELSE	// Normal movement 1. Append p_{i-1} in (the tail of) object's current trajectory 2. Maintain p_i as the object's current location	2. Create a new trajectory for the object and set it as the current 3. Insert p_i in (the head of) object's current trajectory
$> dist_{max}$	// Normal movement 1 ... 2 ... (see right)		4. Maintain p_i as the object's current location

Figure 5: Trajectory stream manager settings

According to the previous discussion we need to define the conditions for assigning an object a new *traj-id*. In particular, we suggest that, under the circumstances where a sufficiently large gap in the temporal dimension exists between two consecutive sampled positions, a new *traj-id* should be assigned to the later recorded position. This is due to the fact that since there is no knowledge of what has happened in between the two positions, it is safer to assume that a new trajectory has started. Moreover, GPS-sampled positions may also include noise, which should be excluded from trajectory reconstruction. A naïve approach computes the speed of the object in each segment of its motion and compares it with a commonly accepted maximum speed depending on the object's type (e.g. 200 km/h for cars). If the computed distance exceeds the maximum, the stream manager rejects the last (marked as noisy) position and waits for the next (perhaps, acceptable) position to reconstruct a new segment. Finally, due to positioning errors, stationary objects may be shown to perform a very slow movement. Therefore we need to determine the accuracy and treat stationary objects accordingly, rejecting positions reported very close to the last known objects' position. As such, the developed stream manager utilizing the positioning accuracy determines such positions, and possibly chooses to ignore and replace them with a single straight line. The previous analysis is summarized in Figure 5 determining the stream manager behaviour.

Future work in the context of GeoPKDD stream manager includes the map-matching technique analyzed in [BPS+05], which deals with the problem of assigning each sampled trajectory point to the corresponding actual network edge. In particular, considering that road networks are represented as edges in a graph, object positions

sampled by GPS or other positioning methods, will not – in general - be located on a network edge (also due to the accuracy of the positioning methods); however, in order for the trajectory to be valid under the network constraint – since this offers an advantage regarding database querying and DW operations – the need to be matched to the appropriate network edge arises.

2.5 Trajectory database management

Designing and implementing a TDW implicates a trajectory database to feed the data cube. The former is a MOD fed in by trajectories that would be the direct outcome of the preparatory processes described earlier (i.e., the trajectory stream manager). The reason to maintain such a database is to take advantage of the MOD functionality (querying, indexing) proposed in the literature and utilize it as preliminary step to extract higher level knowledge, which in turn will feed the DW.

Regarding the implementation of the MOD, our choice is to be built on top of an existing DB engine following the object-relational paradigm. In the literature, most of the current implementation efforts in the field of the MODs [DG00] (exploiting on Informix DataBlades), [PTV+06], [PT06] (exploiting on Oracle DataCartridges) adopt the object-relational paradigm as the technological development framework. Our decision was to work with the Hermes prototype system extending Oracle 10g ORDBMS [PTV+06], [PT06]).

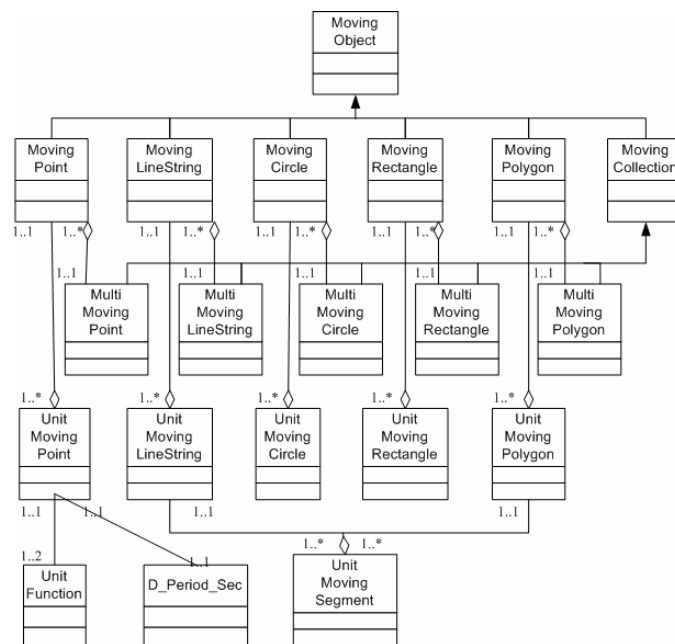


Figure 6: ADTs supporting a MOD [PTV+06]

Hermes, a database engine for handling objects that change location, shape and size, either discretely or continuously in time, has been recently proposed by Pelekis and colleagues in [PTV+06], [PT06]. Hermes can be used either as a pure temporal or a pure spatial system, but its main functionality is to support the modeling and querying of

continuously moving objects. Such a collection of data types and their corresponding operations are defined, developed and provided as an Oracle data cartridge, called Hermes Moving Data Cartridge (Hermes-MDC), which is the core component of the Hermes system architecture. By embedding the functionality offered by Hermes-MDC in Oracle DML, one obtains an expressive and easy to use query language for moving objects. In particular, Hermes-MDC defines a palette of moving object data types, illustrated in the UML class diagram of Figure 6 [PTV+06].

Considering the data requirements posed by GeoPKDD applications (recall Figure 4) and assuming a MOD engine offering the above functionality, the envisioned GeoPKDD trajectory database is illustrated in Figure 7.

- **RawLocations** (obj-id: id, timestamp: Datetime, eastings-x: Numeric, northings-y: Numeric, altitude-z: Numeric, cell-id: Alphanumeric, signal-strength: Numeric)
- **RelTrajectories** (traj-id: id, obj-id: id, timestamp: Datetime, eastings-x: Numeric, northings-y: Numeric, cell-id: Alphanumeric, signal-strength: Numeric)
- **MODTrajectories** (traj-id: id, obj-id: id, trajectory: Moving Point)
- **Objects** (obj-id: id, user-id: Alphanumeric, description: Alphanumeric, positioning: GPS/Cell)
- **Users** (user-id: id, username: Alphanumeric, gender: M/F, age: Numeric, profession: Alphanumeric, marital status: S/M)

Figure 7: The proposed GeoPKDD trajectory database schema

In particular, *RawLocations* table contains the raw information extracted by positioning devices (objects) and transmitted to the trajectory stream manager. The latter detects trajectories and adds a *traj-id* in every tuple. Hence, *RelTrajectories* table (*Rel* stands for relational since *RelTrajectories* is a typical relational table) is produced as output of the trajectory stream manager and transferred to the MOD engine, which transforms collections of $\langle x, y, t \rangle$ to trajectories defined according to the Moving Point ADT supported by Hermes (recall Figure 6); this is *MODTrajectories* table. Of course, additional information about the objects transmitting location information and the actual users recording their movement is also of interest (*Objects* and *Users* tables, respectively)

Considering Figure 7, one notices that the *MODTrajectories* table sounds redundant due to the existence of the *RelTrajectories* table. This is true but *RelTrajectories* is supported by the underlying infrastructure of the Hermes MOD engine. Indeed, a variety of database queries (e.g. thematic, range, similarity) are supported by Hermes. Except for the traditional MOD queries, novel ones [The03] are supported as well. Querying is of assistance in the trajectories' ETL process for deriving semantically rich knowledge to be fed in the warehouse. However, this is directly affected by the types of measures we are mostly interested in. Intuitively top priority queries would be those that would facilitate mining as well as privacy related lines of research. For instance, similarity queries are of direct usage in trajectory clustering/ classification tasks during the data mining phase.

3 Related Work

Research on extracting semantically rich information from raw space-time dependent data has focused on spatial and spatio-temporal DWs. As we aim to treat trajectory warehouses as a branch of spatio-temporal warehousing, the two subsequent sections present existing approaches in the area categorizing the research efforts into, on the one hand, conceptual and logical modeling methodologies, and, on the other hand, implementation issues regarding aggregation techniques as the quintessence of the data warehousing concept.

3.1 Modeling

Research on Spatial Data Warehouses (SDW) is relatively recent. Since the pioneering work of Han et al. [HSK98], several models have been proposed in the literature aiming at extending the classical DW models with spatial concepts and the OLAP tools with spatial operators (hence, SOLAP). However, despite the complexity of spatial data, current SDWs typically contain objects with simple geometric extent. Moreover, while a SDW model is assumed to consist of a set of representation concepts and an algebra of SOLAP operators for data navigation, aggregation and visualization, approaches proposed in literature often privilege either the concepts or the algebra; approaches that address both are rare. Further, whilst early data models are defined at the logical level and are based on the relational data model, in particular on the star schema model [BMH01], [SHK00], recent developments focus on conceptual aspects [JKP+04], [MZ04], [BTM05], [TPG+01].

Further, research on SDW modeling can be classified as addressing application requirements at either the logical or the conceptual data level. Mainstream solutions rely on the (logical level) relational data model [BMH01], [SHK00]. Relatively few developments focus on SDW conceptual aspects [JKP+04], [MZ04], [BTM05], [TPG+01]. The analysis presented in [Riz03] asserts the moderate interest of the research community in conceptual multidimensional modeling. However, a significant percentage of DWs fail to meet their business objectives [Riz03]. A major reason for failure is poor or inappropriate design, mainly due to a lack of established DW design methods [RG00] and DW conceptual data models [RG00].

Extending traditional DW models to deal with spatial data requires allowing both dimensions and measures to hold spatial and topological characteristics. Indeed, dimensions and measures should be extended with spatiality in order to enrich the query formulation and the visualization of the results. However adding spatiality to both dimensions and measures is not enough. SDWs have further specific requirements that have been studied in the state of the art, such as different kinds of spatial dimensions and measures, multiple hierarchies in dimensions, partial containment relationships between dimensions levels, non-normalized hierarchies, many to many relationships between measures and dimensions and the modeling of measures as complex entities [BTM05], [BMH01], [JKP+04].

3.1.1 Spatial dimensions

When adding spatiality to dimensions, most of the proposals follow the approaches by Stefanovic et al. [SHK00] and Bedard et al. [BMH01] that distinguish three types of dimension hierarchies based on the spatial references of the hierarchy members: non-geometric, geometric-to-non-geometric and fully geometric spatial dimensions. The *non-geometric spatial dimension* uses nominal spatial reference (e.g. name of cities and countries) and is treated as any other descriptive dimension [RBM01], [RBP+05]. The two other types denote dimensions where the members of lower or all levels have an associated geometry. In the *fully geometric spatial dimension*, all members of all the levels are spatially referenced while in the *geometric-to-non-geometric spatial dimension*, members are spatially referenced up to a certain dimension level and then become non-geometric. Malinowski et al. [MZ04] extend this classification and consider that a dimension can be spatial even in the absence of several related spatial levels. In their proposal, a spatial level is defined as a level for which the application keeps its spatial characteristics, meaning its geometry as this is represented by standard spatial data types (e.g. points, regions). This allows them to link the spatial levels of a dimension through topological relationships that exist between the spatial components of their members (contains, equals, overlaps, etc). Based on this, a spatial hierarchy is defined as a hierarchy that includes at least one spatial level and a spatial dimension is defined as a dimension that includes at least one spatial hierarchy. An advantage of this modeling perspective is that different spatial data types are associated with the levels of a hierarchy. For example, assuming the hierarchy *user < city < county*, a point type is associated to a user, a region is associated to a city, and a set of regions is associated to a county.

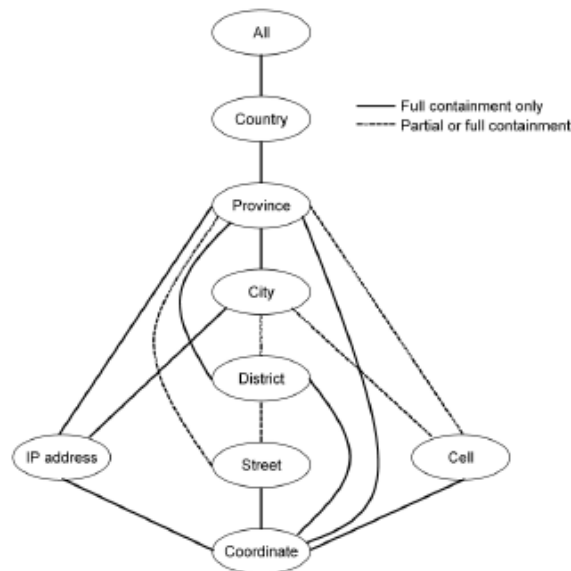


Figure 8: Hierarchy with full and partial containment relationships (from [JKP+04])

Dimensions and their organization into hierarchies are kept at a low complexity in traditional DWs. Levels of traditional non-spatial dimensions are usually organized into containment hierarchies, such as *district < city < county < country*. However when dealing with spatial data, two spatial values may not only be either disjoint or one

contained into the other, rather they may overlap each other. For instance, if we add the dimension level cell before the district level (assuming e.g. the cellular network in mobile telephony), a cell might overlap two districts. To better address application requirements, a larger spectrum of possible hierarchies is being explored. Jensen et al. [JKP+04] propose a conceptual model that supports dimensions with full or partial containment relationships (Figure 8).

As such, the dimension hierarchies can contain levels that may be linked by full or partial containment relationships. For the members of a level linked by a partial containment relationship to members of another level, the degree of containment is to be specified (e.g. 80% of this cell is contained in this district). Support for multiple hierarchies in a single dimension is also an important requirement addressed in the models by Jensen et al. [JKP+04] and by Malinowski et al. [MZ06]. It means that multiple aggregation paths are possible in a dimension (e.g. cells can be aggregated in districts or directly in counties). According to these models, multiple aggregation paths enable better handling of the imprecision in queries caused by partial containment relationships. Moreover, the above models support non-normalized hierarchies i.e., hierarchies whose members may have more than one corresponding member at the higher level or no corresponding member (e.g. a cell may be related to two districts whereas a district may be related to no cells). Finally, simple hierarchies can be characterized as: symmetrical (i.e. all levels of the hierarchy are mandatory), asymmetrical, generalized (i.e. including a generalization/specialization relationship between dimension members), non-strict (same as non-normalized) and non-covering (i.e. some levels of the hierarchy can be skipped when aggregating) [MZ06].

3.1.2 *Spatial measures*

In a similar to spatial dimensions fashion, when adding spatiality to measures, most of the proposals distinguish two types of spatial measures [SHK00], [RBM01], [RBP+05]:

- spatial measures represented by a geometry and associated with a geometric operator to aggregate it along the dimensions,
- a numerical value obtained using a topological or a metric operator.

When represented by a geometry, spatial measures consist of either a set of coordinates as in [BTM05], [MZ04], [PED01], [RBM01], [RBP+05] or a set of pointers to geometric objects as in [SHK00]. Moreover, Bimonte et al. [BTM05] and Malinowski et al. [MZ04] advocate the definition of measures as complex entities. In [BTM05], a measure is an object containing several attributes (spatial or not) and several aggregation functions (eventually, ad-hoc functions) whereas [MZ04] defines measures as attributes of an n-ary fact relationship between dimensions. This fact relationship can be spatial, if it links at least two spatial dimensions, and be associated with a spatial constraint such as, for instance, spatial containment.

An important issue related to spatial measures concerns the level of detail they are described with. Indeed, spatial data are often available and described according to various levels of detail: for instance, the same spatial object can be defined either as an area, according to a precise level of detail, or as a point, according to a less detailed level of information. This is of particular importance with trajectories where the position of the objects is subject to imprecision. Damiani et al. [DS06] propose a model

allowing to define spatial measures at different spatial granularities. This model, called MuSD, allows to represent spatial measures and dimensions in terms of OGC features. A spatial measure can represent the location of a fact at multiple levels of spatial granularity. Such multi-granular spatial measures can either be stored or dynamically computed by applying a set of coarsening operators. An algebra of SOLAP operators including special operators that allow the scaling up of spatial measures to different granularities is proposed in [DS06].

3.2 Aggregation functions and their implementation

A related research issue that has recently gained increasing interest and is relevant for the development of comprehensive SDW data models concerns the specification and efficient implementation of the operators for spatial and spatio-temporal aggregation.

Spatial aggregation operations summarize the geometric properties of objects and, as such, constitute the distinguishing aspect of SDW. Nevertheless, despite the relevance of the subject, a standard set of operators (like, for example, the Avg, Min and Max SQL operators) has not been defined yet. In fact, when defining spatial, temporal and spatio-temporal aggregates some additional problems have to be faced, which do not show up for traditional data. In particular, while for traditional databases only explicit attributes are of concern, the modeling of the spatial and temporal extent of an object makes use of interpreted attributes and the definition of aggregations is based on granularities.

A first comprehensive classification and formalization of spatio-temporal aggregate functions is presented by Lopez et al. [LS05]. The operation of aggregation is defined as a function that is applied to a collection of tuples and returns a single value. To generate the collection of tuples to which the operation is applied, the authors distinguish among three kinds of methods: group composition, partition composition and sliding window composition.

Recall that a (temporal or spatial) granularity creates a discrete image, in terms of granules, of the (temporal or spatial, respectively) domain. Given a spatial granularity GS and a temporal granularity GT, a *spatio-temporal group composition* forms groups of tuples sharing the same spatial and temporal value at granularity GS x GT. An aggregate function can then be applied to each group. On the other hand, a *spatio-temporal partition composition* is used when a finer level of aggregation is required and involves at least two granularities. The first one, which is the coarser, defines collections of tuples (the partitions). To each partition, a *sliding window composition* is performed. Instead of generating a single aggregate value for each partition, an aggregate value for every tuple in the collection at the finer granularity is computed. In order to slide through all tuples in the collection, a spatio-temporal sliding window is used.

In addition to the conceptual aspects of spatio-temporal aggregation, another major issue regards the development of methods for the efficient computation of this kind of operations to manage high volumes of spatio-temporal data. In particular, techniques are developed based on the combined use of specialized indexes, materialization of aggregate measures and computational geometry algorithms, especially to support the aggregation of dynamically computed sets of spatial objects [PTK+02, TP05, RZY+03,

ZT05]. Papadias et al. [PTK+02, TP05] propose an approach based on two types of indexes: a host index, which manages the region extents and associates to these regions an aggregate information over all the timestamps in the base relation, and some measure indexes (one for each entry of the host index), which are aggregate temporal structures storing the values of measures during the history. For a set of static regions, the authors define the *aggregate R-B-tree* (aRB-tree), which adopts an R-tree with summarized information as host index, and a B-tree containing time-varying aggregate data, as measure index.

To illustrate this concept, let us consider the regions R_1, R_2, R_3 and R_4 in Figure 9(a) and suppose that the number of phone calls initiated in $[T_1, T_5]$ inside such regions is recorded as measure in the fact table depicted in Figure 9(b). Then, Figure 9(c) illustrates the corresponding aRB-tree.

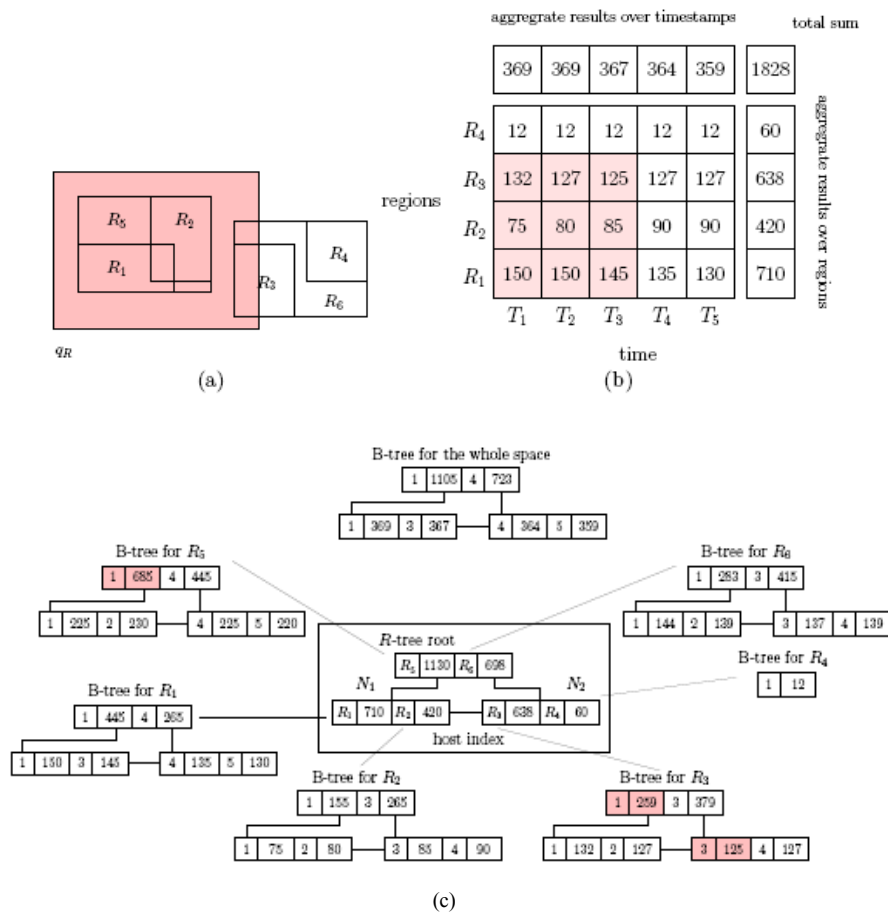


Figure 9: (a) Regions of interest, (b) a data cube example, and (c) the corresponding aRB-tree

This structure is well suited for the efficient processing of window aggregate queries, i.e., for the computation of the aggregated measure of the regions which intersect a given (spatio-temporal) window. In fact, for nodes that are totally enclosed within the query window, the summarized measure is already available thus avoiding to descend these nodes. As a consequence, the aggregate processing is made faster.

For instance, let us compute the number of phone calls inside the shaded area in Figure 9(a) during the time interval $[T_1, T_3]$. Since R_5 is completely included in the window

query there is no need to further explore R_1 and R_2 once one accesses the B-tree for R_3 . The first entry of the root of this B-tree contains the measure for the interval $[T_1, T_3]$, which is the value we are interested in. Instead, in order to obtain the sum of phone calls in the interval $[T_1, T_3]$ for R_3 one has to visit both an entry of the root of the B-tree for R_3 and also one leaf (the colored nodes in Figure 9(c)).

Tao et al. [TKC+04] showed that the aRB-tree can suffer from the distinct counting problem, i.e., if an object remains in the query region for several timestamps during the query interval, it will be counted multiple times in the result. To cope with this problem, they proposed an approach which combines spatio-temporal indexes with sketches, a traditional approximate counting technique based on probabilistic counting [FM85]. The index structure is similar to the aRB-tree: an R-tree indexes the regions of interest, whereas the B-trees record the historical sketches of the corresponding region. However, this index differs from aRB-trees in the querying algorithms since one can exploit the pruning power of the sketches to define some heuristics allowing to reduce query time.

3.3 Other proposals

A related approach includes the work by Shekhar et al. [SLC+02], who proposes a traffic DW model for the Twin-Cities metropolitan area. Although building a warehouse for traffic management, is easier than building a warehouse for trajectories (recall here that the main difficulty is that trajectories may extend to more than one cells), several interesting issues are analyzed in this work. Of particular interest is the analysis regarding the aggregate functions. More specifically, the authors aggregation functions are classified into three classes: distributive, algebraic and holistic following the work by Grey et al. [GBL+96]. An aggregate function (examples are presented in Figure 10) is classified as:

Data type	<i>Distributive</i>	<i>Algebraic</i>	<i>Holistic</i>
Set of numbers	Count, Min, Max, Sum	Average, Standard Deviation, MaxN, MinN	Median, MostFrequent, Rank
Set of points, lines, polygons	Minimal Orthogonal Bounding Box, Geometric Union, Geometric Intersection	Centroid, Center of mass, Center of gravity	Nearest neighbor, Equi-partition

Figure 10: Classification of aggregation functions [SLC+02]

- *distributive*, if a value at a specific level of hierarchy can be computed over the values of the child level;
- *algebraic*, if a value at a specific level of hierarchy can be computed using a set of aggregates over the values of the child level;
- *holistic*, if computing a value at any level of hierarchy requires access to data source.

4 *TDW for GeoPKDD purposes*

Extending traditional (i.e., non-spatial), spatial or spatio-temporal models to incorporate semantics driven by the nature of trajectories induces specific requirements as the goal is twofold: to support high level OLAP analysis and to facilitate knowledge discovery from TDW. Having this in mind, we categorize the identified requirements into modeling, analysis and management requirements. The first considers logical and conceptual level challenges introduced by TDW, the second goes over OLAP analysis requirements, while the third focuses on materialization aspects.

4.1 *TDW Modeling*

In this section we investigate the prerequisites and the constraints imposed when describing the design of a TDW from a user perspective (i.e. conceptual model), as well as when describing the final application as a system in a platform-independent tool (i.e. logical model).

While it is universally recognized that a DW leans on a multidimensional model, little is said about how to carry out its conceptual design starting from the set of user requirements [Riz03]. The domain of conceptual design for multidimensional modeling is still at a research stage. The analysis presented in [Riz03] shows the limited interest of the research community in conceptual multidimensional modeling. As stated by Malinowski et al. in [MZ06] the proposed models either provide a graphical representation based on the E-R model or UML notations with few formal definitions or only provide formal definitions without any user-oriented graphical support. Considering spatial conceptual models for DW, even fewer conceptual models have been proposed ([MZ04], [BTM05], [RBP+05], [TPG+01]) and, to the best of our knowledge, no conceptual model for TDW exists. Indeed, as already presented in Section 3.1, we argue that TDWs can not be dealt with simply modeling trajectory data in a traditional SDW: A TDW is more than a specific application. A conceptual model for a TDW entails particular requirements and among them, the following still comprise open issues:

- *Formal model*: A conceptual model for a TDW should be both spatial and temporal and it should rely on formal definitions [MZ06].
- *Complex types*: Measures should be able to be defined as *complex types*: Indeed propositions for conceptual modeling of DW very often remain very close to the star model where a fact table contains all the simple attributes to be analysed. However in real world applications, properties are complex (compound, multivalued). As by definition conceptual models are close to the way users perceive an application domain, the concepts they propose should reflect this. This need has been highlighted for traditional SDW by Bimonte et al. [BTM05] and is particularly important for trajectories that are complex objects with many spatial, temporal and thematic characteristics to analyse.
- *Hierarchies*: Different kinds of hierarchies appear in real world applications and users should be able to describe them with adapted modeling concepts.

Interesting works about complex hierarchies exist ([MZ06], [JKP+04]) but yet no consensus has been reached.

4.1.1 Thematic, spatial and temporal measures

From a modeling point of view, a trajectory is a spatial object whose location varies in time. At the same time, trajectories have thematic properties that usually are space and time dependent. This implies that different characteristics of trajectories need to be described in order to be analysed. As such, we distinguish (a) numeric characteristics, such as the average speed of the trajectory, its direction, its duration (b) spatial characteristics, such as the geometric shape of the trajectory, and (c) temporal characteristics, such as the timing of the movement. Additionally, as we pay particular attention to uncertainty and imprecision issues, a TDW model should include measures expressing the amount of uncertainty incorporated in the TDW due to raw data imprecision. Uncertainty should also be seen in granularities, while this implies that there are special aggregation operators propagating uncertainty to various levels.

In particular, depending on the application and user requirements, several numeric and spatial (or spatio-temporal) measures are considered.

I. numeric measures

1. *Number of trajectories* found in the cell (or started/ended their path in the cell; or crossed / entered / left the cell)
2. *Number of objects* (instead of counting trajectories, we can measure distinct objects-users);
3. *Number of in/out trajectories or objects* (i.e., how many times a trajectory or an object entered / left the cell)
4. ‘Motion’ measures (distinguishing between primitive and derived measures):
 - primitive ‘motion’ measures include, among others, *distance covered by trajectories* in the cell and *duration of time spent by trajectories* in the cell (applying Sum or Average aggregations);
 - derived ‘motion’ measures include, among others, *speed of trajectories*, *direction of trajectories* and *acceleration of trajectories* in the cell (applying weighted Average aggregations)

II. spatial (or spatio-temporal) measures

5. *representative trajectory* (or route) mined among the actual involved trajectories in the cell.

As a final remark about measures, it is worth to notice that the complexity of computation may vary significantly. Some measures require little pre-computation and can be stored in the DW while the observations of the various trajectories arrive. Since observations arrive in streams, it is important to decide when and what is possible to compute and store in the DW. Braz et al. [BOO+06] propose the following classification of measures according to an increasing amount of pre-calculation effort:

- (a) *no pre-computation*: the measure can be computed by using directly the attributes associated with each single observation;

- (b) *per trajectory local pre-computation*: the measure requires a pre-calculation, which involves only few and close observations of the same trajectory.
- (c) *per trajectory global pre-computation*: the measure requires a pre-calculation, which considers all the observations of each trajectory.
- (d) *global pre-computation*: the measure requires the computation of the aggregate on the total of trajectories in the data source.

For instance, the number of trajectories starting/ending their path in the cell is of type (a) supposing that there are markers which delimit the trajectories. On the other hand, the number of trajectories crossed / entered / left the cell is of type (b). Indeed, one can compute in an exact way only the sub-aggregate associated with the cell. In fact, these measures are holistic and by using the sub-aggregate it is not possible to obtain the exact super-aggregate since the identifiers of the trajectories/objects are forgotten (see [BOO+06] for details). The distance covered by and the duration of time spent by trajectories in the cell are of type (b) and these measures are distributive. As a consequence, the average speed of trajectories in the cell is still of type (b) but it is an algebraic measure since both the duration and the distance are required in order for it to be computed. Also, the number of in/out trajectories or objects is of type (b) and algebraic since more than one sub-aggregates are required in order to compute the super-aggregate (i.e., a sub-aggregate expressing the in/out value for each side of the cell).

The amount of pre-calculation associated with each type of measure has also a strong impact on the amount of memory required to buffer incoming trajectory observations. Note that, since observations may arrive in stream at different rates, and in an unpredictable and unbounded way, short processing time and small memory size are both important requirements.

Similar remarks can be found in [HSK98] where three methods are presented to compute spatial measures in spatial data cube construction. The first one consists of simply collecting and storing the corresponding spatial data but no precomputation of spatial measures is performed. Hence such a method may require more computation on-the-fly. The second method precomputes and stores some rough approximation / estimation of the spatial measures in a spatial data cube. For instance, if the measure is the merge of a set of spatial objects, one can store the Minimum Bounding Rectangle (MBR) of the merge of the objects. Finally, one can selectively precompute some spatial measures. In this case, the question is how to select a set of spatial measures for precomputation. In [HSK98], some criteria for materialization of a cuboid are presented.

4.1.2 Thematic, spatial and temporal dimensions

As starting point with respect to the supported dimensions, a trajectory data cube should support the typical spatial (e.g. coordinate roadway, cell, district, city, state, country) and temporal (e.g. datetime, hour, day, month, year) dimensions, describing the underlying spatio-temporal framework wherein trajectories are moving. Additionally, it is important to allow space-time related dimensions interact with thematic dimensions describing other user-oriented type of information like technographics (e.g. equipment used) or demographics (e.g. age and gender of users). This will allow an analyst not

only to query the data cube, for instance, about the number of objects crossed an area of interest but also to his/her analysis.

Summarizing, a flexible data cube should include thematic, spatial, temporal and spatio-temporal dimensions.

- **Temporal:** defines time durations using a granularity defined by the application scenario. Various calendars (Gregorian, financial etc) can be used to organize time into different time units for convenience.
- **Spatial:** includes geography at various levels of granularity (*coordinate, roadway, cell*, etc.).
- **Thematic (based on user profiles):** involves demographics (*gender, age, occupation, marital status*, etc.) and technographics (*tech specs*, etc.) about users.

An issue concerning the definition of dimensions is the considered level of detail for each dimension. Consider the spatial dimension: since a trajectory is actually a set of sampled locations in time, for which the in-between positions are calculated through some kind of interpolation, the lowest level information is that of spatial coordinates. This, however, implies a huge discretization of the spatial dimension, thus more abstract approaches should be evaluated. For example, cells could be used at the lowest level of spatial dimension instead of coordinates.

4.1.3 Hierarchies on dimensions

Once having defined the dimensions, a hierarchy for each dimension can be created by users or generated automatically by data clustering or data analysis. A general technique used to define hierarchies consists of discretizing the values the dimension ranges over, resulting in a set-grouping hierarchy. A partial order can thus be established among these groups of values. Let us now overview the different proposals and difficulties in creating hierarchies for the dimensions suggested in the previous subsection.

Defining hierarchies over the time dimension is straightforward, since typically there is an obvious ordering between the different levels of the hierarchy. For instance, a hierarchy that could be used on top of the (absolute) datetime is *datetime < date < month < quarter < year*. Other hierarchies over the time dimension could concern *day-of-week, traffic jam hours*, and so on.

On the other hand, creating hierarchies over spatial data is more complicated than doing so over temporal data. In fact, non-explicitly defined hierarchies might exist over the spatial data. For example, in the hierarchy *road < cell < district < city < state < country*, it is not always true that an inclusion relation holds between *district* and *cell* and between *cell* and *road*. A *road*, for example, might cross more than one *cell*. To solve the problem we could use the conceptual model proposed by Jensen et al [JKP+04], which supports dimensions with full or partial containment relationships (recall Figure 8). Thus, when a partial containment relationship exists between the different levels of a dimension, we should specify the degree of containment (e.g. 80% of a specific *road* is covered by a specific *cell*).

Finally, for the thematic (user profile) dimensions, different hierarchies apply for different attributes.

Collecting the features presented above in a single framework, we can depict an example of a three-dimensional snowflake schema for a trajectory data cube, as illustrated in Figure 11.

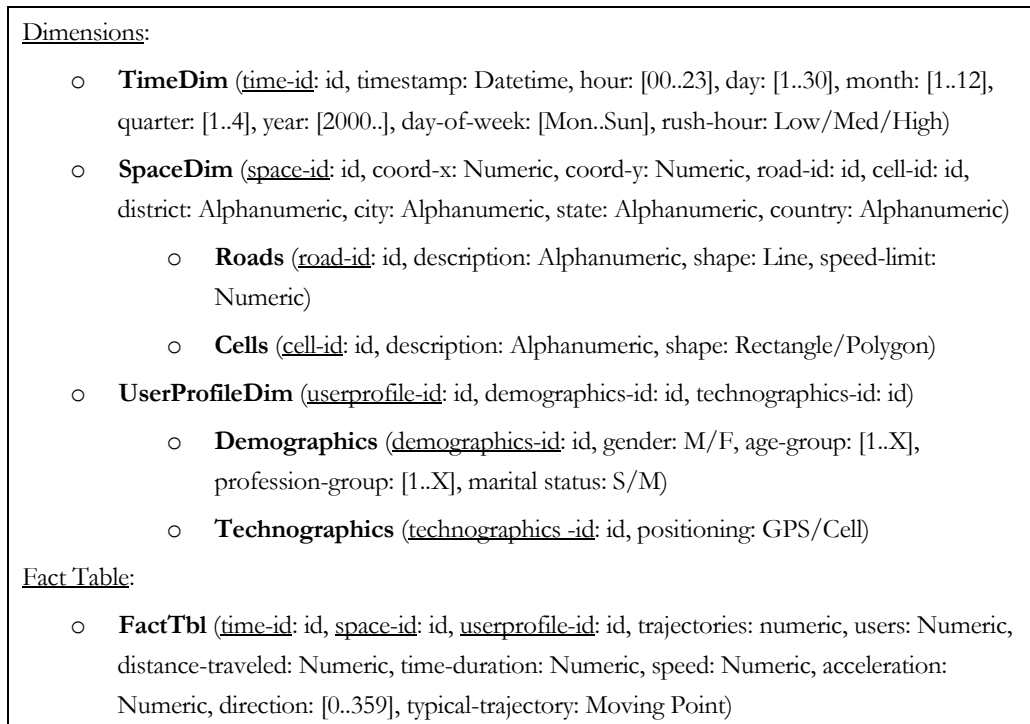


Figure 11: The proposed GeoPKDD trajectory data cube schema

4.2 OLAP issues

In traditional DWs, data analysis is performed interactively by applying a set of OLAP operators. In spatial data warehousing, particular OLAP operators have been defined to tackle the specificities of the domain [PTK+02]. Similarly, in our context, we expect an algebra of OLAP operators to be defined for trajectory data analysis. Such an algebra should include not only the traditional operators, such as roll-up, drill-down and selection properly tailored to trajectories, but also additional operators which account of the specificity of the spatio-temporal data type. In particular:

Roll-up. The roll-up operation allows us to navigate from a detailed level of abstraction to a more general level either by climbing up the concept hierarchy (e.g. from the level of “city” to the level of “country”) or by some dimension reduction (e.g. by ignoring the ‘time’ dimension and performing aggregation only over the ‘space’ dimension).

As shown in [BOO+06], depending on the kind of analysed measures, the roll-up operation in TDWs can introduce some errors (e.g. the number of distinct trajectories in a spatio-temporal cell). Assuming the object or trajectory identifier is not recorded, when summing up along the spatial and/or temporal dimension, one cannot obtain the distinct number of trajectories because there is only aggregated information. This is a particular case of the already discussed distinct counting problem.

Another open issue concerns the application of the roll-up operation when uncertain data exist; this is the case for the trajectories. Indeed, two factors of uncertainty should

be taken into account during the aggregation: the uncertainty in the values and the uncertainty in the relationships. The former refers to the uncertainty associated with the values of the dimensions and measures, which is propagated into the warehouse from the source data. The latter refers to the uncertainty imposed into the warehouse due to the non-explicitly defined concept hierarchies.

Drill-down. The drill-down operation is the reverse of the roll-up operation. It allows us to navigate from less detailed to more detailed data by either stepping down a concept hierarchy for a dimension (e.g. from the *country* to the *city* level) or by introducing additional dimensions (e.g. by considering not only the ‘space’ dimension but also the ‘time’ dimension). Similarly to the roll-up operation, drill-down is also “sensitive” to the distinct counting problem and to the uncertainty associated with both values and relationships.

Slice and Dice. The slice operation performs a selection over one dimension (e.g. *city* = “Athens”), whereas the dice operation involves selections over two or more dimensions (e.g. *city* = “Athens” and *year* = 2006). The conditions may involve not only numeric values but also more complex conditions, like spatial or temporal windows.

In summary, traditional OLAP operations should be also supported by a TDW since they provide meaningful information. An open issue is how other trajectory-dedicated operations could be supported. Examples include:

- **Fold / unfold** operators that dynamically modify the spatio-temporal granularity of measures representing trajectories;
- **Medoid** etc. operators which apply advanced aggregation methods, such as clustering of trajectories to extract representatives from a set of trajectories;
- Operators to propagate/aggregate **uncertainty and imprecision** present in the data of the TDW.

4.3 Feeding and operating the TDW

The previous sections disclosed high level requirements for TDW as these can be captured by extended conceptual and logical DW models. In this section we investigate the management requirements of a TDW from an implementation point of view, but still without restricting the discussion under a specific physical modeling framework.

4.3.1 ETL process

Having as main objective to build a DW specialized for trajectories and considering the complexity and the vast volumes of trajectory data, we need to differentiate our architectural design from the one in traditional DWs. The situation gets even more complicated by the streaming nature of data sources such as logs from location-aware communication devices, which potentially come in continuous flows of unbounded size. Therefore, efficient and effective storage of the trajectories into the TDW should be devised, capable of dealing with continuous incoming streams of raw log data, while the TDW itself must be equipped with suitable access methods to facilitate analysis and mining tasks. This poses extra challenges to be solved as the ability of incrementally

processing the data stream in an efficient and accurate way, and the definition of adaptive strategies to make the hypercubes evolve with the data stream.

Also, due to the peculiarities of trajectories, some problems arise in the loading phase of the fact table. To give an intuitive idea of these issues, consider a TDW with the ‘space’ and ‘time’ dimensions only, discretized according to a regular grid, and a fact table with the number of distinct trajectories found in a cell as the single measure. Moreover, assume that a trajectory is modeled as a finite set of sampled points. For example, Figure 12(a) shows a sampling of a trajectory.

The main issues are the following:

- the rough observations in a sampling cannot be directly used to compute the measures of interest in a correct way, and
- these observations are not independent points; the fact that they belong to the same trajectory has to be exploited when computing some measures.

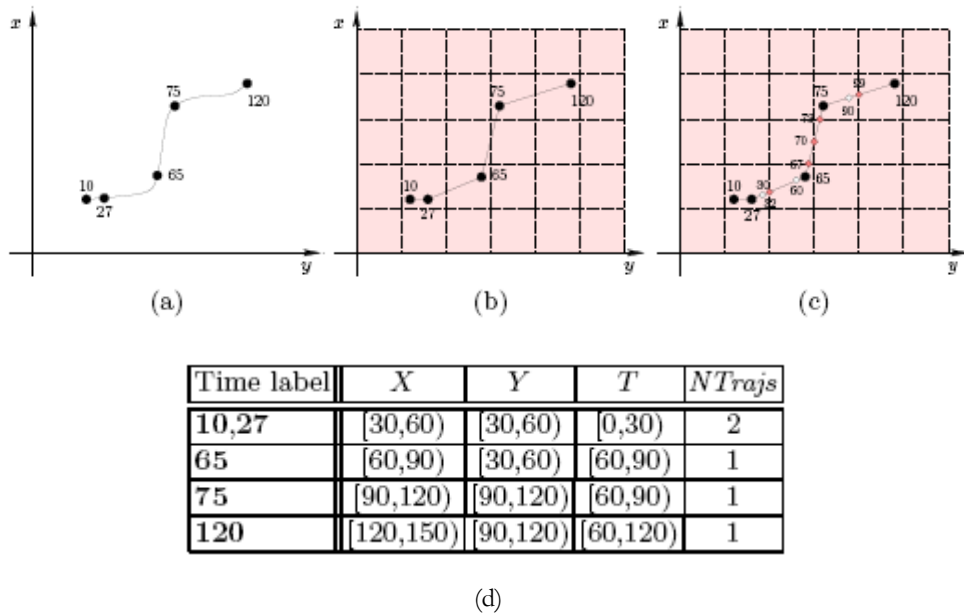


Figure 12: (a) the 2D sampled points of a trajectory, (b) linear interpolation of the trajectory, (c) the interpolated trajectory with the points matching the spatial and temporal minimum granularity, (d) the resulting fact table

It is evident that other cells might be crossed by the trajectory (e.g., the cell [60,90) X [60,90) X [60,90)), meaning that some information can be missing. On the other hand, the same cell can contain more than one observation, the computed measure is not correct because it does not store the number of distinct trajectories (see the cell [30,60) X [30,60) X [0,30)).

In order to solve the first problem, [BOO+06] proposes further intermediate points to be added by linearly interpolating the trajectory. The newly inserted points are the ones which intersect the borders of the cell, considering all its three dimensions.

Figure 12(c) shows the resulting interpolated points as white and gray circles. Note that the white interpolated points, associated with temporal labels 30, 60, and 90, have been

added to match the granularity of the temporal dimension. In fact, they correspond to cross points of the temporal border of 3D cell. On the other hand, the gray points, labeled with 32, 67, 70, 73, and 99, have been instead introduced to match the spatial dimensions. They correspond to the cross points of the spatial borders of some 3D cell, or, equivalently, the cross points of the spatial 2D squares depicted in Figure 12(c).

The second problem concerning duplicates is more complex and an approach to cope with it is presented in Section 3.2. A thorough discussion about errors in the computation of different measures related to the described issues can be found in [BOO+06].

4.3.2 OLAP Servers

DW systems require specialized servers that can support OLAP processing. There are OLAP Servers available for nearly all the major database systems. Among the most popular OLAP Servers are the following ones:

- **Oracle OLAP** [ORA05], provided by Oracle, is fully integrated into the relational database, i.e. all data and metadata is stored and managed from within Oracle Database. The data cube is queried through SQL.
- **Microsoft SQL Server Analysis Services 2005** [MSS06], provided by Microsoft, is a component of the Microsoft SQL Server, but supports feeding also from different data sources. The data cube is queried through MDX.
- **Pentaho Mondrian** [PAS06], is an open-source OLAP server written in Java. It executes queries in MDX language reading data from a relational database (RDBMS) and presents the results in a multidimensional format via a Java API. The data cube is queried through MDX.

Concluding this short section, we acknowledge that several OLAP servers with adequate features exist out there. A key issue in the final choice for GeoPKDD purposes is the availability of the source code; this sounds a critical factor since new OLAP operations specialized for trajectories might need to be developed. As such, we are in favor of the Mondrian open-source OLAP server.

5 Further issues

So far, we have addressed the modeling and operational requirements for building TDWs. In this section we present some further open issues that should be tackled, focusing on support for multiple spatial topologies, multiple trajectory representation issues, trajectory compression, uncertainty handling, and privacy and security aspects.

5.1 Multiple spatial topologies

Certainly, a factor that characterizes a TDW is the interrelationship between the development of the trajectories upon various possible spatial topologies represented by corresponding spatial dimensions. The base level partitioning of a spatial topology directly affects the multidimensional analysis of trajectories. Possible available

topologies may be simple grids (e.g. artificial partitioning), complex polygonal amalgamations (e.g. suburbs of a city), real road networks or mobile cell networks. The first case is the simplest one as the space is divided in explicitly defined areas of a grid and thus it is easy to allocate trajectory points in specific areas. However, counting the number of objects that passed from an area may be proved hard for a TDW. This is because sampling frequency may not help in representing the actual trajectory [BOO+06]. Thus, it may be necessary to reconstruct the trajectory (as an ETL task) to add intermediate points between the sampling data (see an illustration of this approach in Figure 12(c)). The same problem stands for the general case of arbitrary polygons. In case of road networks, trajectories should be reconstructed so as to be network constrained, whereas managing cells is a more complex problem because the areas covered by cells may change from time to time depending on the signal strength of the base stations of the provider. Whatever the base of the spatial dimension relating with the trajectories all spatial topologies are subject to the previously mentioned distinct counting problem [TKC+04]. Obviously, the reconstruction of the trajectories and the multiple counts of an object moving inside a region is straightforwardly dependent on the interpolation (e.g. linear, polynomial) used (if any) by the corresponding trajectory data model. The above discussion implies that an analyst has the ability firstly to analyze a bunch of trajectories according to a population thematic map and at a secondary level according to the road network of the most populated area.

5.2 *Multiple representation of trajectories*

While multiple representation has received a lot of attention in the spatio-temporal database community [HE02], multiple representation in trajectory modeling and particularly TDW is an open issue, as only few proposals exist tackling only spatial data [DS06]. Multiple representation means that we want to store and/or be able to retrieve several representations for the same trajectory. This may result from the description of the same trajectory according to different viewpoints but also, and more importantly, according to different spatial and temporal granularities (Figure 13).

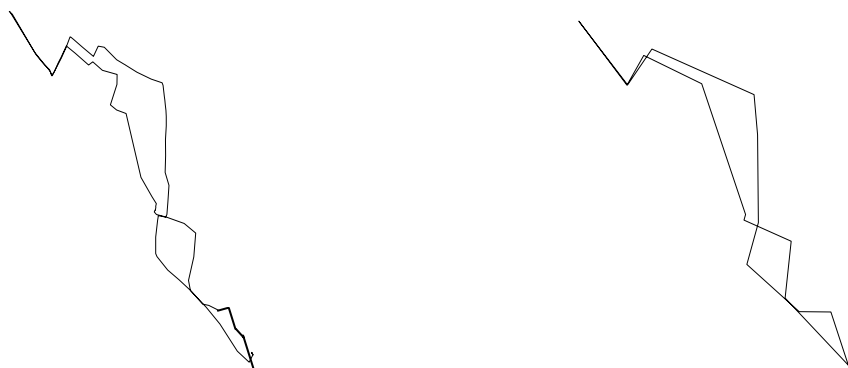


Figure 13: The route of a trajectory in different spatial and temporal granularities

Granularity, here, refers to the notion that the world is perceived at different level of details i.e. in the temporal aspect using more or less time steps and in the spatial aspect considering more or less location details. For instance, consider the trajectory of a person traveling from home in Lausanne to work in Geneva, some tasks might be only

interested in analysing the trajectory from the starting point in Lausanne to the arrival point in Geneva and then use a coarse spatial and temporal granularity. On the contrary, another task might need a more detailed description: at 7.40 am the person leaves home to walk to the bus station, then takes the bus to the train station where she/he waits for 5 minutes, then she/he travels for 30 minutes and so on. In this example, the same trajectory is described at different levels of spatial and temporal granularity. The DW has to provide for concepts to describe both of them as two representations of the same trajectory well as their corresponding levels of granularity. Another case is when the same trajectory has only one representation but that includes different parts at different granularities: for instance, a detailed description of the trajectory between the person's home and the train station will be kept while from Lausanne train station to Geneva train station no specific detail is necessary. Here, the DW has to be aware of the different granularities. The DW has to provide for multiple representation concepts and conversion operations to shift between the multiple granularities.

5.3 Trajectory compression

As addressed by [MB04], it is expected that all the ubiquitous positioning devices will eventually start to generate an unprecedented data stream of time-stamped positions. Sooner or later, such enormous volumes of data will lead to storage, transmission, computation, and display challenges. Hence the need for compression techniques arises. However, existing work in this domain is relatively limited [CWT03, MB03, MB04, PPS06a, PPS06b], and mainly guided by advances in the field of line simplification, cartographic generalization and data series compression. In particular, Meratnia and By [MB04] exploit existing algorithms used in the line generalization field, presenting one top-down (producing thus the TD-TR algorithm) and one opening window algorithm (producing the OW-TR algorithm), which can be directly applied to spatio-temporal trajectories.

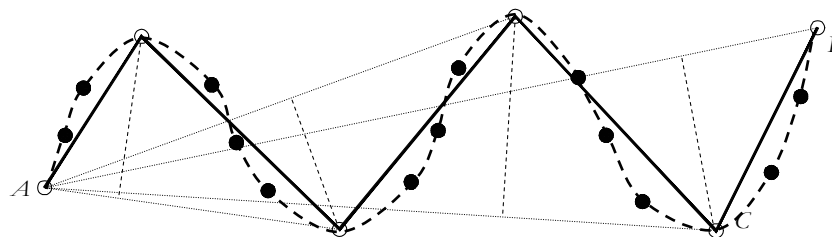


Figure 14: Top-down Douglas-Peucker algorithm used for trajectory Compression to represent different trajectory granularities. Original data points are represented by closed circles [MB04]

Regarding the context of GeoPKDD project, we plan to incorporate both types of algorithms in the trajectory stream manager, since each one serves different requirements; The top-down algorithm, named TD-TR, is based on the well known Douglas-Peucker [DP73] algorithm (Figure 14) introduced by geographers in cartography, and is used in an off-line mode in order to produce trajectories in different granularities. This algorithm calculates the perpendicular distance of each internal point from the line connecting the first and the last point of the polyline (line AB in Figure 14) and finds the point with the greatest perpendicular distance (point C). Then it creates lines AC and CB and, recursively, checks these new lines against the remaining points

with the same method, and so on. When the distance of all remaining points from the currently examined line is less than a given threshold (e.g., all the points following C against line BC in Figure 14) the algorithm stops and returns this line segment as part of the new - compressed - polyline.

Being aware of the fact that trajectories are polylines evolving in time, the algorithm presented in [MB04] replaces the perpendicular distance used in the DP algorithm with the so-called Synchronous Euclidean Distance (SED), also discussed in [CWT03, PPS06a], which is the distance between the currently examined point (P_i in Figure 15) and the point of the line (P_s, P_e) where the moving object would lie, supposed it was moving on this line, at time instance t_i determined by the point under examination (P_i in Figure 15). The time complexity of such an algorithm is $O(N \log N)$.

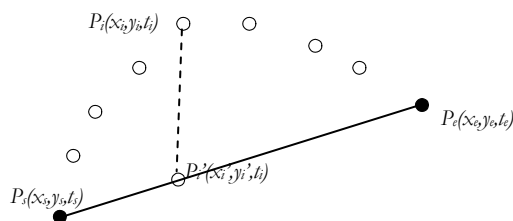


Figure 15: The Synchronous Euclidean Distance (SED): The distance is calculated between the point under examination (P_i) and the point P_i' which is determined as the point on the line (P_s, P_e) the time instance t_i [MB04]

On the other hand, the opening window algorithm (OW-TR) stands for online compression during the streaming process; as such, trajectories feeding the trajectory database can be compressed considering the appropriate compression parameters (i.e. the algorithm threshold). The algorithm starts by anchoring the first trajectory point, and attempt to approximate the subsequent data points with one gradually longer segment (Figure 16). As long as all distances of the subsequent data points from the segment are below the distance threshold, an attempt is made to move the segment's end point one position up in the data series. When the threshold is going to exceed, two strategies can be applied: either the point causing the violation (Normal Opening Window, $NOPW$) or the point just before it (Before Opening Window, $BOPW$) becomes the end point of the current segment, and also the anchor of the next segment. If the threshold is not exceeded, the float is moved one position up in the data series (i.e., the window opens further) and the algorithm carries on until it finds the trajectory's latest point; then the whole trajectory is transformed into a linear approximation.

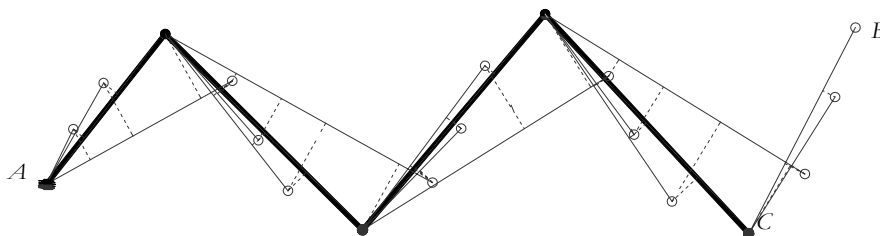


Figure 16: Opening Window algorithm used for online trajectory Compression. Original data points are represented by closed circles

While in the original OW class of algorithms – used in the line generalization field - each distance is calculated from the point perpendicularly to the segment under

examination, in the OPW-TR algorithm presented in [MB04], the *SED* distance is evaluated. Although OW algorithms are computationally expensive - since their time complexity is $O(N^2)$ - they are very popular. This is because, they are online algorithms, and work reasonably well in presence of noise (but only for relatively short data series). Moreover, the time complexity is $O(N^2)$ regarding only the compression of the full data series; when dealing with each point update - that is in the on-line case - the complexity of determining whether each incoming point will be float or the next anchor, is $O(N)$.

5.4 Uncertainty issues

As already discussed in previous sections, the recorded location of a moving object is rather imprecise due to several factors like GPS erroneous measurements and sampling errors. Since the DW is built upon these data, it is obvious that information lying in the DW is also subject to the uncertainty factor.

An interesting direction in the management of uncertainty in TDWs is to determine the way how the location uncertainty introduced in moving objects from the measurement and sampling errors, propagates to the aggregate information stored in the DW. Consider, for example, Figure 17 that illustrates a set of trajectories sliced across the temporal dimension so as to produce a set of points and a rectangular space partition used for aggregation. Adopting the uncertainty model introduced by Trajcevski et al. [TWZ+02], this set of objects can be represented as a set of *points* along with the respective *uncertainty circles* inside which the actual point would be found. Then, considering the aggregation over the number of objects contained in each bucket (i.e. a cell of the data cube), along with this number of objects, the aggregation would also contain information about the percentage of objects reported inside, nevertheless being outside each bucket (and vice versa), applying a form of probabilistic query as the one discussed in [TWH+04]. For example, while bucket B_1 would be reported to contain 3 points, there is also the possibility to contain 5 points, since the uncertainty circles of p_1 and p_2 overlap B_1 - thus p_1 and p_2 might be possibly contained inside it.

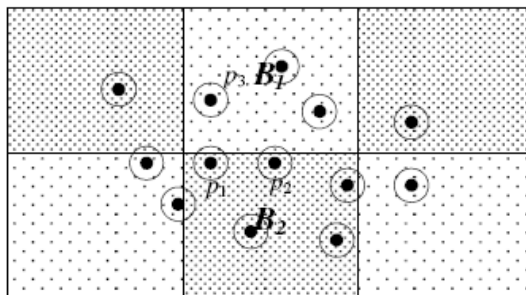


Figure 17: Setting of the uncertainty problem

Another preliminary conclusion gathered from Figure 17 is that objects being spatially far from the bucket's boundary cannot contribute to the uncertainty introduced in the aggregation over the space partitioning. Consider for example point p_3 whose uncertainty circle does not intersect the bucket's boundary; then p_3 will for sure contribute in the number of objects located in B_1 only. This observation leads to the conclusion that only points intersecting the *boundary* of the space partitioning may contribute to the overall aggregated uncertainty. As such, given that the actual data

uncertainty can not be reduced (since it is device dependant), the uncertainty introduced in the aggregation can be minimized by minimizing the *length* of the partitioning boundary. Concluding, the propagation of data uncertainty can be minimized by adopting a partitioning having minimum length for a given area that has to be covered; the above requirement is met in the *square* partitioning.

Another notion of uncertainty, namely the uncertainty in the relationships of spatial objects, has been addressed by Jensen et al. [JKP+04]. As discussed in previous sections, when a partial containment relationship exists between different levels of a dimension, the degree of containment should be specified, e.g. 50% of the roadway is contained in the district. However, incorporating the notion of weighting of the relationships in the data cube is not so straightforward, mainly due to the presence of the measures. Consider, for example, a scenario where the average speed on a road is 50km/h and this road is partially contained, with a weight of 10%, in a district. Then, consider another road with average speed of 120km/h, partially contained, with a weight of 90%, in the same district. In order to find the average speed within the district, we have to take into account both roads. The simplest solution is to adopt a weighted technique; however other approaches like fuzzy techniques could be also investigated. This procedure, though, might increase the uncertainty in the answers, thus the quality of the answers should be ensured by means of some predefined lower and upper bounds. Furthermore, the end user should be aware of the uncertainty accompanying its data at each level of aggregation. Also, capabilities that would allow the end user to query only data that fulfill some permissible uncertainty limit are useful.

5.5 Security and privacy in a mobile context

For GeoPKDD purposes, our research also covers information security and privacy issues in mobile applications. We are specifically concerned with security and privacy requirements which raise in mobile communities such as enterprises operating on field, healthcare organizations and military and civilian coalitions, in which individuals, because of their role in the community, may need to access common information resources through location-aware applications. As an example, consider a mobile application for the personnel and patients of a health care organization. Suppose that individuals are given a location-aware terminal with which they can request information services provided by an application server. The organization consists of individuals who have different functional roles, e.g. Nurse, Doctor and Patient. Depending of the organizational context, the information available to users may differ based on the functional roles of users. For example nurses, unlike doctors, may not be authorized to modify records of patients. Furthermore, the accessibility of information may also depend on the position of the requester. For example, a doctor may be allowed, for privacy preserving purposes, to access the record of a patient only when located in the department she has been assigned to. Another related concern is due to the fact that location-aware applications, and, in particular, LBS and ubiquitous computing, are capable of collecting large amounts of personal location data, which can then be stored, linked with external sources and released to third parties without users' consent. Location data enables intrusive inferences, which may reveal habits, social customs, religious and sexual preferences of individuals, and can be used for unauthorized advertisement and user profiling. Therefore, to protect sensitive information, a strong

control over the access and the eventual disclosure of location information is needed. To address these requirements we have investigated a conceptual framework integrating location-based access control functionalities and location privacy preserving capabilities. Such a framework is based on the GEO-RBAC access control model and the k-anonymity model for spatiotemporal data. The GEO-RBAC model adds spatial capabilities to standard RBAC (Role Based Access Control). Moreover, the model provides a basic mechanism for representing the position of users at varying spatial granularities, that can be used for location privacy preserving purposes. By the same token, an extension of the k-anonymity model which was originally applied to relational data is investigated in order to ensure location privacy for the users.

In this section we overview the main features of the GEO-RBAC model and the k-anonymity model. Note that the design of an architecture for integrating access control and privacy preserving capabilities in TDW, along with the specification of a first prototype, are parts of our next work in the project.

5.5.1 *The GEO-RBAC model: an overview*

Like the RBAC standard [SCFY06], GEO-RBAC consists of three distinct layers which provide a set of increasingly complex functionalities for controlling the access to information resources. The three levels are referred to as Core, Hierarchical and Constrained GEORBAC respectively. Of these, the Core GEO-RBAC represents the minimal and necessary component of any GEO-RBAC access control system, and is grounded on the notion of spatial role. Hierarchical GEO-RBAC supports the specification of spatial role hierarchies, whereas Constrained GEO-RBAC supports the specification of Separation of Duty Constraints between spatial roles. We now overview the main concepts underlying the Core GEO-RBAC. For further details on the model, we refer the reader to [BCDP05]. Core GEO-RBAC is built upon three main concepts: *spatial role*, *position model* and *role schema*.

- ***Spatial Role.*** A spatial role denotes a spatially-bounded organizational role. A spatial role (simply role hereinafter) has besides a *role name*, a *role extent* which defines the boundaries of the spatial region, in which users are enabled to play the given role and thus request the associated permissions. For example, *Nurse(PaediatricDept)* is a role: *Nurse* is the role name and *PaediatricDept* the role extent, that is, the identifier of a spatial object describing a region in the reference space. When the user logs in, a new session is created and a number of roles are activated. Nevertheless, for a role to become effective, i.e. *enabled*, a user must be logically located within the space of the corresponding role extent.
- ***The position model.*** The position occupied by the user is described at two different levels, called *real* and *logical position*, respectively. The real position corresponds to the position of the user on Earth acquired through some positioning technology, whereas the logical position not only has a geometric shape, but also a semantics. For example, logical positions are: a house, an address number, a road. The logical position is obtained from a real position by applying an application-dependent function called *location mapping function* which is specific for each role schema (see below). The advantage is that the logical position is, at least to some extent, independent of the positioning

technology. Moreover it provides a basic mechanism for representing location data at different levels of granularities and thus “hide” the actual position of the user. This feature is important for location privacy preserving purposes.

- **Role Schema.** A novelty of the model is the distinction between role schema and role instance which account respectively for the intensional and extensional dimension of roles. Notice that in the classical approaches roles are simply denoted by names and therefore roles do not have an intensional/extensional characterization. In our model, a role instance is a spatial role defined over a specific extent, in compliance with the role schema. A role schema defines some common properties of roles with a similar meaning. For example $Nurse(Dept, Room, mRoom)$ is the schema for the *nurse* roles. In this schema, *Nurse* is the common name of a set of roles, whereas *Dept* denotes the type of the role extent; *Room* denotes the type of logical position and *mRoom* denotes the location mapping function which computes the room in which the nurse is located based on the real position. Another property of the model is that permissions are assigned to both role schemas and role instances. The permissions assigned to a schema are then inherited and shared by all the instances of that schema. Ultimately, the notion of role schema serves to abstract common properties out of a set of homogeneous roles and thus simplify the specification of roles and ultimately role engineering.

A sample fragment of policy describing role schemas and role instances is reported in Figure 18. As a running example, we consider roles in a healthcare organization. We use the following notation: $REXT_FT$ denotes the set of role extent types; $REXT$ denotes the set of role extents; $LPOS_FT$ denotes the set of logical position types. R , RS , and RI , denote respectively the set of role names, role schemas and role instances.

$$\begin{aligned}
 R &= \{Doctor, Nurse, Patient\} \\
 REXT_FT &= \{Hospital, Dept\} \\
 REXT &= \{Hosp\#1, Dept\#1\} \\
 LPOS_FT &= \{Room, Sector\} \\
 R_S &= \{Doctor(Hospital, Sector, m_Sector), \\
 &Nurse(Dept, Room, m_Room), \\
 &Patient(Hospital, Sector, m_Sector)\} \\
 R_I &= \{Doctor(Hosp\#1), \\
 &Nurse(Dept\#1), \\
 &Patient(Hosp\#1)\}
 \end{aligned}$$

Figure 18: A sample fragment of policy

5.5.2 The *K*-Anonymity Model for Spatiotemporal Data

Several techniques have been proposed over the years with the goal of protecting user’s privacy. Most of these techniques vary based on the type of the database in which they are applied as well as on the impact of the technique in the original values. Two of the early approaches which were used for this purpose were the introduction of noise to the original tabular data and the swapping of original values between rows and columns so that the actual values in the database to be difficult to be retrieved [EB99, Kim86]. Although these techniques were doing well in ensuring the privacy of the data, they were producing data of low usability.

Recent advances in data security and privacy research have created a wealth of techniques to come to the rescue of people's privacy. These techniques are known as anonymization techniques and they rely on the concept of k-anonymity [SS98, Swe02, DT05, BA05]. A database is said to be k-anonymous if every sequence of values in the quasi-identifier appears in the database at least k occurrences. A quasi-identifier (as opposed to an identifier) is a set of attributes from a relation schema which if linked with external data may be used to uniquely identify individuals. Generalization and suppression of data are two of the most popular techniques developed to extract anonymous data from databases.

The inception of new applications generating enormous amount of spatial and temporal user data (location based services, navigation systems, and others) has created a fundamental need of being able to ensure the privacy of the subjects (users, clients, etc.) involved in these collections of data. An extension of the basic k-anonymity model has been proposed to offer privacy to the individuals in a LBS scenario [BWJ05, GH05, GL05, GG03]. In such a scenario, it is not only the personal static information (names, addresses, preferences, etc) which is recorded, but also the location of a user within a time interval, when this user requested some service. This additional piece of dynamic information can under certain circumstances be used by an adversary to track down a specific individual. In our work, we are investigating the applicability of the k-anonymity model for preserving the privacy in spatiotemporal data, by considering the degree of privacy which can be offered through different spatiotemporal k-anonymous algorithms.

More specifically, in our proposed solution methodology, we are making use of frequent spatiotemporal patterns of users which we argue that can serve as location based quasi-identifiers. In other words, if an adversary knows the frequent patterns of movements of some user, it is quite easy for the adversary to link a user pattern with a specific user, even if the identity of the user is not known explicitly. After these frequent patterns have been found, a k-anonymous spatiotemporal system is responsible for ensuring that whenever a user exhibits a behavior which partially matches with a frequent movement pattern, the system needs to take action, so that the user cannot be identified from his behavior. We are currently applying two techniques for anonymizing spatiotemporal data. The first technique is a scheme which has already been applied to simple relational data, and is known as generalization [BWJ05]. The generalization technique ensures that whenever the observed behavior of a user matches with one or more of his location based quasi-identifiers, the location and time of a user's request is expanded (generalized) in such a way that the expanded area contains k-1 other users who may have also sent a similar request with the user subject. In this way, the user can escape the linking of his request with his location-based quasi-identifier.

Yet another precaution that is taken by a k-anonymous location system is to ensure that the privacy of an individual is not jeopardized, even if the generalization technique fails. A failure of the generalization scheme can happen if the maximum possible generalization adhering to some user and/or application defined constraints did not succeed in locating k-1 other users in the nearby area where the user requested the service. In this situation, the system makes use of certain spatial areas, where a user is dissociated from his/her previous system identity, and he/she is given a new identity, so that it will be difficult for the data terrorist to link the user who enters such an area with a user who leaves from this area.

An initial experimentation study with a k-anonymous location system that is based on a generalization and an unlinking technique and an in-house data generator, has indicated that the spatiotemporal data anonymization achieved through such a system exhibits very promising behavior. Further experimental studies are underway in order to fully evaluate the implemented techniques, so that we will be able to fine tune the k-anonymous system parameters. We are also considering the implementation of other similar techniques to achieve better results as well as the implementation of a prototype system that will serve as a testbed for both demonstrating the functionality and facilitating the experimentation with synthetic and real data sets.

6 Conclusions

In this report, we have discussed trajectory warehousing as the means to transform raw space-time dependent data in the form of data cubes to valuable information that can be used in decision making processes. The starting point in our study was the consideration of the literature in spatial data warehousing, as it is the area that presents the highest commonalities with trajectory warehousing. This study has highlighted a set of modeling and management requirements emanating from the specificities of trajectories that should be fulfilled in the development of efficient and semantically rich TDWs. Addressing these requirements forms a roadmap towards the development of an effective TDW for GeoPKDD purposes.

7 References

- [AAD+96] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. *Proc. VLDB'96*, pages 506-521.
- [BA05] R. Bayardo and R. Agrawal. Data Privacy Through Optimal K-Anonymity. *Proc. ICDE'05*, pp. 217-228.
- [BCDP05] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. GEO-RBAC: A Spatially Aware RBAC. *Proc. SACMAT'05*, pp. 29-37.
- [BMH01] Y. Bédard, T. Merrett, and J. Han. Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery. Chapter 3 in *Geographic Data Mining and Knowledge Discovery*, Taylor & Francis, Vol. Research Monographs in GIS, pp. 53-73, 2001.
- [BOO+06] F. Braz, S. Orlando, R. Orsini, A. Raffaeta, A. Roncato, and C. Silvestri. Towards a Trajectory Data Warehouse. Technical Report CS-2006-4, Università Ca' Foscari di Venezia, 2006.
- [BPS+05] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. *Proc. VLDB'05*.
- [BTM05] S. Bimonte, A. Tchounikine, and M. Miquel. Towards a spatial multi-dimensional model. *Proc. DOLAP'05*.

- [BWJ05] C. Bettini, X. S. Wang, and S. Jajodia. Protecting Privacy Against Location-Based Personal Identification. *Proc. SDM'05*.
- [CWT03] H. Cao, O. Wolfson, and G. Trajcevski. Spatio-temporal Data Reduction with Deterministic Error Bounds. *Proc. DIALM-POMC'03*.
- [DG00] S. Dieker and R. H. Güting. Plug and Play with Query Algebras: Secondo. A Generic DBMS Development Environment. *Proc. IDEAS'00*, pp. 380-390.
- [DP73] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10:112-122, 1973.
- [DS06] M. L. Damiani and S. Spaccapietra. Spatial Data Warehouse Modelling. Chapter in *Processing and Managing Complex Data for Decision Support*, Idea Group Publishing, pp. 21-27, 2006.
- [DT05] J. Domingo-Ferrer and V. Torra. Ordinal, Continuous and Heterogeneous K-Anonymity Through Microaggregation. *Data Mining and Knowledge Discovery*, 11(2): 195-212, 2005.
- [EB99] V. Estivill-Castro and L. Brankovic. Data Swapping: Balancing Privacy against Precision in Mining for Logic Rules. *Proc. DaWaK'99*, pp. 389-398.
- [EGS+99] M. Erwig, R.H. Güting, M. Schneider, and M. Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica*, 3(3):265-291, 1999.
- [FGN+00] L. Forlizzi, R. H. Güting, E. Nardelli, M. Schneider. A Data Model and Data Structures for Moving Objects Databases. *Proc. ACM SIGMOD'00*, pp. 319-330.
- [GBE+00] R.H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems*, 25(1):1-42, 2000.
- [GBL+96] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. *Proc. ICDE'96*, pp. 152-159.
- [GG03] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. *Proc. MobiSys'03*, pp. 31-42.
- [GH05] M. Gruteser, B. Hoh. On the Anonymity of Periodic Location Samples. *Proc. ICSPC'05*, pp. 179-192.
- [GL05] B. Gedik, L. Liu. A Customizable K-Anonymity Model for Protecting Location Privacy. *Proc. ICDCS'05*.
- [GMP+05] F. Giannotti, A. Mazzoni, S. Puntoni, and C. Renso. Synthetic Generation of Cellular Network Positioning Data. *Proc. ACM-GIS'05*.
- [HE02] K. Hornsby and M. J. Egenhofer. Modeling Moving Objects over Multiple Granularities. *Annals of Mathematics and Artificial Intelligence*, 36(1-2):177-194, 2002.

- [HSK98] J. Han, N. Stefanovic, and K. Kopersky. Selective Materialization: An Efficient Method for Spatial Data Cube Construction. *Proc. PAKDD'98*, pp.144-158.
- [Inm96] W. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 2nd Edition, 1996.
- [JKP+04] C.S. Jensen, A. Kligys, T.B. Pedersen, C.E. Dyreson, and I. Timko. Multidimensional data modeling for location-based services. *The VLDB Journal* 13:1–21, 2004.
- [Kim86] J. Kim. A Method for Limiting Disclosure in Microdata Based on Random Noise and Transformation. *Proc. Section on Survey Research Methods of the American Statistical Association*, pp. 303-308, 1986.
- [LFG+03] J. A. C. Lema, L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. Algorithms for Moving Objects Databases. *The Computer Journal*, 46(6):680-712, 2003.
- [LS05] I. Lopez and R. Snodgrass. Spatiotemporal Aggregate Computation: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):271-286, 2005.
- [MB03] Meratnia, N., By, R., “A new perspective on trajectory compression techniques”, *Proc. ISPRS DMGIS*, 2003.
- [MB04] N. Meratnia and R. de By. Spatiotemporal Compression Techniques for Moving Point Objects. *Proc. EDBT'04*.
- [MSS06] Microsoft SQL Server: Analysis Services Overview. Available at www.microsoft.com.
- [MZ04] E. Malinowski and E. Zimanyi. Representing Spatiality in a Conceptual Multidimensional Model. *Proc. ACM GIS'04*, pp. 12-21.
- [MZ06] E. Malinowski and E. Zimanyi. Hierarchies in a multidimensional model: from conceptual modeling to logical representation. *Data and Knowledge Engineering*, 2006.
- [ORA05] Oracle, Oracle OLAP, Application Developer's Guide 10g Release 2. August 2005. Available at www.oracle.com.
- [PAS06] Pentaho Analysis Services: Mondrian Project, Available at <http://mondrian.pentaho.org>.
- [PPS06a] M. Potamias, K. Patroumpas, and T. Sellis. “Sampling Trajectory Streams with Spatiotemporal Criteria. *Proc. SSDBM'06*.
- [PPS06b] M. Potamias, K. Patroumpas, and T. Sellis. Amnesic online synopses for moving objects. *Proc. CIKM'06*.
- [PT01] T. Pedersen and N. Tryfona. Pre-aggregation in Spatial Data Warehouses. *Proc. SSTD'01*, pp.460-480.
- [PT03] D. Pfoser and Y. Theodoridis, “Generating semantics-based trajectories of moving objects”, *Intl. J. of Computers, Environment and Urban Systems*, 27(3): 243-263, 2003.
- [PT06] N. Pelekis and Y. Theodoridis. Boosting Location-Based Services with a Moving Object Database Engine. *Proc. MobiDE'06*.

- [PTK+02] D. Papadias, Y. Tao, P. Kalnis, and J. Zhang. Indexing Spatio-Temporal Data Warehouses. *Proc. ICDE'02*.
- [PTV+06] N. Pelekis, Y. Theodoridis, S. Vosinakis, and T. Panayiotopoulos. Hermes - A Framework for Location-Based Data Management. *Proc. EDBT'06*, pp. 1130-1134.
- [PTZ+02] D. Papadias, Y. Tao, J. Zhang, N. Mamoulis, Q. Shen, and J. Sun. Indexing and Retrieval of Historical Aggregate Information about Moving Objects. *IEEE Data Engineering Bulletin*, 25(2):10-17, 2002.
- [RBM01] S. Rivest, Y. Bédard, and P. Marchand. Towards Better Support for Spatial Decision Making: Defining the Characteristics of Spatial On-Line Analytical processing (SOLAP). *Geomatica*, 55(4), 539-555.
- [RBP+05] S. Rivest, Y. Bédard, M.-J. Proulx, M. Nadeau, F. Hubert, and J. Pastor. SOLAP: Merging Business Intelligence with Geospatial Technology for Interactive Spatio-Temporal Exploration and Analysis of Data, *Journal of International Society for Photogrammetry and Remote Sensing (ISPRS)*, 60(1):17-33.
- [RG00] S. Rizzi and M. Golfarelli. Date warehouse design. *Proc. ICEIS'00*, pp. 39–42.
- [Riz03] S. Rizzi. Open problems in data warehousing: eight years later. *Proc. DMDW'03*.
- [RZY+03] F. Rao, L. Zhang, X. Yu, Y. Li, and Y. Chen. Spatial Hierarchy and OLAP-Favored Search in Spatial Data Warehouse. *Proc. DOLAP'03*, pp. 48-55.
- [SCFY96] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.
- [SHK00] N. Stefanovic, J. Han, and K. Koperski. Object-based selective materialization for efficient implementation of spatial data cubes. *IEEE Transactions on Knowledge and Data Engineering*, 12(6):938–958, 2000.
- [SLC+02] S. Shekhar, C. Lu, S. Chawla, and P. Zhang. Data Mining and Visualization of Twin-Cities Traffic Data. Technical Report, University of Minesota, 2002.
- [SS98] L. Sweeney, P. Samarati. Protecting Privacy when Disclosing Information: K-Anonymity and its Enforcement Through Generalization and Suppression. *Proc. IEEE Symposium on Research in Security and Privacy*, 1998.
- [Swe02] L. Sweeney. Achieving K-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5):571-588, 2002.
- [The03] Y. Theodoridis. Ten Benchmark Database Queries for Location-based Services. *The Computer Journal*, 46(6):713-725, 2003.
- [TKC+04] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-Temporal Aggregation Using Sketches. *Proc. ICDE'04*, pages 214–225.

- [TPG+01] J. Trujillo, M. Palomar, J. Gomez, and I. Song. Designing Data Warehouses with OO Conceptual Models. *IEEE Computer*, special issue on Data Warehouses, 34,12:66-75, 2001.
- [TWH+04] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. "Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems*, 29(3):463-507, 2004.
- [TWX+02] G. Trajcevski, O. Wolfson, B. Xu, and P. Nelson. Real-Time Traffic Updates in Moving Objects Databases. *Proc. MDDS'02*.
- [VS99] P. Vassiliadis and T. Sellis. A survey of logical models for OLAP databases. *SIGMOD Record*, 28(4):64-69, 1999.
- [ZT05] D. Zhang and V. Tsotras. Optimizing spatial Min/Max aggregations. *The VLDB Journal*, 14:170-181, 2005.