Project Acronym:
GeoPKDD

Project full title:
Geographic Privacy-aware Knowledge Discovery and Delivery

Project Number: FP6-014915
Instrument: Specific Target Research Project

Project Deliverable D3.7

Final report of WP3

Date of preparation: M7
Revised: M7
Revision: final

Operative commencement date of Contract: 01/11/2005
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: D3.7
Work-Package: WP3
Document Type: Document
Contractual Date of Delivery: M7
Actual Date of Delivery: M7
Title of Document: Technical Report
Title: Deliverable 3.7 – Final report of WP3

Author(s): Gennady Andrienko (FAIS), Chiara Renso (ISTI-CNR), Jose Macedo (EPFL), Monica Wachowicz (Univ. Polytechnic Madrid)

Dissemination Level: This document is public

GeoPKDD consortium participants

<table>
<thead>
<tr>
<th>Partic. Role</th>
<th>Partic. N.</th>
<th>Participant name</th>
<th>Short name</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>1</td>
<td>KDD Lab, joint research group of ISTI-CNR, Istituto di Scienza e Tecnologie dell’Informazione, Pisa. <a href="http://www-kdd.isti.cnr.it">http://www-kdd.isti.cnr.it</a> and Univ. Pisa, Dept. of Computer Science <a href="http://www.di.unipi.it">http://www.di.unipi.it</a></td>
<td>KDDLAB</td>
<td>I</td>
</tr>
<tr>
<td>CR</td>
<td>5</td>
<td>Wageningen UR, Centre for GeoInformation. <a href="http://cgi.gis.wageningenur.nl">http://cgi.gis.wageningenur.nl</a></td>
<td>WUR</td>
<td>NL</td>
</tr>
<tr>
<td>CR</td>
<td>6</td>
<td>Research Academic Computer Technology Institute, Research and Development Division. <a href="http://www.cti.gr">http://www.cti.gr</a> and Univ. Piraeus, Dept. of Informatics <a href="http://www.unipi.gr">http://www.unipi.gr</a></td>
<td>CTI</td>
<td>GR</td>
</tr>
<tr>
<td>CR</td>
<td>7</td>
<td>Sabanci University, Faculty of Engineering and Natural Sciences. <a href="http://www.sabanciuniv.edu">http://www.sabanciuniv.edu</a></td>
<td>UNISAB</td>
<td>TK</td>
</tr>
<tr>
<td>CR</td>
<td>8</td>
<td>WIND Telecomunicazioni SpA, Direzione Reti Wind Progetti Finanziati &amp; Technology Scouting. <a href="http://www.wind.it">http://www.wind.it</a></td>
<td>WIND</td>
<td>I</td>
</tr>
<tr>
<td>CR</td>
<td>9</td>
<td>Polytechnic University of Madrid, Latingeo Laboratory <a href="http://www.latingeo.net">http://www.latingeo.net</a></td>
<td>UPM</td>
<td>ES</td>
</tr>
</tbody>
</table>
Final report of WP3

Table of contents

1. A framework for progressive mining and querying of movement data
2. Reasoning task
3. Geographic visualization of movement patterns
A framework for progressive mining and querying of movement data

Riccardo Ortale, Giuseppe Manco
ICAR - CNR, Rende (CS), Italy.
Roberto Trasarti, Fosca Giannotti, Chiara Renso
ISTI-CNR, Pisa (PI), Italy

1 Introduction and State of the Art

Research on moving-object data analysis has been recently fostered by the widespread diffusion of new techniques and systems for monitoring, collecting and storing location-aware data, generated by a wealth of technological infrastructures, such as GPS positioning, sensor- and mobile-device networks, tracking facilities. These have made available massive repositories of spatio-temporal data, that call for suitable analytical methods, capable of enabling the development of innovative, location-aware applications.

So far, research efforts have been largely geared towards either the definition of new movement patterns, or the development of solutions to algorithmic issues, to improve existing pattern-mining schemes in terms of effectiveness and/or efficiency. As a consequence, several intelligent tools for movement data analysis have rapidly flourished [12]. In the meanwhile, however, the necessary attention has not been paid to the definition of a unifying framework, wherein to set the above pattern-mining tools as specific components of the knowledge discovery process. In the absence of a unifying framework, the process of progressively querying and mining both movement data and patterns is a challenging issue. Indeed, the individual mining techniques can hardly be combined into an actual multi-step process, since their results are typically neither directly exploitable to feed some further analysis, nor uniformly manageable with raw data. We believe that this is a primary limitation for the real-world applications of movement data analysis, where it rarely happens that a single pattern-mining activity (i.e. either of the foresaid tools) suffices to meet the underlying analytical requirements. In the current state of the art, the lack of support to knowledge discovery as an actual multi-step process makes impractical all those applications, that involve multiple stages of analysis and manipulation for both data and patterns, in which the results at each stage are required to become the input to the subsequent stage. As a motivating example, consider the following analytical requirement, that calls for the search of common behavioral patterns in the context of specific spatial patterns: among the movements of people, find the sequence of frequently visited regions on Sunday in the city centre and, among them, find the groups of trajectories that exhibit uniform
movement behavior. Satisfying such a requirement involves a complex two-steps process, wherein multiple mining methods as well as forms of background knowledge need be progressively and seamlessly exploited. The contribution of the introduced DAEDALUS framework is twofold. First of all, a formal model to support progressive mining and querying, in the specific context of geographic knowledge discovery, is described. Secondly, we present the effective implementation of the system, based on the powerful object-relational formalism and relying on emergent technology on Moving Object Database, thus providing for free a number of basic primitives to manage spatio-temporal data. In this approach, to the best of our knowledge, we take a first step towards progressively mining and querying movement data. In this direction, we propose an innovative computational environment, called DAEDALUS\(^1\), that provides effective support to the whole knowledge discovery process. DAEDALUS implements IAPYX\(^2\) data mining query language capable of supporting the user in specifying and refining mining objectives. The IAPYX language is based on an algebraic framework, called 2W Model, capable of accommodating and combining disparate mining tasks into a multi-step knowledge discovery process. DAEDALUS operates as a statement execution layer on top of the Hermes [16] moving-object database.

The approach we are proposing, a framework for supporting a data mining query language on movement data, spawns from the convergence of two different research areas: moving object databases and data mining query languages. Research on moving-object databases has addressed the need for representing movements of objects (i.e. trajectories) in databases, in order to perform ad-hoc querying and analysis. A number of papers address this issue, for example [22, 7]. Among them, we would like to mention here two moving object databases which has been proposed, namely SECONDO [1] and HERMES [16]. The first approach faces the study of abstract moving-object data types, as well as algorithms supporting the related operations, finally implemented in the SECONDO system. The Hermes framework, has been recently introduced by Pelekis et al. [16], that aims at aiding a database developer in modeling, constructing and querying a moving-object database. Hermes is the basic component of the proposed the DAEDALUS framework, introduced in section 3. The last decade has seen a proliferation of approaches to Data Mining Query Languages (DMQL). Some approaches provide an interface between data sources and data mining tasks, adopting an SQL-based style for creating and manipulating data mining models, thus abstracting away from the algorithmic particulars, see for example MINE RULE [13], DMQL [4], MSQL [8], the Mining Views approach [2] and IQL language [15]. Other approaches propose DMQL to support the design of specific procedural workflows which integrate reasoning on the mining results and possibly define ad-hoc evaluation strategies and activations of the Data Mining tasks. see for example Clementine [20] and Weka [21]. Other approaches are based on logics which is used as a unifying formalism for integrating input data sources with data mining tasks and discovered knowledge (\(\text{CL}_{\text{Mine}}\) [5] and \(\text{RDM}\) [11]). A recent survey on the above mentioned issues appeared in [12]. However, to the best of our knowledge, so far no framework

\(^1\)In Greek mythology, Daedalus was a most skillful artificer, or craftsman, so skillful that he was said to have invented images that seemed to move about (source Wikipedia).

\(^2\)Daedalus had two sons: Icarus and Iapyx.
for the specific context of movement data has been proposed.

2W Model was originally inspired from the approach proposed in [9] and subsequently refined by [3], namely, the 3W Model, which stands for *Three Worlds for data mining*: the D(ata)-world, the I(ntensional)-world, and the E xtensional-world. The proposed 2W Model introduces several meaningful differences that overcomes some of its limitations. Firstly, entities in the M-World can represent any required model, whereas I-World models correspond to simple regions, expressible via linear inequalities on the data attributes. Secondly, in the 2W Model, the mining operator is not predefined and acts as a template to extract a model from D-World objects. Thirdly, the definition of entails operators makes unnecessary an explicit representation in the E-World thus ensuring that mining results can be progressively analyzed.

Due to the complexity of spatio-temporal data, we believe that the proposed framework should be based on a rich formalism, capable to representing the complexities and the peculiarities of the data. We choose the object-relational technology, which combines the advantages of the relational data model and SQL language while improving them with the object oriented capabilities. The main feature of the object-oriented database model is that objects, classes and inheritance are directly supported in database schemas and in the query language. In addition, it supports extension of the data model with custom data types and methods. In particular, in an object-relational formalism, each entity is denoted as *object* and each object may have a number of attributes, such as the classic descriptive, numerical, categorical, but also attributes of object type. An object is a cohesive package that consists of a set of attributes, lists and methods.

2 The 2W Model Framework

The data mining query language system we propose is based on an algebraic framework called 2W Model. The 2W Model is a natural foundation for the development of domain-specific data mining query languages. The 2W Model views any knowledge discovery process as the interaction between two separated worlds: the data world (D-World) and the model (M-World) world. The former is a database of domain-specific entities in an object-relational format. The latter is a collection of meaningful models extracted from the raw data. Two generic operators, respectively the mining ($\kappa$) and entailment ($\vdash$) operators, were defined to represent interactions between the two worlds. The mining operator maps one or more objects in the D-World to some model object in the M-World. The entailment operator applies a model to a data object to select the set of data objects which satisfy a given model constraint. This operator enables to further analyze the mining results and, hence, to accommodate disparate pattern mining tasks into a multi-step process. Furthermore, suitable manipulation operators, defined on each world, allow the basic object handling, i.e. preprocessing (reduction in either size or dimensionality) as well as postprocessing (e.g. filtering interesting patterns).

In the object-relational context, we define two conceptual classes of objects: *Data objects*, and *Model objects* contained respectively in D-World and M-World.

The first one contains the objects to be analyzed, as well as their attributes and
their mutual relationships, instead the M-World contains objects representing different kinds of movement patterns extracted by the analysis using a mining algorithm. In the following sections we describe the objects belonging to D-World and M-World, along with the bridge operations between them instantiated in the movement context.

2.1 The Data World

The D-World represents the entities to be analyzed, as well as their properties and mutual relationships. Due to the inherent complexity of spatio-temporal data, the object-relational formalism is chosen as the underlying logic model for the D-World. The object relational model is better suited to represent the complexities and the peculiarities of the data at hands. In addition, it adds to the advantages of the relational model the ones from the object oriented paradigm, which provides direct support for objects, classes and inheritance both at the level of the underlying database schema and in the query formalism. Formally, the D-World can be viewed as a database \( D = \{ r_1(R_1), \ldots, r_n(R_n) \} \) of meaningful entities. The generic entity \( r(R) \) is a relation with schema \( R = \{ A_1 : \text{Dom}(A_1), \ldots, A_h : \text{Dom}(A_h) \} \), where \( A_1, \ldots, A_h \) correspond to descriptive attributes of the data within \( r(R) \) and \( \text{Dom}(A_1), \ldots, \text{Dom}(A_h) \) are their respective domains. Clearly, relation \( r(R) \) is defined as \( r(R) \subseteq \text{Dom}(A_1) \times \ldots \times \text{Dom}(A_h) \). In the object-relational formalism, attribute domains can be either primitive or object data types. Primitive types are assigned to simple features of the data and divide into categorical and numerical domains. Instead, object data types abstractly represent complex real-world entities, that are described through a suitable choice of (lists of) primitive attributes as well as (lists of) object data types and are equipped with application-dependent operations. Hereafter, we omit the specification of relation schema and use the resulting simplified notation to indicate the generic entity. Also, we denote by \( t \in r \) a tuple of relation \( r \), whereas notation \( t[A_i] \) indicates the value of tuple \( t \) over a schema attribute \( A_i \). In the spatio-temporal context, we define three basic types of entities, namely spatial, temporal and moving objects.

**Spatial objects.** A spatial object is an object which has a geometric shape and a position in space. Example geometric shapes are points, lines, polygons, whereas the spatial position is denoted by spatial coordinates \( X, Y \). When the coordinates are expressed in a geographical coordinate system we denote the spatial object as a geographical object.

**Temporal objects.** A temporal object is an entity which has a temporal reference (w.r.t. a time reference system) and a duration.

**Moving objects.** A moving object is an object that changes in time and space. We
denote as trajectory the spatiotemporal evolution of the position of a moving object. Hereafter, we concentrate on moving points, i.e., objects which change in time and position but not in shape (and hence can be viewed as objects exhibiting a point geometry).

### 2.2 The Model World

Movement patterns concerning data entities, their properties and relationships are modeled as suitable decision regions in the model world, which provides an elegant and expressive framework for both exploratory analysis and reasoning. The M-World can be represented as a collection \( \mathcal{P} \) of patterns, unveiled at the different stages of the knowledge discovery process. Each pattern \( p \) is associated with an object-relational schema \( \mathcal{R} \) and represents a (possibly infinite) relation \( r \) over \( \mathcal{R} \). Intuitively, \( p \) represents a decision region over the schema \( \mathcal{R} \), so that a decidable operator \( \models \) can be devised for bounding such a region.

**Definition.** A pattern \( p \) is any (possibly infinite) set \( \{t_1, \ldots, t_n, \ldots\} \) of tuples in \( \mathcal{D} \) such that, for each \( r \in \mathcal{D} \) and \( t \in r \), the assertion \( t \in p \) is decidable. We hereafter denote the property \( t \in p \) as \( p \models t \).

Different types of movement patterns can be defined to populate the M-World, on the basis of the decision regions of interest.

As an example, let \( p \in \mathcal{P} \) be a temporal annotated pattern [6] of the form \( r_1 \xrightarrow{tc_1} r_2 \xrightarrow{tc_2} \cdots \xrightarrow{tc_n} r_n \) (where \( r_i \) is a spatial region and \( tc_i \) is a time constraint of the form \( t^{(i)}_{min} \leq t \leq t^{(i)}_{max} \)). Given a relation Trajectories within the D-World, a moving point \( t \in \text{Trajectories} \) is in \( p \), if \( t \) traverses all regions \( r_i \) in sequence, and the traversal time between \( r_i \) and \( r_{i+1} \) is within the time constraint \( tc_i \).

As another example pattern, we can consider a cluster \( p \) as a set of trajectories equipped with a spatio-temporal distance function \( \text{dist} \) [19]. Here, \( \text{dist}(t_1, t_2) \) denotes the spatio-temporal affinity between \( t_1 \) and \( t_2 \); for example, they both represent entities moving towards a similar direction, or at a similar speed. Hence, a cluster is a set of similar objects, and \( p \models t \) is true whenever \( p \) includes \( t \).

Given the two worlds, in the next section we describe the two bridge operators representing the mining and entailment processes.

### 2.3 Mining and Entailment operators

The population of the M-World starting from the raw data in the D-World is performed through the mining operator \( \kappa \).

**Definition.** Given a relation schema \( \mathcal{R} \), a mining operator is defined as \( \kappa : 2^\mathcal{R} \to 2^\mathcal{P} \). \( \kappa \) represents a generic mining scheme, that receives an input relation and instantiates in the M-World an instance.

As an example, assume that \( \mathcal{D} = \{\text{Trajectories}\} \), where \text{Trajectories} is a relation in the D-World. If \( \kappa \) represents the T-pattern mining scheme in [6], \( \kappa(\text{Trajectories}) \) results into an instance T-pattern.

Once accomplished the forward population of the M-World with the required patterns, these can be employed in the opposite direction, i.e. to backwardly populate the...
D-World with further data. Interestingly, this does not involve the explicit representation of further (composite) objects as in the E-World of the 3W Model. More simply, the raw data that falls within the decision region of a certain pattern is accumulated in the D-World as new data.

The entailment operator $\models: \mathcal{P} \times \mathcal{D} \rightarrow \mathcal{D}$, is the basic step of the population process.

**Definition.** Entailment operator Let $p$ be a pattern in $\mathcal{P}$ and $r$ a relation in $\mathcal{D}$ over an object-relational schema $\mathcal{R}$. The basic population operator $p \models r$ yields a new relation including each tuple $t \in r$ within the decision region of $p$. Formally,

$$p \models r \triangleq \{ t \in r | p \models t \}$$

In practice, the entailment operator $\models$ allows to discover which data objects support a given pattern object. If $p$ is a temporal annotated pattern relative to the Trajectories, the expression $p \models \text{Trajectories}$ represents a new relation including those moving points from Trajectories, whose routes traverse the temporal annotated pattern $p$.

The entailment operator allows the application of any model to the data in a relation of the D-World always produces a further relation within the D-World. This ensures that mining results can be progressively analyzed on a par with raw data via further manipulations, as exemplified next.

Consider the case where one wishes to uncover the groups objects that move close to each other within a certain temporal annotated pattern. In such a case, temporal annotated patterns are first extracted into the M-World via a specific mining operator $\kappa$ from the Trajectories relation. This results into an instance $\kappa(\text{Trajectories})$, that groups all the unveiled patterns. The latter are then treated on a par with raw spatio-temporal data, to the purpose of identifying the trajectories inside the required pattern, which is accomplished by means of the inter-world population operator. Moving clusters [10, 19] can then be discovered in the required pattern, by applying a second mining operator $\kappa'$ to the newly obtained raw data. In the 2W Model, the algebraic formulation of the foresaid knowledge discovery workflow is $\kappa' (p \models \text{Trajectories})$, where $p \in \kappa(\text{Trajectories})$ is any suitable pattern to investigate for moving clusters. The above expression reveals the fundamental role of the population operator in the definition of a knowledge discovery workflow. Indeed, the operator enables the progressive and seamless discovery of further patterns in the raw data resulting at the end of a previous analytical process.

### 3 Daedalus: implementing the 2W Model

Hereafter we discuss the design of the DAEDALUS framework, an innovative computational engine that supports the user in specifying and refining mining objectives as well as combining multiple strategies for analyzing movement data. More specifically, the process of knowledge discovery from moving object data can be seen as an interaction between the data mining engine and the end user, where the latter formulates a query or a statement, and the former returns the required results.

Queries within DAEDALUS can be formulated by exploiting the IAPYX language. The query execution framework acts on top of a moving object database, Hermes [16], an Object-Relational storage layer for persisting both D-World and M-World objects.
D-World objects are simply persisted as instances of some predefined data-type in the rich moving-object data model natively provided by Hermes. M-World entities depend on the pursued applicative purposes and, hence, require the previous definition of suitable pattern types, to be represented in Hermes.

3.1 The IAPYX Language

In this section we present the IAPYX language which is the core component of DAE-DALUS and that implements the 2W Model vision of the data mining query language. IAPYX comprises several different statements, aimed either at manipulating and processing raw data, or at mining patterns and possibly use their entailment in order to populate further some moving points. In particular, data manipulation is accomplished by means of the set of methods defined on the objects. In the following we present some of the spatio-temporal primitives inherited from the Hermes implementation of Moving point object:

**f.intersection.** This primitive computes the spatio-temporal intersection between the moving point and a spatial object. When the moving point doesn’t intersect the spatial object the result is null.

**at.period.** It cuts the moving point in a specific period of time defined by a temporal object. When the moving point is not defined in the period of time the result is null.

**f.begin, f.end.** They return the initial and the final spatial coordinates of the moving point, respectively.

**f.initial_timepoint, f.final_timepoint.** These functions are similar to the previous ones, referring to the time.

Analogously, other manipulation methods has been defined on data and model objects, which we omit here due to lack of space.

The Mining operations are aimed at executing specific mining algorithms and in materializing their results into object-relational tables. Currently, we support two main operators, namely the extraction of T-patterns and clustering. Finally, entailment operations apply patterns to moving points and select those moving points complying with the patterns. The complete grammar of the IAPYX language is formally defined below:

```plaintext
Examples of query statements are reported in section 4. In the next section we describe the implementation details of the objects of the two worlds, Data and Model.
```
3.2 Data and Model objects

DAEDALUS build on top of the Hermes moving object database (MOD) an implementation of both data and model objects. This allows to explicitly map entities into pre-existing spatio-temporal objects. Indeed, since Hermes is an extension cartridge of Oracle, it provides access to the Oracle spatial native data model, (the SDO_GEOMETRY) for representing spatial objects. Finally, temporal objects, as defined in [16], represent either specific points in time, intervals or periods. Again, they can be easily represented into the Oracle object-relational model as objects having a timepoint attribute and a duration.

The temporal object represent a precise point in time, an interval of time or a period of time. The implementation of it is the following:

```
TEMPORAL OBJECT {
  begin TIMEPOINT OBJECT;
  duration NUMERICAL)
}
```

The object-relation definition of the moving point object reflects the trajectory definition given in section 2.1. Indeed, each trajectory is represented as a list of segments composed by a spatial component and a temporal component, as shown below:

```
MOVING POINT OBJECT {
  p LIST OF SEGMENT {
    space SPATIAL SEGMENT;
    time TEMPORAL OBJECT}
  xe NUMERICAL;
  ye NUMERICAL;
  functionType CATEGORICAL [...]}
```

Where the categories of functionType are: Linear, Arc, or Constant. This functions are used to interpolate the position of the moving point between the \([x_b, y_b]\) and \([x_e, y_e]\). This representation of Moving point is previously presented in [16].

The underlying object-oriented model in Hermes (and its Oracle kernel) also allows to directly define patterns as objects. As previously pointed out, the objects defined inside DAEDALUS are T-Pattern and Cluster. Here, a T-Pattern is defined as a list of region-interval pairs, plus a numerical attribute which represents the support, as reported below:

```
T-PATTERN OBJECT {
  t LIST OF REGIONS-TIME OBJECT{
    region SDO_GEOMETRY; interval INTERVAL};
  support NUMERICAL
}
```

The cluster representation is presented below:

```
CLUSTER OBJECT {
  t LIST OF PAIR OBJECT{
    mp_id NUMERICAL; eps NUMERICAL
}
```

Informally, a cluster can be represented as a list of moving points. Further properties (such as core distance, in the style of [19]) can be made explicit as well.
3.3 Architecture

The software architecture of the DAEDALUS system is reported in Fig. 2. The data and model object repository is Oracle version 11g [17] extended with a number of object types ranging from the Hermes moving objects [16] to the model object types defined ad-hoc for T-Pattern and Cluster. The core component of DAEDALUS includes the Controller, the Parser, the DBManager, the Objects translator and the Algorithms Manager. The Controller processes the IAPYX statements and coordinates the tasks performed by the other components, while the Parser interprets each individual statement. The DBManager provides centralized access to the data layer and the Algorithms Manager is activated by the Controller for executing algorithms called during the processing plan. The result is converted into an object-relational representation exploiting the Objects Translator and stored into the object-relational data storage through the DBManager. The GUI component provides the user with a front-end, where to formulate IAPYX statements and visualize the corresponding results.

It is worth noticing that further components with new roles can be integrated within DAEDALUS with minimum effort and maintenance, i.e. by simply specifying their interfaces. Indeed, modularity and extensibility are also major aspects of the inner design of DAEDALUS, as it supports the integration of plug-ins devoted to extend the capabilities of the system. Figure 2 highlights in yellow the pluggable components. Precisely, multiple mining algorithms can be deployed into the Algorithms Manager environment, to the purpose of enlarging the spectrum of the supported mining operators. The system provides also a way to integrate new Data objects or new Model objects representations in the Objects Translator component. This allows a developing user to add new mining operators and possibly, new model objects.

4 Application Example

In this section, we perform a qualitative evaluation of DAEDALUS by means of a methodological approach inspired by similar approaches in the literature (e.g. in [7]) for the illustration of the expressive power of data models and related query languages. In particular, we settle in the context of a specific case study, i.e. the identification...
(and further manipulation) of common behavior of trajectory data and formulate some analytical queries. The purpose is providing a taste of the expressiveness and usefulness of the IAPYX language as well as the computational power of DAEDALUS. The aim of the analysis examples reported here is to highlight the main features of the DAEDALUS system for analyzing movement of peoples in the Milan area. The analyzed trajectories data is a raw dataset of points, representing GPS revelation of movement of people in Milan\textsuperscript{3}. We will see in the next problem statements the specific analysis tasks that have been performed exploiting the powerful of the proposed data mining query language.

The first basic task is to discover a set of common behaviors of moving users in different days of the week.

The preliminary step is to reconstruct the moving points built from a set of GPS localizations. Such localizations are assumed to be already available within the D-World as raw data organized into a table called Milan\_gps\_traces. The individual localization refers to a certain user through a corresponding id and provides the relative longitude and latitude coordinates along with a temporal stamp. The resulting moving points are organized into another table of the D-World, called Milan\_trajectories. The below statement builds the required moving points by relying on the BUILD auxiliary operator.

\begin{verbatim}
CREATE OBJECT Milan_trajectories AS BUILD MOVING_POINT
FROM (SELECT userid,lon,lat,d DATETIME
FROM Milan_gps_traces)
WHERE MOVING_POINT.max_duration = 3600 AND
      MOVING_POINT.max_distance = 0.5 AND
      MOVING_POINT.max_speed = 0.1
\end{verbatim}

\textsuperscript{3}The dataset has been donated by Octotelematics to use inside the GeoPKDD project consortium (http://www.geopkdd.eu). The entire dataset tracked one week of movements of about 17,000 GPS-equipped cars travelling around the Milan area.
The next two steps, once the trajectories objects have been built (fig.3), are to exploit the \texttt{intersection} and the \texttt{at\_period} primitives to obtain a table of trajectories in the city center, a specific object (with id equals to \texttt{Center}) in the Areas table given as background knowledge. Furthermore the resulting trajectories are split by day of the week, indicated by Periods table:

```sql
CREATE TABLE Traj\_center AS
SELECT t.id, t.object.\texttt{f\_intersection}(a.object) as obj
FROM Milan\_trajectories t, Areas a
WHERE a.id = 'center' and obj is not null
```

```sql
CREATE TABLE Traj\_center\_by\_day AS
SELECT t.id as tid, t.object.\texttt{at\_period}(p.object) as cut, p.id as pid
FROM Traj\_center t, Periods p
WHERE cut is not null
```

![Figure 4: The result of the intersection between the set of trajectories and the center area](image)

To this point (fig.4), we can discover the most frequently followed paths on each day of the week, by unveiling the T-Patterns from the corresponding routes. The following query shows the mining call relative to the Sunday trajectories which have pid equals to \texttt{Sunday} in the \texttt{Traj\_center\_by\_day} table:

```sql
CREATE MODEL Pattern\_Sunday AS MINE T\-PATTERN
FROM (SELECT t.tid, t.cut
	FROM Traj\_center\_by\_day t WHERE t.pid = 'Sunday')
WHERE T\-PATTERN.size = 0.004 AND T\-PATTERN.time = 180 AND T\-PATTERN.support =0.02
```

In the above, \texttt{size}, \texttt{time} and \texttt{support} represent parameters of the T-pattern mining algorithm. The interested reader is referred to [6] for further details.

The output of the foregoing mining call is the set of Sunday frequent paths with their typical travelling time (fig.5). The resulting patterns are stored as T-pattern objects in the table Pattern\_Sunday.

We want to analyze further a specific common behavior extracted in the previous analysis dividing the trajectories into smaller meaningful groups. Given the computed
set of models, we select one of them (i.e. the one with id=12) as interesting. The first step is to extract all the trajectories that satisfy that particular model exploiting the entails operation:

```
CREATE ENTAILMENT Entail_Pattern_id12
FROM (SELECT t.id, t.object, m.id, m.object
     FROM Milan_trajectories t, Pattern_Sunday m
     WHERE m.id=12)
```

In fig.6 we show the result of the entail operation. With this subset of trajectories we build a new set of model using the optics algorithm of clustering:
Then we can create the entailment and select all the trajectories from the original dataset which support one of the cluster created by the process.

```
CREATE MODEL Cluster_on_Pattern_12 AS MINE OPTICS
FROM (SELECT t.id, t.object
       FROM Milan_trajectories t, Entail_Pattern_12 e
       WHERE e.id = t.id)
WHERE OPTICS.npoints = 10 AND
      OPTICS.eps = .01 AND OPTICS.method = END
```

The Graphical User Interface of DAEDALUS allows to visualize the subset of trajectories in different colors using the e.id2 which is the cluster identifier to have a visual representation of them.

Having a dataset of trajectories and a set of regions representing the districts of the city, we want to build the OD Matrix between them. The following script implements the building of the OD Matrix starting from a set of trajectories and a set of regions selected as interesting for this application (i.e. the regions can realize a simple grid or the districts of the city).

```
CREATE TABLE t_begin AS
SELECT t.id, t.object.f_begin() as object FROM Milan_trajectories t

CREATE TABLE t_end AS
SELECT t.id, t.object.f_end() as object FROM Milan_trajectories t

CREATE TABLE tr_begin AS
SELECT t.id as id1, c.id as id2 FROM t_begin t, od_regions c
WHERE GEOINTERSECTION(t.object, c.object, 0) is not null

CREATE TABLE tr_end AS
SELECT t.id as id1, c.id as id2 FROM t_begin t, od_regions c
WHERE GEOINTERSECTION(t.object, c.object, 0) is not null

CREATE TABLE od_matrix AS
SELECT b.id2 AS fromRegion, e.id2 AS toRegion, count(*) AS num
FROM tr_begin b, tr_end e where b.id1 = e.id1
GROUP BY b.id2, e.id2 ORDER BY count(*) DESC
```

The **OD Matrix** is a well known result used for traffic management and it is considered as basic step for a lot of traffic analysis. It’s important to notice that this script can be executed also on different slice of the data: i.e. different days of the week, or different periods during the day or for only the trajectories that support a specific T-Pattern or Cluster.

### 5 Conclusions and Future Work

The research results of the Data Mining Query Language task in GeoPKDD produced the framework DAEDALUS, consisting in both an algebraic model 2W Model, the IAPYX language for progressively mining and querying movement data, and an innovative computational engine for processing IAPYX statements. DAEDALUS is proposed as an unifying framework for incorporating movement pattern-mining tools as specific components of the geographic knowledge discovery process.
The short term future work are moving towards the enrichment of the DAEDALUS system with new movement mining algorithms to be exploited in real case studies. For example, the Milan dataset is being used in a larger application where DAEDALUS is aimed at performing disparate analysis tasks for Traffic Management.

Concerning the objective of a long term research on this issue, we envisage a more tighter integration with both the reasoning and visual analytics tools, to provide the user with more powerful pre and post-processing capabilities, thus supporting the whole geographic knowledge discovery process.

References


[4] Jiawei Han and Yongjian Fu and Wei Wang and Krzysztof Koperski and Osmar Zaiane DMQL: A data mining query language for relational databases SIGMOD’96 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD’96)


Where the state of the art was

A flood of data related to moving objects is available today, and it will expand in the near future, particularly due to the automated collection of positioning data from mobile phones and other location-aware devices. This flood of data enables a novel class of applications, where the discovery of consumable, concise, and applicable knowledge is a key step. The presence of a large number of location-aware wirelessly connected mobile devices presents a growing possibility to devise location-based services exploring spatio-temporal data representing the footprints of moving objects, which we call raw trajectories. They represent the spatial track of a moving object in a fixed time interval.

There are currently interesting practical opportunities for exploring various kinds of spatio-temporal relationships, including trajectory patterns. From the point of view of the application, trajectory patterns can be
considered as the spatio-temporal evidence of movement behaviour. For example, in a traffic management application, a trajectory pattern could represent a congestion or could disclose a frequent origin-destination path of working activities. Clearly, the discovery, representation and analysis of trajectory patterns challenge the research community with respect to methods for aggregating, generalising and explaining those patterns. In this scenario, data mining techniques play a fundamental role due to the fact that data mining is the application of specific algorithms for extracting patterns from data. However, data mining algorithms recently developed for mining raw trajectories have mainly produced trajectory patterns which are difficult to be interpreted in an application domain. The main problem has been the difficulties to correlate those patterns with movement behaviour in order to improve our knowledge, for example how to avoid congestion or making things easy for pedestrians.

Movement behaviour is a particularly complex process, since several factors can affect the movement itself, including the nature of the moving object, its motivation and the geographic environment where the object moves through. Our main research premise is that a standard pattern extraction algorithm, despite being highly specialised, cannot cope with such a complexity in terms of trajectory patterns interpretation, understanding and evaluation. Indeed, as defined by Dodge et al. in (Dodge et al., 2008), a movement behaviour depends on the context since each movement takes a particular meaning when happens in a geographical environment. For example, a specific trajectory pattern showing a group of trajectories moving slowly in a city may be identified as a lane forming due to a traffic light or a traffic jam, depending on the contextual geographical knowledge (for example if the moving objects are moving in dense populated neighbourhood, or on a motorway).
Contribution of the Reasoning Task

The basic idea is to define an approach to incrementally enriching raw trajectory data with context geoinformation. The first step is the definition of semantic trajectory as sequence of stops (movement suspension) and moves (the actual movement). A further enrichment step exploits the knowledge capabilities provided by the ontology to integrate context geoinformation with the semantic trajectory patterns. Eventually, a reasoning step allows to infer new knowledge (e.g. the inferred movement behaviour) that can be added back to the ontology. For example, semantic trajectories patterns showing that typically stops are happening in tourist places can be associated with the movement of tourists. In contrast, semantic trajectory patterns stopping at offices in the morning and residential areas in the evening can be identified as a home-work routing behaviour.

Semantic Enrichment Process

The analysis of raw trajectories strongly relies on the contextual geographical knowledge. For example, the analysis of urban movement can result in meaningless results if each raw trajectory is not associated to the moving object (e.g. car, dog, person, etc), features in the geographical
space (e.g. building, street segments, hospital, etc), landmarks (e.g. Eiffel Tower, Everest, etc) and events (e.g. football matches, accidents, etc). The conceptual representation of a trajectory via its association to several semantic concepts is essential to provide the means for disclosing valuable information about movement behaviour.

From a user perspective, representing trajectories using geographical knowledge increases their understanding, but also helps users to query about movement behaviour using their own terminology. For example, it is more natural to a urban planner write the query: "Give me all tourist activities.", than formulate some complex query that expresses the meaning of tourist activities. Let’s assume, for example, that tourist activities are identified by trajectories having tourist behaviour as people stopping in tourist places (such as Museums) and stopping also in accommodation places. Based on this definition, the following query should be issued: "Give me all trajectories that contains stops located at a tourist place and also contains stops in accommodation places .". In the former query the concept tourist activities is a main behaviour concept while in the latter query this concept is implicit in the query. Furthermore, the query itself can refer to concepts (tourist place, accommodation place) that are the “semantic generalisation” of “atomic” concepts stored in a knowledge base (museums, hotels, …). The corresponding query could be rephrased as “Give me all trajectories which stop their movement inside a museum or near a monument (o any other geographical object defined as tourist place) and stop their movement inside a Hotel or a B&B (o any other geographical place defined as accommodation place)”. It is clear that going from such kind of query on raw trajectory data to the former desired query “give me all tourist activities”, an incremental process should have built where semantics and reasoning play a central role.

The research challenge relies on developing an approach that employs formal ontologies to enable an enrichment process that augments the semantics of both raw trajectory data and mined trajectory patterns.

Ontologies have certainly become a research topic in several disciplines, ranging from philosophy, geography, geomatics up to machine learning and artificial intelligence. The definition given by (Gruber, 2008) is used to define ontology as “a technical term denoting an artifact that is designed for a purpose, which is to enable the modeling of knowledge about some domain, real or imagined.” Such ontologies determine what can be represented and what can be inferred about a given domain, using a specific formalism of concepts. The main objective of our current research is to exploit a process that can give the user a semantic interpretation of trajectory patterns in terms of movement behavior. The process we propose is based on the abstraction of a semantic model of trajectories, the application of
data mining algorithms on them for generating trajectory patterns, and a final further enrichment step that consists of exploiting an ontology reasoning engine for inferring movement behavior. Figure 2 illustrates the flow of such an enrichment process.

An important added value of relying on an ontology formalism is that it comes with an embedded reasoning engine that gives an increased expressive power for movement behavior interpretation, putting together the different levels of knowledge of an application domain. This is further discussed in the next section.

Fig. 2. The trajectory semantic enrichment process

**The Semantic Representation of Trajectories**

The trajectories captured by positioning systems (e.g. GPS and GSM equipments) are usually represented as a sequence of \(<\text{sample point}, \text{time}>\) pairs, called *raw* trajectory. The main characteristic of data acquired by those mechanisms is that they rigorously reveal the geometric facet of a trajectory, but suffers from a lack of semantics representing the movement behavior of the moving object.

A trajectory, as defined in (Spaccapietra et al., 2008), is the *user defined record of the evolution of the position of an object that is moving in space during a given time interval in order to achieve a given goal*. Moving objects do not necessarily move continuously during a trajectory. Consequently, trajectories may themselves be semantically segmented by defining a temporal sequence of time sub-intervals where alternatively the
object position changes and stays fixed. We call the former the *moves* and the latter the *stops*. We can then see a trajectory as a sequence of moves going from one stop to the next one (or as a sequence of stops separating the moves). Yet, identifying stops (and moves) within a trajectory depends on the application requirements. Formal definitions of stops and moves are given in (Spaccapietra et al., 2008), whereas the semantic conceptualisation of a trajectory has been introduced in (Alvares et al., 2007), and in (Wachowicz et al., 2008). In Figure 3 we show our ontology-based representation of stops and moves, in terms of concepts and their relationships.

More precisely, boxes represent the main concepts whereas arrows represent relationships between two concepts. Every Trajectory is composed of stops (\textit{trajCompOfStop}) and almost every stop is connected to other two stops by two moves. Therefore, we have represented the concepts of \textit{Stop} and \textit{Move} and four explicit relations connecting them, namely, \textit{fromStop}, \textit{toStop}, \textit{inMove} and \textit{outMove}. Only the first and the last stop are connected with a single move. Furthermore, every stop is connected to an interval (\textit{stop_Has_Time}) that represents the time of the stop.

![Fig. 3. The semantic trajectory representation](image)

**Mining Semantic Trajectories**

In this section, our objective is to demonstrate how semantic trajectories can be associated with trajectory patterns and later with movement behavior. The examples given below might look straightforward, but it is important to point out they are used to illustrate the feasibility of our approach.

The problem of finding frequent patterns can be stated as: given a data set D and a user specified minimum support (\textit{minsupp}), the task of fre-
quent pattern discovery is to find all sets of items with support at least \( \text{minsupp} \).

Frequent patterns (Agrawal et al., 2003) can be applied to both stops and moves independently. Frequent Stops (Moves) means finding sets of stops (or moves) that are most frequent. More details of patterns on semantic trajectories can be found in (Alvares et al., 2007), and in (Bogorny et al., 2008).

The following example represents frequent stop patterns:

\[
\text{Hotel[weekday], Museum[weekday]} \quad (s=0.21)
\]

This pattern is a mining expression and states that people that stay in a hotel during the week also go to (in any order) a Museum with a support of 21%. The conceptual representation is depicted in Figure 4. A Frequent-StopPattern is defined as a set of Stop, by means of the relation \( \text{fpContainsStop} \).

![Fig. 4. Trajectory patterns representation](image)

Similarly, in (Bogorny et al., 2008) authors have defined frequent moves and sequential stops and moves. Besides, it is important to point out that several other mining algorithms have been defined in the literature to extract patterns from both raw and semantic trajectories (Nanni et al., 2008). However, we focus here only on frequent patterns since the proposed approach could be applied to other patterns as well.

**Reasoning on Trajectory Patterns**

Once semantic trajectories and mining models have been defined, the next step is to express them using an ontology formalism, to combine them with domain knowledge, in order to define complex movement behavior.
Indeed, domain knowledge is composed of geographical knowledge about where the trajectory movement takes place using the application domain of reference, such as tourist or traffic management applications. Here, we focus on an urban human movement. The geographical knowledge is based on an urban ontology that describes the places where people move through (e.g. museums, hotels, universities, theatres, etc). Figure 5 shows an example of the ontology obtained by combining the three aspects described so far. Indeed, the core part of the ontology is the semantic trajectory, appearing on the top side of the picture. The central part of the ontology figure shows the geographical knowledge, linked to the Stop concept by means of the is_at property, stating that a stop happens at a specific city place. Notice that the taxonomy defining interesting city places classifies two different types: tourist and accommodations places. Each of these semantic concepts represents a number of geographical features belonging to the application domain, such as monuments, hotels, museums and so on.

![The integrated ontology](image)

**Fig. 5.** The integrated ontology

New application concepts are described by means of *axioms*. Each axiom is a combination of logical operators that implicitly describes a class of objects. In the following example, we are interested in characterising trajectory patterns according to a movement behaviour by giving a possi-
ble interpretation of mined patterns (that in the ontology are represented as instances) respect to that knowledge. In the following we show two examples of axioms, expressed in OWL syntax, to specify a typical tourist behavior.

TouristActivity =
fpContainsStop some (Stop and (is_at some TouristPlace)) and fpContainsStop some (Stop and (is_at some AccomodationPlace))

The above axiom defines a tourist activity as a frequent pattern that contains stops that are located at a tourist place (e.g. Museum, Monument …) and contains stops located at accommodation places (e.g. Hotel, B&B,…). Similarly, we can exploit this definition for characterizing individual trajectories, thus defining the concept of tourist trajectory.

TouristTrajectory =
trajCompOfStop some (Stop and (is_at some TouristPlace)) and trajCompOfStop some (Stop and (is_at some AccomodationPlace))

The added value of having such an ontology-based approach, allows us to define axioms in terms of high-level semantic concepts, abstracting away from the geometric coordinates of the geographical features. Indeed, in this approach, each stop is treated as a semantic concept (e.g. Museum, Monument) instead of using the spatial coordinates. Moreover, the domain ontology gives a further abstraction level on top of semantic patterns, i.e. the AccomodationPlace class that subsumes hotels and B&B. Therefore, we can reason in terms of AccomodationPlace instead of each specific subclass (Hotel) or specific instance (CentrumHotel). As a further analysis step, the availability of a reasoning engine allows to build an automatic system for the characterisation of trajectory patterns in terms of people behaviour.

The Athena System

In order to better understand the feasibility of this approach, we have implemented a system where trajectory patterns are first extracted by means of a frequent pattern algorithm, then they are classified in the appropriate
behavior classes, according with a semantic trajectory ontology and related axioms.

As a first step, we built a prototypical ontology using Protégé (Protégé, 2008) editor with Pellet reasoner (Pellet, 2008). Here, stop and moves and patterns data were imported from a database directly into Protégé. The obvious drawback of this approach is that importing very large raw trajectory data sets is not efficient, since both Protégé and Pellet are not scalable for large datasets (i.e. large collection of individuals). Indeed, we have to point out that the use of a formal ontology as a solution for enriching trajectory mining patterns is not a "silver bullet". In fact, the knowledge representation by means of a formal ontology brings a fundamental advantage of having ready-to-use inference system (i.e. reasoner) providing useful reasoning services for granted. However, reasoning on individuals (i.e. ABox reasoning) are not scalable in any of the Description Logics language flavours and both trajectory and data mining data pose a big challenge in this respect. Actually, reasoning scalability is an important open issue investigated in the Description Logic research domain.

For this reason, we investigated the use of Oracle11g Semantic Technologies (Oracle, 2008) as an ontology storage system and associated reasoner. The advantage of using Oracle technologies is that it is optimized for handling large datasets of individuals. Oracle semantic technologies store ontologies, represented as RDF triples, in relational tables. Both defined and inferred knowledge is stored, which results in faster processing of queries as all inferences are readily available. Oracle reasoning services are executed on demand and include traditional inference classification, satisfiability, subsumption, and instance checking. The drawback is that the Description Logic language that Oracle implements, called OWLPRIME, is a subset of OWL DL and has some limitations in expressiveness. To overcome these OWLPRIME limitations, Oracle provides rules that allow the designer to complement the basic OWLPRIME reasoning with more sophisticated and application dependent inference mechanisms. The rules can be added to a rule base that can be used conjointly with the ontology during semantic query execution.

Architecture

The Athena architecture overview is illustrated in Figure 5. Here, the user directly poses a query using the ontology concepts where trajectories/patterns are classified by the reasoner. The ontology is populated by instances coming from relational tables storing semantic trajectories, patterns and geographical features.
More in detail, we used a dataset of trajectories coming from GPS installed on cars moving in the Milan area\(^1\). This dataset has been stored as raw trajectories in the moving object database Hermes (Pelekis and Theodoridis, 2006), which is based on Oracle. From them, we extracted a collection of stops computed over a simplified geographical domain containing Museums, Theatres, Hotels, Bed and Breakfast, Monuments. From these database tables, we developed a Java procedure described as follows:

- starts from a dataset of raw trajectories and geographical places and computes the semantic trajectories stored as table of stops;
- runs the frequent pattern algorithm on stops table (here we used PatternList (Bonchi et al., 2006)) and stores back the results in the frequent patterns tables;
- imports the ontology from Protégé to Oracle 11g;
- reads the database tables and translates them into RDF triples;
- inserts the triples in the ontology as instances;
- runs the reasoner to infer new triples.

### Real Case Scenario

The objective of the analysis is to understand tourist movements in Milan. A first simple analysis is to visualise all tourist trajectories, i.e. trajectories

\(^1\) This dataset has been collected by Octotelematics and has been given for internal use inside the consortium of the European Project GeoPKDD (http://www.geopkdd.eu). It counts 17000 trajectories of one week.
that contain stops at tourist places and accommodation places (see step 3 and 4 below). A further objective is to understand the movement behavior among tourists. Starting from the semantic trajectories computed by system, expressed as sequences of places, we can use a data mining algorithm to discover the trajectory frequent stop patterns. Among these trajectory patterns, we can choose a specific pattern of interest (i.e. MadonninaB&B, Duomo and Castello at step 5 below) to find the trajectories that support this patterns. This operation can be done by joining together the semantic trajectory and raw trajectory data using the IDs (see step 6 below). A further analysis step is to group these similar trajectory patterns in subgroups depending on their common destination (using a clustering algorithm (Rinzivillo et al., 2008)), thus discovering that indeed two types of movement behavior are present: tourists coming from outside the city and moving to the centre, and tourists moving from the city centre to outside (see step 7 below).

This analysis process is illustrated in details below:
1 – The objective of the analysis is to discover tourist common behaviour in the city. First of all we select all the geographical places that are interesting for the analysis (corresponding to the ontology classes) such as Hotel, Bed and Breakfast, Museum, etc. and the original raw trajectories in Milan. The following picture shows the Milan trajectories and some colored areas, corresponding to geographical objects of interest for the analysis, encoded as classes in the integrated ontology shown in Fig. 4.

![Diagram of Milan trajectories with colored areas corresponding to classes in the integrated ontology.]

2 – The execution of Athena computes semantic trajectories, data mining patterns, and imports them, along with geographical objects, in Oracle11g Semantics Technologies. A map file is provided to define the correspondence between ontology concepts and database tables. When everything is loaded, the system calls the reasoner over the ontology and the imported instances in order to infer new facts (e.g. tourist activities).

\[
\text{Execute Athena(Map file).}
\]

3 – The inferred facts are stored in the ontology, therefore we can query\(^2\) the system to select the trajectories that are kind of tourist according to the definition in the ontology. Here we discovered that trajectories with ID Trajectory192, Trajectory345 have been classified by the ontology as tourist.

\[
\text{SELECT m, r FROM table}
\]

\(^2\) Parts of queries are omitted due to query readability. The rest is the specification of the name of the semantic model created, the type of rules used and the aliases:

\[
\text{SEM_Models('MODELGEOPKDD') ,SEM_rulebases('owlprime') ,SEM_ALIASES(SEM_ALIAS('','http://www.owl-ontologies.com/GeoPKDDOnto.owl#')),null)).}
\]
Tourist trajectories can be joined with the raw trajectories exploiting the ID. Indeed, the geometric coordinates allows us to draw them on a map. Below in yellow the subset of the raw geometric trajectories showing a tourist behavior, based on the definition of the described urban ontology.

To discover common behavior, we exploit the Frequent Stop Patterns data mining algorithm, and we select one of the resulting computed patterns (id=47). This pattern is characterised by people stopping at a specific B&B, namely MadonninaB&B, and two different monuments, Duomo and Castello.

In order to understand which are the raw trajectories that support this specific tourist frequent pattern, we can, through the use of the Stop ID, Trajec-
tory ID, and the StopBelongsToTraj ontology property, join the pattern with raw trajectories, shown in the picture below. This is a powerful capability of the Athena system since it allows us to combine, in the same query, semantic information with raw geometric data.

7 – A further data mining analysis step can be run using the found tourist trajectories, to have a better understanding of common tourist behavior. Here we used a density-based clustering algorithms on trajectories. The result shows two very well defined groups of trajectories: from outside the city to the center (A) and vice versa (B). This can suggest to the user that the first group describes tourists in their first day of holidays and the second group describes tourists in their last day of holidays.

This example, despite its simplicity, shows the feasibility of the proposed ontology-based semantic enrichment process over raw trajectory data, exploiting the synergy between geographical knowledge, automated reasoning, data mining to perform very complex analysis of movement be-
haviour processes. It is interesting to notice that a different kind of analysis is possible with this methodology. Indeed, finding patterns which have not been classified under any behaviour definition ontology class will result in an “unknown” behaviour. This in turn may call for a further in-depth analyses based on the iterating steps of the knowledge discovery process, possibly arriving to a better understating of new unexpected movement behaviour.

Future Work

In the GeoPKDD Reasoning Task we proposed an approach to provide the interpretation of movement behavior. This approach exploits a formal ontology as a representation and reasoning mechanism that enables semantic interpretation of raw trajectories and their mining patterns. The obtained ontology is immersed in a domain ontology representing geographical knowledge. This allows getting the maximum semantics in term of encoded application geographical knowledge. Indeed, this ontological approach allows us to reason in terms of semantic concepts. This added value, when applied to mining patterns, produces an automated interpretation of them within an application domain, which we have called as semantic enrichment process.

The research carried out so far on exploiting semantics and reasoning for movement understanding has just tackled the tip of the iceberg. Much investigations are needed towards both short term and long term results.

Short term research work is towards the development of a more robust and complete Athena system. First, we are investigating the expressiveness of OWLPRIME combined with Oracle user-defined rules in order to get maximum expressivity while retaining a good computational efficiency. Second, we aim at improving our system by integrating more data mining algorithms. Indeed, the proposed approach can be applied also to raw trajectories and associated mined patterns. Obviously, in this case the ontology misses the semantic trajectory part and no relations can be expressed between trajectory and pattern, thus offering less expressiveness. Third, we are working towards a more effective mapping between the ontology and the database. Indeed, so far the mapping is done manually using a static mapping file. An automatic or semiautomatic approach is envisaged.

More on a long term, another challenging issue that could be investigated is how to exploit the semantic ontology in the steps of a knowledge discovery process, specially in the pre-processing step. The aim is to de-
rive a suitable raw trajectory data in order to avoid extracting uninteresting patterns. A further challenge will be the combination of ontology-based semantic enrichment process with visualization techniques. Therefore, we are planning to investigate an integration of the proposed approach with visual analytics techniques. The idea is to enrich the Athena approach with more sophisticated visualization features. But, also visual analytics may benefit from the Athena approach to have support in discovering more meaningful, semantic based patterns. This could offer better support for user evaluation of inferred movement behavior.

References


http://www.w3.org/TR/owlfeature


Abstract

The document describes what was achieved in the project GeoPKDD in the research on visualisation and visual analytics of movement data. The project has brought into existence an array of new methods enabling the analysis of really large collections of movement data. Some of the methods are applicable even to data not fitting in the computer main memory. These include the techniques for database aggregation, cluster-based classification, and incremental summarisation of trajectories. The remaining methods can deal with data that fit in the main memory but are too big for the traditional visualisation and interaction techniques. Among these methods are interactive visual cluster analysis of trajectories and dynamic aggregation of movement data. The visual analytics methods are based on the interplay of computational algorithms and interactive visual interfaces, which support the involvement of human capabilities for pattern recognition, association, interpretation, and reasoning. The project has also moved forward the theoretical basis for visual analytics methods for movement data. We have identified analysis tasks and problems requiring further research.

Introduction: the concept of visual analytics

The concept and research discipline of Visual Analytics emerged in response to the grand challenge posed by the overwhelming and rapidly growing amounts of diverse data and information from numerous sources. People need to make sense from these oceans of diverse data in order to make right and timely decisions. A grand challenge is to adequately support the analyst in

- distilling the relevant nuggets of information from disparate information streams;
- understanding the connections among relevant information;
- gaining insight from data.

Current technologies cannot support the scale and complexity of the growing analytical challenge. On the one hand, purely automatic analysis procedures are only possible for well-defined problems whereas most of the real-world problems are ill-defined. Such problems can only be solved with the participation of human analysts applying their creative and versatile thinking, imagination, multifaceted knowledge and experience, and common sense. On the other hand, while the computer performance grows at a rapid pace, basic human skills and abilities do not change significantly, and therefore large-scale problems have to be reduced to a scale that humans can comprehend and act on. Hence, the advances in the computer technology by themselves are insufficient. Moreover, they are doomed to be under-utilised unless some principally new solutions are found which fundamentally improve the division of labour between humans and
machines so that the computational power could amplify the human perceptual and cognitive capabilities. Finding such new solutions is the task of Visual Analytics.

The term “Visual Analytics” stresses the key role of visual representations as the most effective means to convey information to human’s mind and prompt human cognition and reasoning. As is stated in (McCormick et al. 1987, p.3), “An estimated 50 percent of the brain's neurons are associated with vision. Visualization <…> aims to put that neurological machinery to work”.

Visual Analytics is defined as the science of analytical reasoning facilitated by interactive visual interfaces (Thomas & Cook 2005). Visual analytics combines automated analysis techniques with interactive visualizations so that to support synergetic work of humans and computers where the computational power amplifies the human abilities and is, in turn, directed by human’s background knowledge and insights gained.

Data and problems involving geographical components are an appropriate target for Visual Analytics (Andrienko et al. 2007, 2008a). Generally, geographic analysis relies heavily upon the human analyst’s sense of the space and place, tacit knowledge of their inherent properties and relationships, and space/place-related experiences. These are incorporated into the analysis through the use of appropriate human-oriented representations of the space, such as maps.

Movement data, which have been the focus of GeoPKDD, are inherently complex as they involve (geographical) space and time. In addition to their own intrinsic complexities, these components are interdependent, which multiplies the overall complexity. As a result, movement data cannot be adequately modelled (at least at the present time) for a fully automatic analysis. At the same time, movement data, which are mostly acquired by automatic position tracking, are usually very poor semantically. The records basically consist of time stamps and coordinates. Semantic interpretations must emerge as a result of exploration and analysis where a human analyst plays the key role. Appropriate visual representations of movement data and outcomes from automated analysis procedures are paramount for this process.

**State of the art before the project start**

**Visualization of individual movements**

The most common techniques for the visualization of movements of discrete entities are static maps with directed linear symbols (Vasiliev 1997), animated maps (Andrienko et al. 2000), and space-time cubes, where two dimensions represent space and the third dimension represents time. The latter technique has been introduced by Hägerstrand (1970) and recently gained high popularity (Kraak 2003, Kwan & Lee 2004, Kapler & Wright 2005) thanks to modern computer technologies providing opportunities for interactive three-dimensional visualization. Map and cube displays are often complemented with graphs and diagrams exhibiting position-irrelevant aspects of the movement (Kraak 2003, Mountain 2005).

However, purely visual and interactive techniques are not scalable to data about movements of multiple entities and to long time series of observations, even when the data are not really massive. It may be difficult to analyze just two trajectories represented
on a map or in a space-time cube if they have common locations or segments, or even a single long trajectory with loops or repeated segments, and a display of ten trajectories may be completely illegible. Hence, the limitations from the side of human perception are much stricter than the limitations from the side of computer memory and performance.

**Aggregation-based visualization**

The approaches to visualisation of larger amounts of movement data involve reducing the size of the data by means of either aggregation or filtering. Filtering, in fact, does not solve the problem: at each moment, the analyst can see only a small portion of the data and can hardly reconstruct the overall picture. The state of the art in aggregation-based approaches is well represented in the works by D. Mountain and colleagues (e.g. Dykes & Mountain 2003, Mountain 2005). One of the techniques is the temporal histogram, which represents the data aggregated by time intervals, for example, the number of locations visited or the distance travelled. Spatial aggregation is done by imposing a regular grid over the territory and counting trajectory points fitting in each cell. The resulting densities are visually represented by colouring or shading the grid cells on a map display. The densities counted for consecutive time intervals can be shown on an animated map display. A grid with densities can be treated as a surface. Smooth surfaces can be built using kernel methods. There are computational methods for uncovering various topological features such as peaks (maxima), pits (minima), channels (linear minima), ridges (linear maxima), and saddles (channels crossing ridges). These features can then be visualized on a map. Kwan & Lee (2004) build density surfaces of human activities. The base of a surface may be not only the geographical space but also an abstract two-dimensional space where one dimension is time and the other is person’s distance from home or another specific place. The surfaces corresponding to different activity types can be overlaid or juxtaposed for comparison.

What can be observed in these approaches to aggregating movement data is that the data are treated as a collection of independent discrete events. Each position record in movement data is treated as representing an event of presence of some entity in some position at some time. This view does not fully capture the essence of movement as continuous change of spatial position such that each position depends on the previous one. Hence, it cannot be sufficient for a comprehensive analysis of movement data, but may still be helpful for certain types of analysis (sub)tasks.

Another way of aggregating movement data described in the literature is based on considering the data as a set of moves between predefined places (typically, the places are spatial compartments). Each move is treated as a vector characterized by its origin and destination (i.e. the places where it starts and ends), by the start and end times, and, possibly, by additional attributes such as duration and length (travelled distance). Moves with coinciding origins and destinations are united into aggregate moves, which are characterized by the number of the original moves and by other derived attributes such as minimal, maximal, and average duration and length. The results may be visualized as an origin-destination matrix where the rows and columns correspond to the places and symbols in the cells or cell colouring or shading encode the derived attribute values (Guo et al. 2006, Guo 2007). A disadvantage of such visualization is the lack of spatial context. Aggregate moves can also be visualized on a map by bands or arrows connecting pairs of
locations (Tobler 1987, 2005). The widths of the bands or arrows are proportional to the volumes moved between these locations. Unfortunately, such a map may be illegible because of intersecting and overlapping symbols. Therefore, Tobler suggests a specific method for spatial smoothing of aggregate moves and generation of continuous flow maps, where vector fields or streamlines portray continuous flow patterns. However, this method is based on the assumption that some of the locations attract the moving entities, some others repulse, and the flows are directed from the latter to the former. This assumption restricts the approach to certain types of movement such as people migration. It is hardly applicable, for instance, to city traffic where vehicles and pedestrians are moving in all possible directions and locations attracting or repulsing significant proportions of the moving people occur very rarely.

Both origin-destination matrix and flow map ignore the temporal aspect of the movement data. An obvious extension is to divide the time span of the data into intervals and to build a flow map and/or matrix for each interval. The result may be an animated display where each state corresponds to one interval or a series of displays shown simultaneously (the technique is known as “small multiples”).

**Combining visualisation with computational analysis**

By the beginning of the GeoPKDD projects, there were only a few works reported in the literature on combining visual and computational techniques for analysis of movement data. Some of them have been already mentioned: building surfaces and applying surface topology analysis techniques (Mountain 2005) and computation of flow patterns for continuous flow maps (Tobler 1987, 2005). Buliung & Kanaroglou (2004) use computational methods of ArcGIS to build a convex hull containing all trajectories, compute the central tendency and dispersion of the paths, and represent the results on a map as the averaged path of all entities. Such geometric summarization can work well only when the trajectories are similar in shape and close in space.

Huisman and Forer (1998) compute and visualise space-time prisms, which show where and how long an individual might be between two known positions in space and time. Prisms are visualised as volumes in a space-time cube. Combining the prisms of several individuals uncovers the opportunities for possible meetings between them. The results of the combination are represented as “interaction surfaces”: a surface corresponds to some time interval and shows the number of person-minutes that can be spent in each place (grid cell) during this interval.

Combining visualisation with data mining is reported in the work of Laube et al. (2005), who analyse movements of football players during a game. The approach is to divide the whole time into short intervals and encode the movements of the entities (players) on these intervals by symbols representing the movement directions or other movement characteristics. Data mining methods search through the resulting symbol sequences for certain specific types of collective movement patterns such as synchronous movement and “trend setting”, i.e. movements of some entity being repeated by other entities after a time lag. The results are visualised in two ways: in a matrix where the columns correspond to the time intervals, the rows to the entities, and the colours of the cells represent the movements, and on a sort of map showing the fragments of the trajectories making a pattern.
Achievements of GeoPKDD in visualisation and visual analytics

The work in GeoPKDD concerning visualisation and visual analytics has been done along the following main lines:

– Developing a theoretical basis suitable for systemising current and potential approaches and guiding the design of new methods.
– Designing scalable visual analytics methods applicable to very large datasets.
– Designing innovative visualisation and interaction techniques.
– Empirical studies of different types of movement data and associated analysis tasks, investigation of the applicability of existing methods to different data and tasks, identifying needs in new methods.

Theoretical basis

One of the stimuli for developing a theoretical basis was to understand when the existing approaches (e.g. aggregation of movement data as independent discrete events) are applicable and effective. In (Andrienko & Andrienko 2007, Andrienko et al. 2008b) we introduced a formal model of collective movement of multiple entities as a function \( \mu: E \times T \rightarrow S \) where \( E \) is the set of moving entities, \( T \) (time) is the continuous set of time moments and \( S \) (space) is the set of all possible positions. Another representation is \( \mu(e,t)=s, e \in E, t \in T, s \in S \), which shows that \( e \) and \( t \) are independent variables and \( s \) is a dependent variable. The function \( \mu \) can be extended to include also various movement attributes (speed, direction, etc.) and/or movement-related attributes of the entities such as heartbeat: \( \mu: E \times T \rightarrow S \times A_1 \times A_2 \times \ldots \times A_N \) or \( \mu(e,t)=(s, a_1, a_2, \ldots, a_N) \). These additional attributes are dependent variables, analogously to the space.

As it is argued in (Andrienko & Andrienko 2006), in order to analyse data having two or more independent components, one may need to decompose the original function of several variables into multiple single-variable functions (of course, this should not be understood literally but as a metaphor for what is done in practice). For a function of two independent variables, there are always two possible decompositions.

Hence, as a function of two independent variables, \( \mu \) can be decomposed in two complementary ways:

- \( \{\mu_e: T \rightarrow S \mid e \in E\} \), where each function \( \mu_e: T \rightarrow S \) describes the movement of a single entity. We shall call the function \( \mu_e \) the trajectory of the entity \( e \). The decomposition of \( \mu \) into a set of \( \mu_e \) may thus be called trajectory-oriented view.
- \( \{\mu_t: E \rightarrow S \mid t \in T\} \), where each function \( \mu_t: E \rightarrow S \) describes the situation at a time moment \( t \), consisting of the spatial positions and, possibly, additional attributes of all entities. The decomposition of \( \mu \) into a set of \( \mu_t \) may be called situation-oriented view.

Note that the set of \( \mu_t \) is ordered according to the time. Theoretically speaking, it is continuous: a certain situation exists, in principle, for any element of the continuous set \( T \). In practice, however, only a finite set of different situations can be retrieved from the data and explored.
Figure 1 gives a graphical illustration of the two possible views of the movement of multiple entities.

Each of the views permits a different kind of analysis. However, there is an orthogonal dimension that further differentiates the possible analyses: absolute versus relative view of space (Peuquet 1994, 2002). According to the absolute view, space is an independently existing container where the entities are placed. According to the relative view, space is a positional attribute attached to the entities.

The absolute view focuses on space as the subject matter. Movement that occurs in space is thus considered as a property of the space. Respective analysis tasks are, for example, studies of the use of space, its accessibility and permeability (the ease or difficulty of moving in different directions), connectivity between different parts, etc. Such analysis tasks may be called space-centred. The identities of the moving entities are irrelevant for space-centred analyses and may be ignored. This means, in particular, that the data may be aggregated in such a way that multiple entities are handled together as a unit. Another implication is that the positions of the entities at different time moments (together with other attached characteristics) may be treated as independent discrete events. Hence, the aggregation methods suitable for independent discrete events are applicable in this case.

The relative view, in contrast, focuses on the entities as the subject matter. Movement is considered as a property of the entities while space is treated as a collection of relationships between the entities. Respective analysis tasks may be called entity-centred. An example is the investigation of emerging spatial relationships in a population of animals: whether the animals tend to move in large or small groups, in pairs, or separately from others, whether the groups have leaders, whether they are arranged in a particular way (such as the V-shape of a flock of flying geese), etc. The identities of the moving entities are important in this case since it is necessary to trace the groupings and arrangements over time. Hence, any kind of aggregation where multiple entities are merged into a single unit would not be suitable.

Table 1 summarizes our reasoning concerning the possible views of movement, respective analysis tasks, and the suitability of current techniques.
<table>
<thead>
<tr>
<th><strong>View of movement</strong></th>
<th><strong>Situation-oriented</strong></th>
<th><strong>Trajectory-oriented</strong></th>
</tr>
</thead>
</table>
| **Class of tasks**   | **Tasks**: study of space use, accessibility, permeability  
                      | **Techniques**: spatial and temporal aggregation of movement data as independent discrete events, generation of surfaces + visualization on animated map and map series | **Tasks**: study of space connectivity  
                      | **Techniques**: origin-destination matrix, discrete flow map (as display animation or display series)  
                      | **Tasks**: study of movement flows, routes, use of pathways  
                      | **Techniques**: did not exist before GeoPKDD* |
| **Space-centred**   | **Tasks**: study of the spatial distributions of the entities (esp. in connection to their activities)  
                      | **Techniques**: aggregation by activity types in addition to spatial and temporal, generation of activity density surfaces | **Tasks**: study of the movement behaviour of a single entity; comparison of the movement behaviours of several/multiple entities  
                      | **Techniques**: space-time cube (not scalable) |
| **Entity-centred** | **Tasks**: study of emerging spatial relationships (grouping, separation, spatial arrangement, etc.)  
                      | **Techniques**: map animation and map series showing individual entities (not scalable) | **Tasks**: study of relative movements and interactions between entities  
                      | **Techniques**: combination of space-time prisms and generation of interaction surfaces; ad-hoc data mining + visualisation (addressing specific types of patterns) |

* The technique of continuous flow map (Tobler 1987, 2005) is limited to predefined areas and assumes coherent movement from repulsive to attractive areas.

As may be seen, this theoretical framework reveals the major deficiencies among the available methods:

- Scalable methods for studying movement flows, routes, and use of pathways;
- Scalable methods for studying the movement behaviour of a single entity (especially during a long time period) and comparison of behaviours of several or multiple entities;
- Scalable methods for the study of emerging spatial relationships.
Scalable visual analytics methods

Aggregation in a database

Although basic approaches to spatial, temporal, and attribute-based aggregation of movement data were described before the beginning of GeoPKDD, the existing implementations relied on in-memory data processing, and hence were not scalable to very large databases. In GeoPKDD, we have investigated the opportunities for the use of aggregation functions available in standard relational databases. In this case, only aggregated data are loaded in the main memory, which makes the approach principally applicable to databases of arbitrary size.

For the aggregation, the time is divided into intervals, and the space (i.e. the territory where the entities move) is divided into appropriate compartments. Then, various aggregates are computed for each pair of space compartment and time interval from the position records fitting in this compartment and this interval: number of different entities, number of visits, total time spent, statistics of the movement-related attributes (minimum, maximum, average, median, etc.). The resulting time series of values characterising the spatial compartments are loaded in the main memory and visualised by means of interactive maps, map series, time graphs and other visualisation techniques suitable for time series. In our prototype implementation, spatial compartments are defined by building a regular rectangular grid of a desired resolution. This is sufficient for our research purposes, while the extensibility of the approach to arbitrarily shaped areas, such as traffic analysis zones (used in transportation planning) or road segments, is obvious.

Besides the use of the database technology, we also suggest new techniques for the aggregation and subsequent visualisation of movement data. Thus, for exploring cyclical temporal patterns, we suggest two-dimensional temporal aggregation (additionally to the spatial aggregation), e.g. by days of the week and by times of the day, or by years and by months. An appropriate visual display for so aggregated data is a map with mosaic diagrams, as in Figure 2. For this map, the data about the movement of a large number of cars in a city have been aggregated by spatial compartments, days of the week, and hours of the day. Each mosaic diagram on the map summarises the daily and weekly patterns of the traffic in the corresponding compartment. The columns of the diagrams correspond to the days of the week and the rows to the hourly intervals of the day. The colours of the “tiles” of the “mosaics” encode the values of the median speed in the respective days of the week and hours of the day. Green corresponds to fast speeds (the faster, the darker) and red to slow speeds (the slower, the darker; in particular, dark red represents speeds below 15km/h). It may be seen that the speeds are always very low in the inner city except for the north-western part. The diagrams in the compartments on the western and eastern belt roads are mostly green but contain red or yellow spots indicating decreased speeds. Thus, in the northern part of the eastern belt road the red spots in the diagram indicate very low speeds in the morning hours of the work days. In some compartments on the northwest the speeds are also low in the mornings of the work days but also in the midday and afternoon on Wednesday and Thursday. On the northern belt road, especially the eastern part of it, the situation appears quite bad. In some compartments, the speeds are very low during the whole day and increase only in the night.
Another innovative aggregation and visualisation technique we have developed in GeoPKDD takes into account the directions of the movement. The data are aggregated not only by space and time but also by the direction (course). Movement directions are often indicated in the original track records. If this is not the case, they can be computed from pairs of consecutive positions of the same entity. The directions are typically specified by numeric values representing angular degrees from 0 to 359. We divide this range into intervals corresponding either to four main compass directions (north, east, south, and west) or to four main and four intermediate directions. Position records fitting in the same spatial compartment and temporal partition are additionally grouped by the movement directions. A separate group is made from records where the speed is below a chosen threshold. This is treated as the absence of movement (in real data consecutive position measurements are never exactly the same even when the object does not move, which means that the speed values of non-moving objects may differ from zero). Then, various counts and statistics of attribute values are computed for the groups.

To visualize the resulting aggregate data, we suggest a special technique in which the data are represented on a map by directional bar diagrams. Analogously to the wind rose used in meteorology, the bars are oriented in four or eight compass directions and their lengths are proportional to the values of the currently selected summary attribute corresponding to the respective directions.
Figure 3. The directional bar diagrams show movement data aggregated by compass directions. The lengths of the bars are proportional to the numbers of the cars that moved in the respective directions during a selected time interval. The radii of the circles are proportional to the numbers of the cars with the speeds below a selected threshold (here 5km/h). On the right, only dominant directions are shown, specifically, where the values are at least 25% (an interactively selected threshold) higher than the next highest value.

Thus, the diagrams in Figure 3 (left) portray the numbers of the cars that moved in different directions on Monday between 7 and 8 AM. The bars are coloured depending on their orientation; a particular colour is assigned to each direction. This helps in gaining an overall view of the prevailing movement directions throughout the whole territory. Besides the directional bars, some diagrams include grey circles representing the groups of records with the speeds below the chosen threshold. The radii of the circles are proportional to the values of the currently selected summary attribute computed for these groups of records. The radii can be easily compared with the lengths of the bars. In Figure 3 the circles represent the numbers of the distinct cars that had the speeds below the chosen threshold of 5km/h. Such speeds occur predominantly in the central part of the city but also on the northeast, where the circles located on a segment of a motorway may indicate its congestion.

Visual exploration of the movement directions with the use of this kind of display is supported by a number of interactive facilities:

- switch from one summary attribute to another, e.g. from the number of entities to the average or median speed;
- select another temporal partition, i.e. another interval, day of the week, time of the day, etc., depending on how the data have been aggregated;
– hide some directions in order to focus on the remaining direction(s), e.g. to see where northward movement occurs;
– choose the mode of presenting only the dominant direction(s) in each spatial compartment. A direction is treated as dominant when the corresponding value of the current summary attribute exceeds the highest value among the remaining directions by a chosen threshold, which may be either absolute (i.e. minimum difference between the values) or relative (i.e. minimum ratio).

The screenshot on the right of Figure 3 shows the dominant movement directions defined by the relative threshold of 25%. It may be seen that movements towards the centre prevail on most radial streets and that movements to the east (green bars) dominate on the motorway on the south. In some compartments there are two or more dominant directions. This means that the respective attribute values differ by less than 25%.

This aggregation and visualisation technique can be used for the exploration of space permeability or impedance to the movement. An example is shown in Figure 4. To explore the traffic on a particular road, only the space compartments (grid cells) covering this road have been selected. The data have been aggregated according to the four main compass directions. The bar diagrams represent the median speeds in the eastern (green) and western (purple) directions. The diagrams are substantially asymmetric, which means different speeds of the movement in the eastern and in the western directions. Lower speeds, in turn, signify higher obstruction to the movement.

*Figure 4.* The bars represent the median speeds of the movement toward the east (green) and west (purple) between 11 and 12 AM on Wednesday along a motorway on the north of Milan.

Our scalable aggregation and visualisation methods have been reported in (Andrienko and Andrienko 2008).

**Extraction and interpretation of significant places**

An important task in the analysis of individual movement behaviour is extraction of significant places of the moving agent. Thus, in case of data about a person, these are the places of home, work, shops, school(s) and/or kindergarten(s) attended by person’s child or children, homes of person’s friends and relatives, etc. Significance of a place is indicated by considerable amounts of time spent there and/or repeated visits to this place. Hence, in order to discover the significant places of some moving agent, one should
extract the stops, i.e. the time intervals when the agent did not move and the corresponding spatial positions. This can be done by means of database queries. Then spatial clustering can be applied to the extracted positions of the stops to find the places of repeated stops. To interpret the places, it is useful to take into account the typical times and durations of the stops occurring in these places.

For illustration, we use a dataset consisting of positions of a private car, which has been GPS-tracked during about a year. The data have been voluntarily given to us by the car owner. To discover and interpret the significant places of the person, we first extract the positions of the stops lasting 3 hours or more by means of a database query. Then we apply the spatial clustering tool to the extracted positions, which produces two major clusters. We visualise the distribution of the stop times over the days of a week and the hours of a day by means of segmented time histograms with the segments corresponding to the clusters (Figure 5).

![Figure 5. The temporal histograms show the weekly (A) and daily (B) distributions of the stops of the personal car with the duration of 3 hours or more. The map on the right shows the spatial positions of the clusters. The dark grey dots are the positions of occasional stops, which do not make spatial clusters.](image)

In Figure 5A we can see that the stops of cluster 1 (red) occur on all days of the week and the stops of cluster 2 (blue) occur from day 1 to day 5, i.e. from Monday to Friday. Figure 5B shows us that the stops of cluster 1 occur mostly in the second half of the day; the maximum occurrences are from 19 to 20 o’clock. The stops of cluster 2 occur mostly in the morning hours. Such a distribution makes us quite confident that cluster 1 is located near person’s home and cluster 2 is near person’s work. In a similar way, we extract, analyse, and interpret the places of shorter visits (Andrienko et al. 2007). In particular, to find the places of person's shopping, we apply interactive filtering to consider separately the times of visits in the working days and on the weekends.

**Progressive clustering of trajectories**

When it comes to exploring and analyzing large amounts of data, clustering is one of the general approaches since it allows an analyst to consider groups of objects rather than individual objects, which are too numerous. It should be noted that clustering is not a
standalone method of analysis whose outcomes can be immediately used for whatever purposes (e.g., decision making). An essential part of the analysis is interpretation of the clusters by a human analyst; only in this way they acquire meaning and value. To enable the interpretation, the results of clustering need to be appropriately presented to the analyst. Visual and interactive techniques play here a key role.

Trajectories are complex spatio-temporal constructs, which require sophisticated clustering methods. There are two main approaches to clustering complex data: (i) defining ad hoc notions of clustering and clustering algorithms tailored to the specific data type; and (ii) applying generic notions of clustering and generic clustering algorithms by defining some distance function, which measures the similarity between data items. In the second case, the specifics of the data are completely encapsulated in the distance function. In our research, we pursued the second approach.

Trajectories are characterised by a number of non-trivial and heterogeneous properties including the geometric shape of the path, its position in space, the life span, and the dynamics, i.e., the way in which the spatial location, speed, direction and other point-related attributes of the movement change over time. Because of this complexity, trajectories cannot be adequately represented as points in a multi-dimensional feature space so that all features could be treated uniformly. Each property must be treated in its own way. However, creating a single distance function properly accounting for each property is hardly a reasonable endeavour irrespective of the effort this would require. On the one hand, not all characteristics of trajectories may be simultaneously relevant in practical analysis tasks. On the other hand, clusters produced by means of such a universal function would be very difficult to interpret. A more reasonable approach is to give the analyst a set of relatively simple distance functions dealing with different properties of trajectories and provide the possibility to combine them in the process of analysis.

We suggest and instrumentally support a step-wise analytical procedure called “progressive clustering”. The main idea is that a simple distance function with a clear meaning and principle of work can be applied on each step, which leads to easily interpretable outcomes. However, successive application of several different functions enables sophisticated analyses through gradual refinement of earlier obtained results. Visualization and interaction techniques play here a crucial role. We illustrate the concept by example of the data about the movement of multiple cars in a big city.

We take a subset of car trajectories made in the morning of a working day and apply clustering with the distance function “common destination”. This is a very simple function that computes the spatial distance between the ending positions of the trajectories. Despite the simplicity, the function is very useful as it allows us to discover the most popular destination areas in the city. The biggest clusters of trajectories by common destination are shown in Figure 6.
At the next step, we select only the trajectories of cluster 3, which end in the centre of the city, and apply the clustering tool with the distance function “route similarity” (Andrienko et al. 2007). This is a quite complex function, which repeatedly searches for the closest pair of positions in two trajectories. It computes the mean distance between the correspondent positions along with a penalty distance for the unmatched positions and combines the two distances at the end. The function has been designed to tolerate incomplete trajectories (i.e. where some parts in the beginning and/or in the end are missing) and positioning errors. It properly handles unequal time intervals between records. The application of the clustering tool with this distance function to the cluster of trajectories ending in the centre produces a number of clusters (the biggest are presented in Figure 7), which give us the idea about the popular (frequently occurring) routes towards the centre. In a similar way, we can explore the other clusters that have been obtained at the first step.
Figure 7. Clustering with the distance function “route similarity” has been applied to the cluster of trajectories ending in the city centre (cluster 3 in Figure 5). The screenshot demonstrates the biggest 6 clusters.

In (Andrienko et al. 2007) we demonstrate that progressive clustering may also be useful in analysing individual movement behaviour.

Besides the advantages from the sense-making perspective, progressive clustering provides a convenient mechanism for user control over the work of the computational tools as the user can selectively direct the computational power to potentially interesting portions of data instead of processing all data in a uniform way. Details about the progressive clustering procedure and a discussion of its strengths and weaknesses are published in (Rinzivillo et al. 2008). One of the weaknesses is that the clustering algorithm works with data loaded in the computer main memory; hence, the procedure is not sufficiently scalable. However, we have recently found a way to do cluster analysis of very large sets of complex data, in particular, trajectories.

**Interactive visual clustering of large collections of trajectories**

We propose an approach to extracting meaningful clusters from large databases by combining clustering and classification, which are driven by a human analyst through an interactive visual interface. The essence of the approach can be shortly described as
follows. First, the analyst takes a manageable subset of the objects and applies clustering to it. The analyst experiments with the clustering parameters for gaining meaningful results with respect to the analysis goals. Then, the analyst builds a classifier, which can be used for attaching new objects to the existing clusters. The analyst may also modify the clusters for their better understandability and/or conformance to the goals. The produced classifier is applied to the whole dataset. For this purpose, the data are loaded from the database by portions that can fit in the main memory. Each object is either attached to one of the clusters or remains unclassified, if it does not fit in any cluster. The result of the assignment is stored in a database table, and the object is discarded from the main memory. In this way, the whole dataset is processed. When necessary, the analyst may repeat the procedure (take a subset – cluster – build a classifier – classify) to the unclassified objects.

A classifier consists of the distance function that has been used for the initial clustering, a number of cluster prototypes, and the corresponding distance thresholds. Cluster prototypes are selected objects from a cluster such that any member of the cluster is close to at least one prototype, which means that the distance to this prototype computed by the distance function is less than the distance threshold of this prototype. The classification is done by comparing each object from the database to the cluster prototypes, i.e. computing the distances by means of the distance function. An object is attached to a cluster if its distance to one of the prototypes is below the respective threshold. If an object is close to prototypes of two or more clusters, the closest prototype is chosen. If an object is not sufficiently close to any of the prototypes, it remains unclassified.

We have developed a tool which automatically builds an initial version of classifier according to results of clustering and allows the user to inspect and modify it by means of interactive visual techniques. There are at least two motives for revising a classifier. First, density-based clusters sometimes have high internal variation and are difficult to understand. The analyst may wish to refine them by dividing into parts with smaller internal variation and/or by removing some of the members. Second, the analyst may wish to tune the selection of cluster prototypes and distance thresholds to his/her understanding of the distinctive properties of the clusters. After the classifier is inspected and, if necessary, edited, the analyst can apply it to the whole database.

Let us illustrate the idea using the clusters of trajectories to the city centre we have previously obtained at the second step of the progressive clustering (see the previous section and Figure 7). Figure 8 presents the six biggest clusters with their prototypes. The ordinary cluster members are shown in neutral dark grey colour using thin lines. The cluster prototypes are shown by thick lines and painted in the colours assigned to the clusters. As cluster 1 (top left in Figure 7) and cluster 4 (top middle in Figure 7) had originally quite high internal variation, we have interactively refined them. We have split cluster 1 into 3 parts corresponding to slightly different routes, as is illustrated in Figure 9. The first part remains as cluster 1 (top left in Figure 8), while the other two parts make two additional clusters. Cluster 4 has been refined by removing some trajectories deemed insufficiently consistent with the bulk of the cluster.
Figure 8. The prototypes of the six biggest clusters of car trajectories to the city centre presented in Figure 7.

Figure 9. Cluster 1 visible in Figure 7 has been refined by extracting two parts corresponding to slightly different routes. The main part of cluster 1 may be seen in Figure 8 (top left).

We apply the refined classifier to the whole set of trajectories. The classification process takes 11.4 minutes for about 176,000 trajectories. As a result, 3430 trajectories have been attached to the 11 clusters represented by the classifier; the rest has not been classified. Figure 10 demonstrates 6 biggest clusters extracted from the database. The sizes of the clusters are specified in parentheses above the images of the clusters.
Figure 10. The clusters extracted from the whole database by means of the classifier (6 biggest clusters are shown).

Figure 11. 3 clusters corresponding to the original cluster 1 (Figure 7 top left), which has been refined (Figure 9).
Figure 11 presents the three clusters corresponding to our refinement of cluster 1. The clusters are shown on the same map for easier comparison. The trajectories are drawn with 20% opacity. The “red” and “violet” clusters can be also seen separately in Figure 10. The “pink” cluster is much smaller; it has only 51 members.

This example illustrates our approach. A comprehensive description and discussion will be published soon. Essentially, the approach is generic, i.e. applicable to different types of structurally complex objects. A key feature is the division of labour between computer and human and a true synergy where each side helps the other. Thus, existing clustering algorithms are usually not scalable to very large datasets not fitting in the computer main memory. Our visual analytics approach solves this problem at the cost of involving a human analyst, who directs the work of the computer towards the discovery of meaningful, relevant clusters.

**Spatial summarisation of trajectories**

Representation of trajectories in a generalised and summarised form is necessary for reducing display clutter and increasing legibility. Another function of generalisation and summarisation is promoting abstraction and distilling of key features out of low-detail noise. Clustering of trajectories may be the first step towards generalisation as it allows the analyst to consider groups (clusters) of similar trajectories instead of individual trajectories. The second step would be to represent each cluster in a summarised way so that it is easy to understand what is common among the cluster members and also to estimate the amount of internal variation.

In looking for a method to summarise trajectories, it is necessary to take into account the following consideration. Trajectories are usually not disjoint in space; they intersect and overlap. As a consequence, summarized representations of groups of trajectories would also intersect and overlap when drawn on the same map display, which can make the picture incomprehensible. A possible solution is to present an overview display of all clusters in a “small multiples” style, where each cluster is shown on a separate map. Since each of these maps has to be quite small, the clusters need to be represented in such a way that only the principal features of each cluster are visible, but these features are very easy to grasp. In the previous sections, we have used “small multiples” for the original trajectories without summarisation. The small images of the clusters are not always clear, and the key features of the clusters are hardly seen, especially when the clusters are big. Hence, the representation of the clusters in “small multiples” needs to be highly schematic and remain clear irrespective of the size of the cluster. Such a representation of a cluster may be called *graphical spatial model* (Cheylan et al. 1997, van Elzakker 2004 pp.65-71), or, shortly, *spatial summary*.

On the other hand, the analyst may wish to examine selected clusters in more detail. For this purpose, the level of generalisation and summarisation should be lower. Hence, we need a parametric method for building spatial summaries of clusters of similar trajectories so that the degree of schematisation and the level of detail of a summary is adjustable to the available display size and to the user’s needs.

A possible way to obtain spatial summaries of clusters of trajectories is to summarise trajectories into aggregate moves between appropriately defined areas (Andrienko &
Andrienko 2008). An aggregate move between two areas summarises a set of fragments of trajectories starting in the first area and ending in the second area. An aggregate move is represented on a map by an arrow with the thickness proportional to the number of trajectory fragments summarised in the move. The main problem here is the generation of appropriate areas for building aggregate moves between them. We have found an approach that works sufficiently well for diverse examples of movement data (but needs further elaboration).

Our method, first, extracts characteristic points from the available trajectories, which include the start and end points, the points of turns and the points of stops. For determining the turns and stops, the method requires two user-specified thresholds: the minimum angle of a turn and the minimum duration of a stop, i.e. keeping the same position. Then, the extracted points are clustered by spatial proximity. Here, another user-specified parameter is used: the maximum radius of a point cluster (i.e. of the circle enclosing all its points). This parameter determines the degree of generalisation. Finally, the centres of the clusters are used for partitioning the territory by Voronoi (Thiessen) polygons. The polygons are then used as the areas for the summarisation. Owing to the use of the characteristic points of the trajectories for defining the Voronoi cells, the following generalisation of the trajectories to sequences of visited cells preserves quite well the mainstream shapes. We are now developing a method to measure the quality of the generalisation, i.e. how much the original shapes of the trajectories are distorted.

Figure 12 demonstrates the spatial summaries of the six clusters shown in Figure 10. The polygons that have been used for building these summaries are not shown: they play an auxiliary role and can be omitted after the summarisation is done. The route followed by each group of trajectories is clearly seen from this representation. It should be noted that the tool producing the images filters the aggregate moves in each spatial summary for a better clarity: the moves corresponding to a small proportion of trajectories are omitted. However, the user can also view the spatial summaries of selected clusters without filtering. Thus, Figure 13 left demonstrates the spatial summaries of three selected clusters with all their aggregate moves visible. On the right, the clusters are shown without summarisation for comparison (these are the same clusters as in Figure 11). It may be seen that the spatial summaries exhibit quite well the major movement directions and the frequently occurring route segments. At the same time, they give us an idea about the internal variation: where deviations from the major course occur, how far, and how many trajectories are involved.
As already mentioned, the degree of generalisation can be regulated by the parameter specifying the maximum radius for the clusters of characteristic points. As the centres of
the clusters are used as generating points for Voronoi cells, the maximum radius affects the sizes of the cells, which, in turn, determine the degree of generalisation: the larger the cells, the coarser the spatial summaries. The effect of the parameter on the degree of generalisation is illustrated in Figure 14, where three spatial summaries of the same cluster (cluster 1) are built with the maximum radii 3000m (A), 2000m (B), and 1000m (C). The latter summary appears somewhat cluttered. For a better understanding of the features of the cluster, it makes sense to apply filtering, as in Figure 14 D. Our tools support interactive dynamic filtering: the image changes as the user moves the slider defining the minimum (or maximum) number of trajectory segments summarised in an aggregate move. Only the moves satisfying the filter remain visible.

Figure 14. Spatial summaries of the same cluster (cluster 1) differing by the level of abstraction. A) Radius = 3000m. B) Radius = 2000m. C) Radius = 1000m. D) Same as C, but the minor moves have been omitted to reveal the major flow.

A good property of the spatial summaries is their scalability: a spatial summary can effectively represent a group of ten, hundred, thousand, or million trajectories since the representation does not depend on the absolute numbers but only on the proportions. It may be noted that the way in which we build the summaries is not scalable. Indeed, at the first stages characteristic points are extracted from the trajectories and then clustered.
The time for the extraction of the characteristic points depends linearly on the number of trajectories, and the time for the clustering depends on the number of extracted points as $O(n \log n)$. However, building a mesh of Voronoi cells does not essentially require all characteristic points from all trajectories. In fact, we need just one generating point for each cell. We use the centres of the clusters of characteristic points for achieving a good correspondence of a spatial summary to the core spatial features (in particular, mainstream shapes) in a group of trajectories. This effect can be achieved without processing all trajectories of the group. It is sufficient to have a subset of prototypical trajectories representing the main features of the group members.

We have described in the previous section how cluster prototypes can be used for cluster-based classification of a large dataset. The same prototypes can also be used for producing Voronoi tessellation of the territory. When the partitioning is defined, the following building of spatial summaries can be done in an incremental mode during the classification process: when a trajectory is attached to a cluster, the spatial summary of this cluster is updated. This technique is scalable as the computation time is linear with respect to the number of trajectories. Examples of results are shown in Figure 15.

![Spatial summaries of clusters of trajectories built incrementally during the process of classification.](image)

These are the summaries of the same clusters as in Figure 12. The Voronoi cells for the summaries in Figure 12 have been built using all trajectories of the big clusters, i.e. the clusters extracted from the whole database. The summaries in Figure 15 are based on
Voronoi cells generated using only the prototypes of the initial small clusters (Figure 8) detected in a subset of trajectories. A consistency between the summaries is notable. The existing differences are minor.

A tricky issue in partitioning the territory on the basis of characteristic points from cluster prototypes is the possible existence of large parts of the territory where there are no points. If these parts remain undivided, trajectories deviating from the cluster prototypes will be disproportionately distorted. To decrease the distortion, we introduce additional generating points for Voronoi cells. For this purpose, we produce a set of points regularly distributed over the territory, but a point from this set is added to the generating points for the Voronoi tessellation only if it is sufficiently far from the available points (the distance to the nearest point exceeds the doubled maximum radius).

In the nearest future, we plan to investigate more in depth the properties of the spatial summarisation, develop a method for measuring its quality, and extend the approach to spatio-temporal summarisation. Spatio-temporal summaries will be visualised in a three-dimensional display (space-time cube), where two dimensions represent the space and the third dimension represents the time.

Innovative visualisation and interaction techniques

Dynamic aggregation

In the previous section, we have described several methods for scalable aggregation and summarisation of movement data, which work without loading all data in the main memory. However, aggregation and summarisation are useful also in a case when the amount of data permits loading in the main memory since they reduce the display clutter and promote abstraction. When the data are in the main memory, it is possible to do aggregation and summarisation in a dynamic way, as described below.

A **dynamic aggregator** is a special kind of object which is linked to several other objects (members of an aggregate) and computes certain summary attributes such as the member count and the minimum, maximum, average, median, etc. from the attribute values of the members. The aggregator reacts to various kinds of interactive filtering applied to its members by adjusting the values of the summary attributes: only the members that pass through the current filter are taken into account in computing the values.

In our experimental software, we have two types of dynamic aggregators of movement data: **aggregate moves** and **summation places**. Aggregate moves have been introduced in the previous section. A **dynamic aggregate move** is defined by two places (areas) A and B and is “aware” of all trajectories that visit A and B in this specific order without passing any intermediate place. The members of the aggregate move are the respective trajectory fragments. An aggregate move produces the count of its members and statistical summaries of the lengths of the fragments, their durations, and speeds. A **summation place** is an area in space which is “aware” of all trajectories passing through it and stores the positions and times of entering and leaving the area. The fragments of the trajectories lying inside the area are the members of the place. A summation place counts its members and computes such summary attributes as the minimum, maximum, average, median, and total time spent in the area, statistics of the speeds, etc., as well as the
number of starts (first points of trajectories) and the number of ends (last points). Summation places are automatically created from the places where aggregate moves originate or end. It is also possible to create other summation places, for example, from cells of a regular grid.

Aggregate moves are visually represented on a map by arrows, as shown in the figures in the previous section. They can also be represented by symbols in cells of an origin-destination matrix. Summation places may optionally be visualised on a map by drawing the outlines of the areas and/or filling the interiors. Irrespective of whether the places themselves are shown or not, current values of selected summary attributes may be visualised on a map by graduated symbols or by diagrams. It is also possible to use non-cartographic displays such as scatterplot, parallel coordinates plot, or frequency histogram.

The interactive filters, which make dynamic aggregates re-compute the values of the summary attributes, include temporal filter (selection of a time interval), spatial filter (selection of a “window” in space), attribute filter (selection of trajectories by their attributes such as duration and length), and cluster filter (selection of clusters). In re-computing, the aggregates take into account only the active members, i.e. the members that have passed through all currently set filters. When an aggregate move has no active members, it does not appear on a map; otherwise, the thickness of the corresponding vector is adjusted to the current value of the represented summary attribute. The same happens to the symbols or diagrams representing the summary attributes attached to the summation places. In particular, when some cluster of trajectories is selected, the summation places and aggregate moves summarize only the trajectories belonging to this cluster.

There is an essential difference between the filtering of aggregate moves, which has been demonstrated in Figure 14 D, and the dynamic reaction of aggregate moves to filtering of trajectories. These are two independent mechanisms, which can be used separately or in combination. The filtering of aggregate moves can hide some of them from the view but does not change the values of their attributes. When filtering is applied to trajectories, the values of the attributes of some aggregate moves change. This may affect the visual appearance of the aggregate moves; in particular, some moves may become invisible because none of their members is active. If, additionally to this, some filtering is applied to the aggregate moves, it may happen that the new attribute values change the satisfaction of the filter conditions: a previously active move may turn into inactive and be hidden and vice versa.

Let us illustrate the technique of dynamic aggregation by example. When exploring the routes of the cars in a city, we have detected that people use quite different routes for getting from the north-east of the city to the city centre. Figure 16 shows 8 different routes in a summarised form and the frequencies of these routes. The “orange” and “light blue” route seem to be the most direct among all. One would expect these routes to be taken more frequently than the others. However, the “light blue” route is the least frequent among all, and the “orange” route is much less frequent than the “red” one. A possible reason is that the drivers prefer to use the roads where they can move with a higher speed. In particular, the “red” route goes along the northern belt road, where the allowed driving speed may be higher than on the smaller roads inside the inner city.
To see how the frequencies of using the different routes vary depending on the time of the day on working days, we apply interactive temporal filtering to the trajectories. First, we select only the trajectories that occurred on the working days, i.e. from Monday till Friday. Then, we filter the trajectories by the times of a day. Figure 17 shows the results of the filtering by 2-hour intervals. The trajectories themselves are not shown, only the aggregate moves, which react to the filtering of the trajectories. It may be seen that the relative frequencies of the different routes are not the same in different hours of the day.
A possible reason for choosing one or another route may be the traffic situations on different roads: if the movement on some road is obstructed, a driver may prefer to use another road. In order to see whether the choice among the different routes may be related to obstructed traffic, we generate summation places as Voronoi polygons using the same method as in the spatial summarisation of trajectories. For each summation place, the system automatically computes a number of attributes, including the mean and median speed of the movement inside the area. These attributes are dynamically recomputed when the results of filtering of the trajectories change.

Figure 18. The bar diagrams portray the mean (yellow) and median (dark blue) speeds of the movement in different places along the routes.

Figure 18. In Figure 18, we have visualised the mean and median speeds of movement in the summation places by bar diagrams. The yellow bars correspond to the mean speeds, and the dark blue bars to the median speeds. The eight screenshots correspond to the consecutive 2-hour intervals of a day starting from 05-07h and ending with 19-21h. It may be seen that the speeds along the routes vary over the day; however, the mean and median speeds in the place where the “orange” route turns to the south from the northern belt road are always low. This may indicate obstructed traffic on the exit from the belt road. This, in turn, may be the main reason for the less frequent use of the “orange” route as compared to the “red” route. Note that the speeds along the “orange” route after passing this problematic road exit are usually higher than along the “red” route, except
for the time interval 05-07h. In this interval, as well as in the interval 09-11h, we see an increase of the relative frequencies of the routes using the eastern belt road, where the speeds are higher.

This example demonstrates a possible use of dynamic aggregation for studying the variation of collective movement patterns over time and their dependency on the movement conditions. Generally, the range of applicability of dynamic aggregation in terms of possible analysis tasks is rather wide. The limitation with regard to the size of data is posed by the in-memory processing; however, it may be expected that the progress of the computer hardware and database technologies will soon permit dynamic filtering and dynamic aggregation of large amounts of data without loading all data in the main memory.

**Analysis of interactions between moving agents**

In analysing movements of related entities, the analyst may be interested in uncovering the interactions between the entities in the process of their movement. Movement data usually consist of time-stamped position records and do not contain any explicit information about interactions; hence, it is only possible to detect *indications* of possible interactions. An important indication is *spatial proximity* between two or more objects at some time moment or during a time interval. The notion of spatial proximity depends on a number of factors; some of them are listed in Table 2.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of movement</td>
<td>walking, cycling, driving, …</td>
</tr>
<tr>
<td>Type of relation in focus (analysis task)</td>
<td>possibility to observe, possibility to talk, possibility to touch, …</td>
</tr>
<tr>
<td>Place</td>
<td>city centre, shopping mall, nature park, highway, …</td>
</tr>
<tr>
<td>Time</td>
<td>early morning, rush hours, late evening, night, …</td>
</tr>
</tbody>
</table>

An example dataset requiring the analysis of possible interactions between moving agents was collected by tracking movements of 303 schoolchildren while they were playing an outdoor mobile game. According to the rules of the game, the children were supposed to visit various places in a city and answer place-related riddles. The players were organised in competing teams. The goals of the analysis are to find out whether the players cooperated within the teams and whether there were conflicts between members of different teams. Detecting and examining indications of possible interactions between the players may help answer these questions.

In case of a large dataset, possible interactions need to be extracted from the data by means of computational techniques. We have developed a simple and fast computational method for extracting possible interactions from movement data. The user is expected to specify threshold values for the spatial and temporal distances between positions of two objects. The method first searches for pairwise interactions. For each pair of objects, it tries to find respective positions in their trajectories such that the spatial and temporal distances between them are within the given thresholds. In case of detecting such
positions, the following positions of the trajectories are checked. After extracting pairwise interactions, the method combines interactions sharing a fragment of a trajectory.

After the possible interactions have been extracted, they need to be visualized for enabling a human analyst to explore and interpret them. By this moment, we have developed techniques and tools to support the *elementary level* of analysis (Bertin 1983), when the analyst considers particular instances of interaction between individual agents. For this purpose, the system creates a map layer with a special kind of geographical objects representing the interactions and a table with information about the interactions: the start and end times, duration, number of interacting agents, and their identifiers. Instances of interactions are visualized on a map as is shown in Figure 19.

![Figure 19. Examples of interactions represented on a map.](image)

The representation includes the “footprints” of the moving agents, i.e. the fragments of their trajectories made during the interaction. The smaller hollow squares mark the beginnings of the trajectory fragments and the bigger filled squares mark their ends. The corresponding (i.e. close in space and time) points of the trajectories are connected by lines. If only one trajectory point is close in space and time to one or more points from another trajectory, this point is marked by a small hollow circle instead of representing a trajectory fragment. Interactions may be surrounded by bounding rectangles for better visibility. In Figure 19, the footprints of the interacting players are coloured according to their teams. On the right, two interactions between the same two persons are shown together with their trajectory lines.

Details about an interaction can be accessed by selecting it on a map with the mouse. A special popup window (Figure 20) will inform the user about the time interval of the interaction, the identifiers and names (if available) of the objects, and the start and end times of their participation in the interaction. Additionally, any attributes of the trajectories can be chosen for viewing. If two or more interactions exist at the cursor position, the data about all these interactions are displayed.
Figure 20. A popup window with information about two selected interactions, which occurred in the same place.

The map representation of the interactions does not convey the times of the interactions and the order in which they occur. To represent the temporal dimension, we have designed a special timeline display. An example is shown in Figure 21.

Figure 21. A timeline display of the trajectories and interactions.

Here, the horizontal dimension represents the time. The coloured lines represent the trajectories; the lines are positioned according to life times of the trajectories. The lifelines are coloured according to the teams of the moving agents. The trajectories have been filtered so that only the trajectories of the agents participating in at least one interaction are visible. The yellow segments inside the lifelines represent the interactions.
For each interaction, there are segments in the lifelines of all trajectories involved in it. Putting the mouse cursor on a marker of an interaction displays a popup window with details about the interaction. In the display fragment in Figure 21, temporal zooming is applied: the horizontal dimension represents a selected time interval.

While the map is good for conveying the spatial positions of the interactions and the timeline display is effective in conveying their temporal positions and durations, they do not adequately support the interpretation of the behaviours of the interacting agents. For the latter purpose, a display representing both the spatial and the temporal aspects of the interactions is required. We visualise interactions in a space-time cube by line segments connecting corresponding points of trajectories. When the corresponding points are very close, the lines may be hardly visible. In such a case, increasing the thickness of the lines helps. In Figure 22, the space-time cube display on the left shows several interactions between two players from different teams, which occurred in about the same place shortly one after another (the current state of the cube represents the time interval of 10 minutes length). On the right, the same data are visualised on a map. It appears that these two representations are complementary and that both are required for understanding the behaviours of the players.

Figure 22. Left: interactions between two players are marked in a space-time cube. Right: the same interactions as they appear on a map.

Besides creating a new map layer and a corresponding table, the system adds several new attributes to the table describing the trajectories: the number of interactions involving each moving agent, the number of other agents with which the agent interacted, and the identifiers of the trajectories of those agents. Figure 23 left shows a fragment of a table display presenting the values of these attributes. On the right, there is a summary display showing the frequencies of the numbers of interactions per player. We can learn that 23 players had two or more interactions; however, in all but one cases these were interactions with the same participants, like in the case illustrated in Figure 22. Only the player with the identifier 176 interacted with two other players.
Figure 23. Left: information about the interactions in the table describing the trajectories. Right: the frequencies of the numbers of interactions per person.

Figure 24. Filtering helps the analyst to focus on the interactions of a particular agent. To be able to consider particular interactions, the analyst needs flexible and convenient tools for interactive filtering. Our visual analytics toolkit supports the following types of filtering:
- Temporal filter: the user selects a time interval; all displays show only the interactions and trajectory fragments that occurred within this interval.
– Spatial window: the user specifies a rectangular region on a map; all displays show only the interactions which are located within this region.

– Attribute filter: the user can specify constraints on values of one or more attributes of the interactions, for example, duration and/or number of objects. Only the interactions satisfying these constraints will be visible.

– Direct selection of objects (in particular, trajectories or interactions) for viewing.

– Specially designed “aggregate filter”, which allows the user to find interactions involving specific objects.

Thus, in Figure 24 demonstrates a joint effect of two filters. By means of the aggregate filter, we have selected the interactions of the player 176 (the only one who interacted with two other players) and the trajectories of the players involved in these interactions, including player 176. By means of the temporal filter, we have restricted the view to a 30-minutes interval in which the interactions have occurred. The interactions are marked in the space-time cube by thick yellow line segments (we use yellow this time for a better contrast with the colours of the trajectories).

More details about the methods for the extraction, visualisation, and interactive examination of possible interactions between moving agents are available in (Andrienko et al. 2008c). However, this work should be considered as preliminary. A major problem is that inspecting and interpreting instances of interactions is a time-consuming activity, which cannot be done for a large number of interactions. Hence, there is a need in automated classification of interactions according to their essential properties. For this purpose, it is necessary to define the essential properties of interactions and the ways of extracting these properties from movement data. This is a topic of our further research. We intend to develop a visual analytics method where the analyst interactively defines the types of interactions he/she is interested in (by example or by specifying relevant properties) and the computer finds interactions of these types in the data.

**Summary**

In GeoPKDD, we have made substantial progress in developing visual analytics methods for movement data. A fundamental characteristic of these methods is their scalability with respect to the size of the data. Some of the methods are applicable even to data not fitting in the computer main memory. These include the techniques for database aggregation, cluster-based classification, and incremental summarisation of trajectories. Such methods did not exist before GeoPKDD. The remaining methods can deal with data that fit in the main memory but are too big for the traditional visualisation and interaction techniques. Among these methods are interactive visual cluster analysis of trajectories and dynamic aggregation of movement data, which also have no analogues in the pre-project state of the art. By our work in the project, we have also moved forward the theoretical basis for visual analytics methods for movement data. We have introduced a functional model of movement data and on this basis systemised the possible analysis tasks requiring different visual analytics methods.

Not all of our methods have been finished within GeoPKDD. Thus, the method for the summarisation of trajectories needs some elaboration, and the research on the analysis of
interactions between moving agents has just started. The work in the project has also uncovered a number of problems requiring further research.

**What will be available in short, medium, and long term**

We plan to continue the research work that has been partly done or initiated within GeoPKDD. First, we shall complete our work on the generalisation and summarisation of trajectories:

- Find appropriate measures for assessing the quality of the spatial generalisation.
- Elaborate the generalisation method (replacement of trajectory points by areas) so that it could optimise the quality for the chosen level of generalisation.
- Extend the method to spatio-temporal generalisation and summarisation.

Second, we shall continue the work on the analysis of interactions between moving agents. We expect to be able soon to define the potentially relevant features of interactions and the ways for automatic recognition of these features in movement data. On this basis, we shall create techniques for interactive visual clustering and classification of interactions, similar to the methods we have developed for trajectories.

Our medium- and long-term plans include the work on three major research topics. The first is supporting knowledge construction and reasoning. On the one hand, an analyst will take advantage of available ontologies related to space, time, and movement and available methods for formal inference in the course of interactive visual analysis of movement data. On the other hand, in the course of analysis, the analyst will gradually construct a knowledge base about the data, which describes and, possibly, explains the findings and explicitly links related pieces. The knowledge will be represented in a form suitable both for communication to humans and for formal inference and other kinds of automated processing. These ideas have been intensively discussed among the project partners, and the work on implementing them can start quite soon.

The second topic for mid-term research is addressing road-free movements, such as movements of wild animals. Our experiments have shown that some of the methods we have developed do not work well for such a kind of movement. Specifically, the distance functions used in the methods for clustering and cluster-based classification and summarisation turn to be ineffective, although they neither explicitly take into account the road network nor involve any assumptions concerning the character of the movement. At present, we see the main difference of trajectories of wild animals from trajectories of cars in the amount of fluctuation, which is much higher in case of animals. Hence, we need to develop fluctuation-tolerant distance functions and/or methods for smoothing trajectories and reducing fluctuations. Another problem that must be dealt with is the temporal sparseness of position records, which is typical for data about animals due to technical reasons such as the limited lifetime of the batteries of the positioning devices.

The latter problem is related to the third research topic, which is analysis of temporally sparse movement data in general, not only data about animals. In all currently existing methods it is assumed (in most cases implicitly) that position records in movement data closely represent continuous space-time paths and, hence, intermediate positions can be obtained by means of interpolation between known positions. However, there are many
cases when large temporal gaps between available position records do not permit handling movement data in this way. For instance, positions of moving people may be recorded only in the moments when they make phone calls or when they pass special sensors. We shall address this kind of movement data in our new research projects.

**What needs further investigation**

Right from the beginning of the GeoPKDD project the participants recognised that movement data need to be analysed together with multidimensional attributes of the moving entities and the environment where they move (Andrienko et al. 2008b). The consortium also recognised the need for methods to discover relationships between movement and other spatio-temporal phenomena and processes. However, little has been done in these directions not only by the consortium but also by the worldwide research community. The reason is that much effort has been required to handle the analytical complexity of movement data as such. However, the achievements of GeoPKDD lay a base for further research on more comprehensive analysis of movement taking into account various related characteristics and phenomena. This refers, in particular, to the development of visual analytics methods.

Another topic that has not been touched within the project is analysis of emerging spatial relationships among moving entities (e.g. groupings, alignments, etc.) and their evolution over time. Some work on this topic has been done by other researchers, who develop computational methods oriented to specific types of patterns (e.g. Laube et al. 2005). We believe that there is a space here for visual analytics approaches, which can move forward the state of the art.

Very little has been done so far on analysing three-dimensional movements where the altitude or depth is a significant component of a position. This refers to the movement of aircrafts, birds, underwater vehicles and animals, etc. Three-dimensional spatial data are quite challenging for visualisation, and the temporal component of movement data increases the complexity. Analysis of three-dimensional movements may be an area where a good combination of visual and algorithmic approaches is especially needed.

By now, the researchers developing methods for analysis of movement data have mostly ignored the nature of the moving entities, which are typically considered abstractly as moving points. However, the nature of the entities has a significant impact on the properties of the movement. We have already mentioned the difference between the movements of people (by cars) and wild animals. There are many other types of moving entities and, respectively, diverse types of movement. For example, the movements of human eyes in viewing a scene (in psychological experiments) do not obey the physical laws inherently involved in the movements of physical bodies. Very different laws are involved in the movement of infectious diseases. Even some types of physical bodies moving in physical space cannot be treated as just moving points. For instance, the sizes and masses of icebergs change while they move, which affects their further movement, and icebergs can also split into several parts. The diverse types of movement depending on the nature of the moving entities require multidisciplinary research involving domain specialists (e.g. in psychology, epidemiology, iceberg monitoring, etc.). Visual analytics approaches may be particularly helpful in facilitating mutual understanding and cooperation among specialists with diverse backgrounds.
Conclusion

The project GeoPKDD has been very fruitful and has brought into existence an array of new methods enabling the analysis of really large collections of movement data. The visual analytics methods we have developed in cooperation with the project partners are based on the interplay of computational algorithms and interactive visual interfaces, which support the involvement of human capabilities for pattern recognition, association, interpretation, and reasoning. We have also made some progress in developing theoretical foundations, which can reveal the needs for new methods and direct the work on creating such methods. The work in the project has greatly improved our understanding of movement data, their inherent characteristics and frequent problems. We have experimented with several types of real movement data and uncovered similarities and differences between them. We have identified analysis tasks and problems requiring further research.

References


