Visual Mobility Analysis using T-Warehouse

A. Raffaetà[1], L. Leonardi[1], G. Marketos[2], G. Andrienko[3], N. Andrienko[3], E. Frentzos[2], N. Giatrakos[2], S. Orlando[1], N. Pelekis[2], A. Roncato[1], C. Silvestri[1]

[1] Dip. di Informatica, University Ca' Foscari Venezia, Italy
[2] Dept. of Informatics, University of Piraeus, Greece
[3] Fraunhofer Institute for Intelligent Analysis and Information Systems, Germany

Abstract. Technological advances in sensing technologies and wireless telecommunication devices enable novel research fields related to the management of trajectory data. As it usually happens in the data management world, the challenge after storing the data is the implementation of appropriate analytics for extracting useful knowledge. However, traditional data warehousing systems and techniques were not designed for analyzing trajectory data. Thus, in this work, we demonstrate a framework that transforms the traditional data cube model into a trajectory warehouse. As a proof-of-concept, we implemented T-Warehouse, a system that incorporates all the required steps for Visual Trajectory Data Warehousing, from trajectory reconstruction and ETL processing to Visual OLAP analysis to mobility data.

1 Introduction

The usage of location aware devices, such as mobile phones and GPS-enabled devices, is widely spread nowadays, allowing access to vast volumes of trajectory datasets. Effective analysis of such trajectory data on the one hand imposes new challenges for their efficient management, while on the other hand it raises opportunities for discovering behavioral patterns that can be exploited in applications like traffic management and service accessibility.

Data Warehousing and Online Analytical Processing (OLAP) techniques can be employed in order to convert this vast amount of raw data into useful knowledge. Specifically, the variable number of moving objects in different urban areas, the average speed of vehicles, the ups and downs of vehicles' speed can be analyzed in a Trajectory Data Warehouse (TDW) and provide us with useful insights, like discovering popular movements. DWs are optimized for OLAP operations that include the aggregation or de-aggregation of information (called roll-up and drill-down, respectively) along a dimension of analysis, the selection of specific parts of a cube (slicing and dicing) and the reorientation of the multidimensional view of the data on the screen (pivoting) [15].

The motivation behind a TDW is to transform raw trajectories into valuable knowledge that can be used for decision making purposes in ubiquitous applications, such as Location-Based Services (LBS), traffic control management, etc. Intuitively, the high volume of raw data produced by sensing and positioning

technologies, the complex nature of data stored in trajectory databases and the specialized query processing demands make extracting valuable information from such spatio-temporal data a hard task. For this reason, the idea is to develop specific traditional aggregation techniques to produce summarized trajectory information and provide *visual* OLAP style analyses.

It is worth noticing that visual representations of data are essential for enabling a human analyst to understand the data, extract relevant information, and derive knowledge. One of the objectives of visualization is to aid abstraction and generalization [23]. With relatively small and simple data, this can be achieved by appropriate positioning and/or appearance of visual elements representing individual data items. When the data are large and complex, a common approach is to apply computational techniques for data abstraction and generalization, in particular, aggregation. The visualization is then applied to the resulting aggregates. Trajectory Data Warehouse offers a powerful technological support to visual analysis of movement data by efficiently aggregating the data in various ways and at different spatial and temporal scales.

One could mention an abundance of applications that would benefit from the aforementioned approach. As an example, let us consider an advertising company which is interested in analyzing mobility data in different areas of a city in order to decide upon road advertisements (placed on panels on the roads). More specifically, the analysis concerns the demographical profiles of the people visiting different urban areas of the city at different times of the day so as to decide about the proper sequence of advertisements that will be shown on the panels at different time periods. This knowledge will enable them to execute more focused marketing campaigns and apply a more effective strategy.

The above analysis can be efficiently offered by a TDW. However, various issues and challenges have to be considered to develop such a system:

- the presence of a preprocessing phase dealing with the explicit construction of the trajectories, which are then stored into a Moving Object Database (MOD) that offers powerful and efficient operations for their manipulation;
- the implementation of an efficient trajectory-oriented Extract-Transform-Load (ETL) process;
- the incorporation of appropriate aggregation mechanisms suitable for the trajectory oriented cube model;
- the design of a Visual OLAP interface that allows for multidimensional and interactive analysis.

Based on our recent results in the field [17, 16] which to the best of our knowledge are the only works that cope with the problem in all its aspects, as a proof-of-concept, we propose T-Warehouse, a system for Visual Trajectory Data Warehousing. Our contribution can be summarized as follows:

- We describe the architectural aspects of our framework as well as various research challenges that are tackled;
- We suggest the appropriate spatial and temporal visualisation techniques supporting OLAP analysis of movement data. Among these there is a novel

technique called *cross visualisation* that we have designed to represent specific measures of trajectory warehouse, namely numbers of trajectories traversing borders of grid cells.
– We investigate the power, flexibility and efficiency of our framework for applying OLAP analysis on real world mobility data.

The rest of the paper is organized as follows: Section 2 presents the architecture of T-Warehouse and its various components. Section 3 illustrates the functionalities offered by T-Warehouse, by focusing on the visualization tools. Section 4 describes the case study concerning GPS-equipped cars moving in the urban area of Milan (Italy). By using this large dataset we provide an experimental evaluation of the accuracy of our method for computing spatio-temporal aggregates and we demonstrate how different kinds of analysis can be implemented by using T-Warehouse. Section 5 discusses some related work and finally Section 6 draws some conclusions.

2 System Architecture

The overall architecture of T-Warehouse is illustrated in Fig. 1. More specifically, mobile devices are transmitting periodically the latest part of their trajectory, according to some user-defined parameters. This vast amount of data collected by all subscribed users is forwarded to a stream-based module (trajectory reconstruction software), whose purpose is to perform some basic trajectory preprocessing. This may include parameterized trajectory compression (so as to discard unnecessary details and concurrently keep informative abstractions of the portions of the trajectories transmitted so far), as well as techniques to handle missing/erroneous values. These trajectories are stored to Hermes MOD [18] which addresses the need for representing movements of objects (i.e., trajectories) in databases in order to perform querying and analysis on them and for providing efficient indexing, query processing. On Hermes MOD, appropriate querying and ETL processes are applied (possibly taking into account various types of infrastructural geodata) so as to derive information about trajectories (e.g. trajectory content in different granularities, aggregations, motional metadata etc.) to feed in the TDW. Finally, incorporating GIS layers (e.g. geographic, topographic or demographic layers) and combining them with trajectory data results in a conceptually richer framework providing thus more advanced analysis capabilities. Below, we thoroughly illustrate the main components accompanied by our contributions.

2.1 Trajectory Reconstruction

In real-world applications the movement of a spatio-temporal object is often given by means of a finite set of *observations*, i.e., time-stamped positions along with object-ids. The finite set of observations taken from the actual continuous movement is called a *sampling*. A first important task consists in grouping and

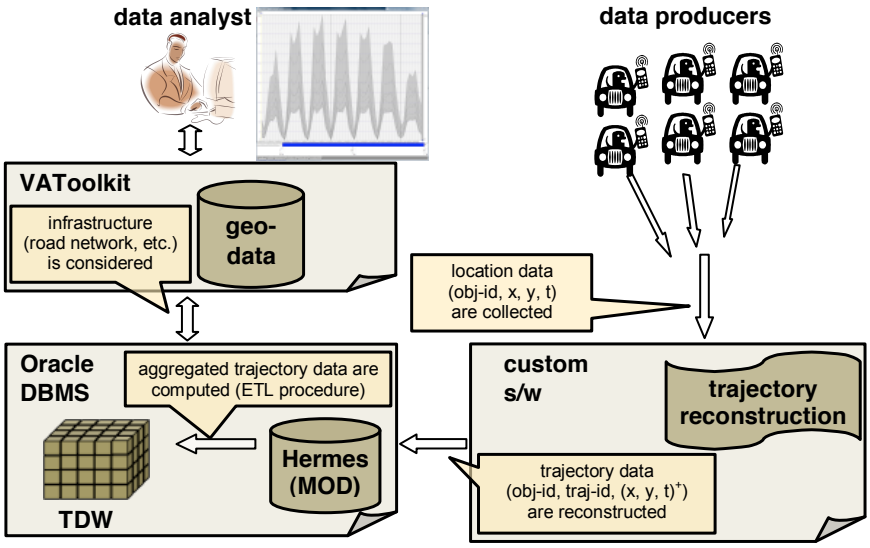


Fig. 1. T-Warehouse architecture

filtering these raw points arriving in streaming in order to generate several meaningful *trajectories*, which are portions of the whole movement of an object [16]. In many situations an (approximate) reconstruction of each trajectory from its sampling is needed. Among the several possible solutions, in this paper we use *linear local interpolation*, i.e., objects are assumed to move straight between two observed points with constant speed. The linear (local) interpolation seems to be a quite standard approach to the problem (see, for example, [20]), and yields a good trade-off between flexibility and simplicity.

The *trajectory reconstruction* module in Fig. 1 accomplishes this task by employing an appropriate algorithm [16]. Due to the fact that the notion of trajectory cannot be the same in every application, we define the following generic trajectory reconstruction parameters:

- *Temporal gap* between trajectories (gap_{time}): the maximum allowed time interval between two consecutive time-stamped positions of the same trajectory for a single moving object. As such, any time-stamped position of object o_i, received after more than gap_{time} units from its last recorded position, will cause a new trajectory of the same object to be created (case *a* in Fig. 2).
- *Spatial gap* between trajectories (gap_{space}): the maximum allowed Euclidean distance in 2D plane between two consecutive time-stamped positions of the same trajectory. As such, any time-stamped position of object o_i, with distance from the last recorded position of this object greater than gap_{space}, will cause a new trajectory to be created for o_i (case *b* in Fig. 2).

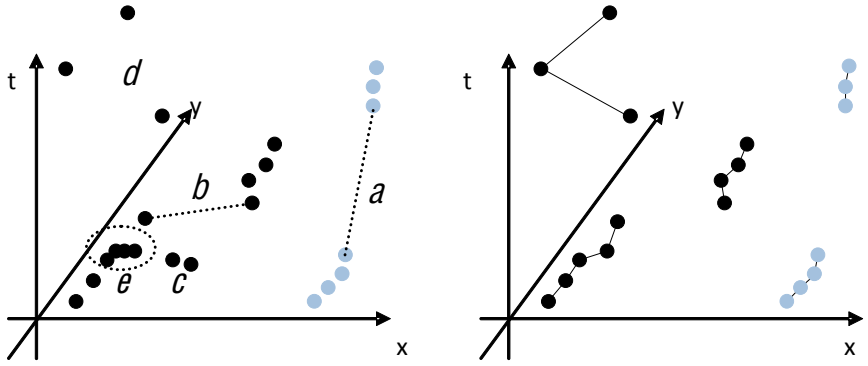


Fig. 2. Raw locations and reconstructed trajectories.

- *Maximum speed* (V_{max}): the maximum allowed speed of a moving object. It is used in order to determine whether a reported time-stamped position must be considered as noise and consequently discarded from the output trajectory. When a new time-stamped location of object o_i is received, it is checked with respect to the last known position of that object, and the corresponding speed is calculated. If it exceeds V_{max}, this location is considered as noise and (temporarily) it is not considered in the trajectory reconstruction process (however, it is kept separately as it may turn out to be useful again - see the parameter that follows) (case c in Fig. 2).
- *Maximum noise duration* ($noise_{max}$): the maximum duration of a noisy part of a trajectory. Any sequence of noisy time-stamped positions of the same object will result in a new trajectory given that its duration exceeds $noise_{max}$. For example, consider an application recording positions of pedestrians where the maximum speed set for a pedestrian is $V_{max} = 3\ m/sec$. When he/she picks up a transportation mean (e.g., a bus), the recorded instant speed will exceed V_{max}, flagging the positions on the bus as noise. The maximum noise length parameter stands for supporting this scenario: when the duration of this sequence of "noise" exceeds $noise_{max}$, a new trajectory containing all these positions is created (case d in Fig. 2).
- *Tolerance distance* (D_{tol}): the tolerance of the transmitted time-stamped positions. In other words, it is the maximum distance between two consecutive time-stamped positions of the same object in order for the object to be considered as stationary. When a new time-stamped location of object o_i is received, it is checked with respect to the last known position of that object, and if the distance of the two locations is smaller than D_{tol}, it is considered redundant and consequently discarded (case e in Fig. 2).

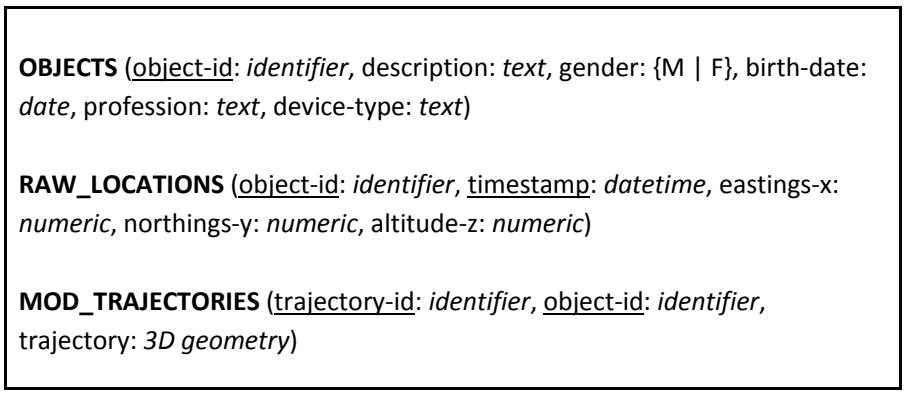


Fig. 3. An example of a MOD.

The algorithm that utilizes the aforementioned parameters is thoroughly presented and evaluated in [16]. It expects as input a set of observations, and a list containing the partial trajectories processed so far by the trajectory reconstruction manager; these partial trajectories are composed by several of the most recent trajectory points, depending on the values of the algorithm parameters.

As a first step, from each observation the algorithm extracts the object identifier and checks whether the object has been processed so far. If so, it retrieves its partial trajectory from the corresponding list, while, in the opposite case, creates a new trajectory and adds it to the list. Then, it compares the incoming point with the tail of the partial trajectory by applying the above mentioned trajectory reconstruction parameters. In this way, the algorithm decides if the incoming point can be considered as part of an existing trajectory or a new one has to be created.

2.2 TDW Schema and loading

Let us assume a MOD that stores raw locations of moving objects (e.g. humans); a typical schema, to be considered as a minimum requirement, for such a MOD is illustrated in Fig. 3.

OBJECTS includes a unique object identifier (*object-id*), demographic information (e.g. *description, gender, date of birth, profession*) as well as device-related technographic information (e.g. GPS type). RAW_LOCATIONS stores object locations at various time stamps (i.e., observations), while MOD_TRAJECTORIES maintains the trajectories of the objects, after the application of the trajectory reconstruction process. Formally, let $D = \{T_1, T_2, \ldots, T_N\}$ be a collection of trajectories of a set of moving objects stored in the MOD. Assuming linear interpolation between consecutive observations, the trajectory $T_i = \langle (x_{i_1}, y_{i_1}, t_{i_1}), \ldots, (x_{i_{n_i}}, y_{i_{n_i}}, t_{i_{n_i}}) \rangle$ consists of a sequence of n_i line segments

in a $3D$ space, where each segment represents the continuous "development" of the corresponding moving object between consecutive locations (x_{i_j}, y_{i_j}) sampled at time t_{i_j} (see the right picture of Fig. 2). Projecting T_i on the spatial $2D$ plane (temporal $1D$ line), we get the *route* r_i (the *lifespan* l_i, respectively) of the trajectory. Additional motion parameters can be derived, including the traversed length *len* of route r_i, average speed, acceleration, etc.

As we mentioned before, our aim is to feed the TDW with aggregate data so as to offer OLAP analysis. Therefore, we need an appropriate TDW schema that can handle trajectory data. Following the multidimensional model [1], a data cube for trajectories consists of a fact table containing keys to dimension tables and a number of measures. The dimensions of analysis include a *spatial* (SPACE_DIM) and a *temporal* (TIME_DIM) *dimension* describing geography and time, respectively. Non spatio-temporal dimensions can be also considered. For example, the schema in Fig. 4 contains the dimension OBJECT_PROFILE_DIM which collects demographical information, such as *gender, age, job*, of moving objects.

Dimensions are organized in hierarchies that favor the data aggregation process. In Fig. 4 the spatial hierarchy is rooted in *Partition_Geometry*, which represents the smallest spatial unit we consider (i.e., a rectangle belonging to a grid which partitions the spatial domain). Further every *Partition_Geometry* is contained in exactly one *district* and the remaining levels of the spatial hierarchy are *city*, *state* and *country*. Similarly, we consider an *interval* of minutes as the minimal temporal unit. Such intervals belong to a *hour*, that can be flagged as a typical one or a *rush_hour*, and is included in one *day*. A day is contained in both a *month* and it is also a *day_of_week*. Finally, the temporal hierarchy is composed by *quarter*, *year*.

Let us now describe the measures of interest for our T-warehouse. We recall that measures represent aggregated information about trajectories of certain profiles that intersect the spatio-temporal cells.

The measure *Pres* for a base cell $bc = (R, T, P)$ represents the number of trajectories having profile P lying in the spatial region R in the time interval T. It is calculated by counting all the distinct trajectory ids belonging to P that pass through the spatio-temporal cell (R, T).

The measure *Distance*, i.e. the average traveled distance of a trajectory in a cell, for a base cell $bc = (R, T, P)$ is computed by introducing an auxiliary measure, called *sum_distance*, defined as follows

$$sum\_distance(bc) = \Sigma_{i \in P, TP_i \in (R,T)} len(TP_i)$$

where TP_i is the portion of the trajectory i which lies within the region R during the time interval T and $len(TP_i)$ is its length. *sum_distance* represents the total distance travelled by trajectories having profile P in R during T. Then, the measure *Distance* can be computed as:

$$Distance(bc) = \frac{sum\_distance(bc)}{Pres(bc)}$$

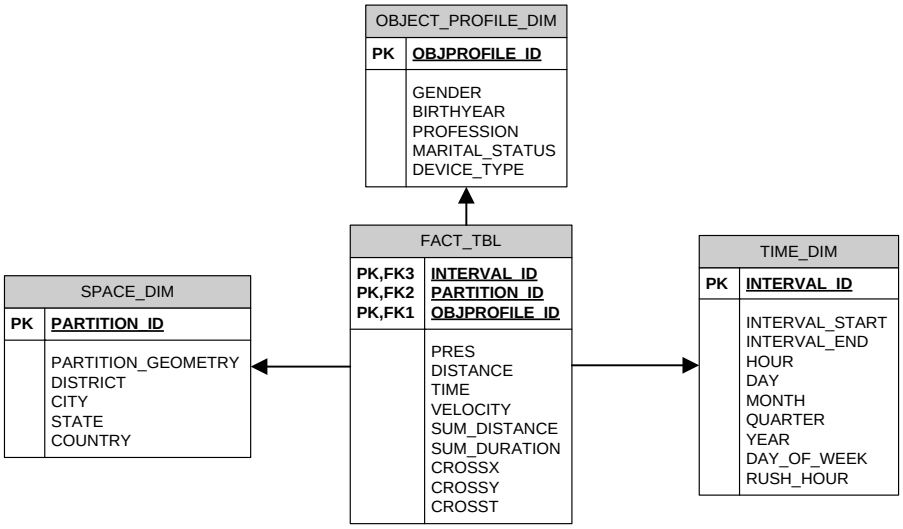


Fig. 4. An example of TDW.

The average travel duration of a trajectory in $bc = (R, T, P)$, represented by the measure *Time*, is computed in an analogous way:

$$Time(bc) = \frac{sum\_duration(bc)}{Pres(bc)}$$

where, *sum_duration* is also an auxiliary measure defined as the summation of the duration *lifespan(TP)* of each portion TP of the trajectories having profile P inside (R, T).

$$sum\_duration(bc) = \Sigma_{i \in P, TP_i \in (R,T)} lifespan(TP_i)$$

The measure *Velocity* is calculated by dividing the auxiliary measure *sum_distance* with *sum_duration*:

$$Velocity(bc) = \frac{sum\_distance(bc)}{sum\_duration(bc)}$$

In a likewise fashion, we could compute and store acceleration by utilizing speed and duration. The remaining measures ($CrossX$, $CrossY$, $CrossT$) are auxiliary measures that will be defined in the following subsection.

It is worth remarking that for base cells all these measures are computed in an exact way by using the MOD. This is possible thanks to the fact that our MOD Hermes [19] provides a rich palette of spatial and temporal operators for handling trajectories. Unfortunately, rolling-up these measures is not straightforward due to the count distinct problem [21] as it will be discussed in detail in the next subsection.

In order to calculate the measures of the data cube, we have to extract the portions of the trajectories that fit into the base cells of the cube. In [16], we proposed and evaluated two alternative strategies for computing the measures: a cell-oriented (COA) and a trajectory-oriented (TOA) one. Fig. 5 illustrates the application of the COA approach on the two trajectories that lie within three spatio-temporal cells. First, the procedure searches for the portions of trajectories under the constraint that they reside inside each spatio-temporal cell (R, T) (the start/end of each portion has been marked with a circle containing a dot). Then, the algorithm proceeds to the decomposition of the portions with respect to the user profiles they belong to. The efficiency of the above described COA solution depends on the effective computation of the parts of the moving object trajectories that reside in the spatio-temporal cells. This step is actually a spatio-temporal range query that returns not only the identifiers but also the portions of trajectories that satisfy the range constraints. To efficiently support this trajectory-based query processing requirement, we employ the TB-tree [20], a state-of-the-art index for trajectories that can efficiently support trajectory query processing. On the other hand, the TOA approach discovers the spatio-temporal cells where each trajectory resides in. The main challenge here is to avoid checking all cells. This becomes possible by utilizing Minimum Bounding Rectangles of trajectories as rough approximations of them and by exploiting the fact that the granularity of cells is fixed in order to detect (possibly) involved cells in constant time. Further details about the two approaches as well as a comparison study can be found in [16].

2.3 Aggregation

In order to allow for OLAP processing, T-Warehouse offers aggregation capabilities over measures, i.e., operations for computing measures at some higher level of the hierarchy starting from those at lower level. The aggregate functions computing the super-aggregates of the measures are categorized by Gray et al. [12] into three classes according to the complexity required for this computation:

- *distributive*, the super-aggregates can be computed from the sub-aggregates;
- *algebraic*, the super-aggregates can be computed from the sub-aggregates with a finite set of auxiliary measures; and
- *holistic*, the super-aggregates cannot be computed from sub-aggregates, even if we employ auxiliary measures.

According to this classification, *sum_distance* and *sum_duration* are *distributive* since we can aggregate such measures by using the function *sum* whereas *Velocity* is *algebraic*: we need the auxiliary measures $\langle sum\_distance, sum\_duration \rangle$. For a cell C arising as the union of adjacent cells, the aggregate function performs a component-wise addition, thus producing a pair $\langle sum\_distance_f, sum\_duration_f \rangle$. Then the average speed in C is given by $sum\_distance_f / sum\_duration_f$.

The most complex measures are *Pres*, *Distance* and *Time* which are *holistic*. In fact, since a trajectory might span multiple base cells, in the aggregation

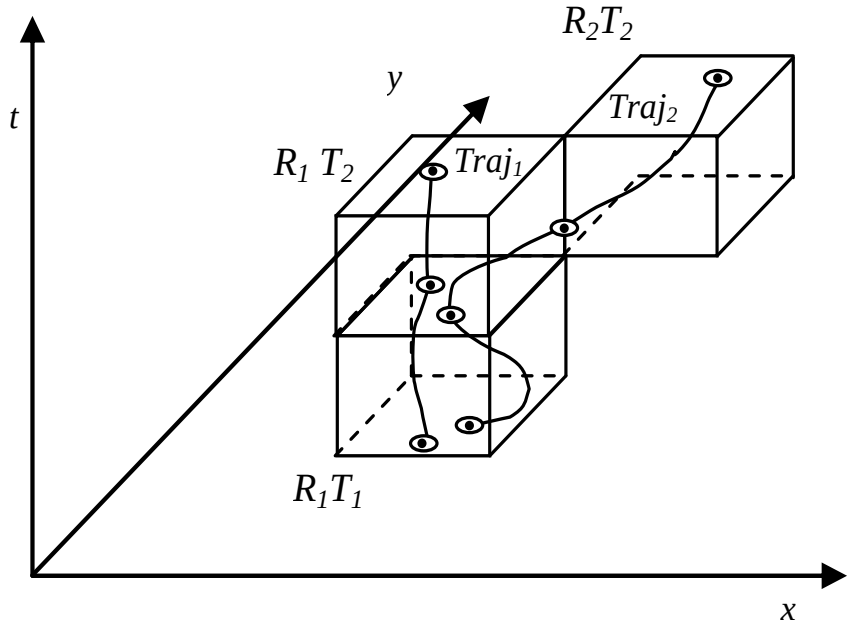


Fig. 5. Applying the Cell Oriented Algorithm.

phase we have to cope with the so called *distinct count problem* [21]: if an object remains in the query region for several timestamps during the query interval, one should avoid to count it multiple times in the result. This is problematic since, once loaded in the TDW, the identifiers of the trajectories are lost. This problem causes aggregation hindrances in OLAP operations for the above measures.

Notice that once a technique for rolling-up the measure *Pres* is devised, it is straightforward to define a roll-up operation for *Distance* and *Time*. In fact the latter can be implemented as the sum of the corresponding auxiliary measures (*sum_distance* and *sum_duration*) divided by the result of the roll-up of *Pres*. Hence, we will focus only on the measure *Pres*.

In order to implement a roll-up operation over *Pres*, a first solution is to define a *distributive* aggregate function, denoted by $Pres_{Distr}$, which simply obtains the super-aggregate of a cell C by summing up the measures *Pres* in the base cells composing C. In the literature, this is a common approach to aggregate spatio-temporal data but, as we will show in Section 4.2, it produces a very rough approximation.

Following the proposal in [17], an alternative solution is to define an *algebraic* aggregate function, denoted by $Pres_{Alg}$. More formally, let $C_{(x,y),t,p}$ be a base cell, which contains, among the others, the following measures:

- $C_{(x,y),t,p}.Pres$: the number of distinct trajectories of profile p intersecting the cell.
- $C_{(x,y),t,p}.CrossX$: the number of distinct trajectories of profile p crossing the spatial border between $C_{(x-1,y),t,p}$ and $C_{(x,y),t,p}$, where $C_{(x-1,y),t,p}$ is the adjacent cell (on the left) along with x-axis.
- $C_{(x,y),t,p}.CrossY$: the number of distinct trajectories of profile p crossing the spatial border between $C_{(x,y-1),t,p}$ and $C_{(x,y),t,p}$, where $C_{(x,y-1),t,p}$ is the adjacent cell (below) along with y-axis.
- $C_{(x,y),t,p}.CrossT$: the number of distinct trajectories of profile p crossing the temporal border between $C_{(x,y),t-1,p}$ and $C_{(x,y),t,p}$, where $C_{(x,y),t-1,p}$ is the adjacent cell (below) along with t-axis.

Let $C_{(x',y'),t',p'}$ be a cell consisting of the union of two adjacent cells with respect to a spatial/temporal dimension, for example $C_{(x',y'),t',p'} = C_{(x,y),t,p} \cup C_{(x+1,y),t,p}$ (when aggregating along x-axis). In order to compute the super-aggregate corresponding to $C_{(x',y'),t',p'}$, we proceed as follows:

$$C_{(x',y'),t',p'}.Pres = C_{(x,y),t,p}.Pres + C_{(x+1,y),t,p}.Pres - C_{(x+1,y),t,p}.CrossX$$

The other measures associated with $C_{(x',y'),t',p'}$ can be computed as follows:

$$\begin{aligned} C_{(x',y'),t',p'}.CrossX &= C_{(x,y),t,p}.CrossX \\ C_{(x',y'),t',p'}.CrossY &= C_{(x,y),t,p}.CrossY + C_{(x+1,y),t,p}.CrossY \\ C_{(x',y'),t',p'}.CrossT &= C_{(x,y),t,p}.CrossT + C_{(x+1,y),t,p}.CrossT \end{aligned}$$

The computation of $C_{(x',y'),t',p'}.Pres$ can be thought of as an application of the well-known Inclusion/Exclusion principle for sets: $|A \cup B| = |A| + |B| - |A \cap B|$. Note that in some cases $C_{(x+1,y),t,p}.CrossX$ is not equal to $|A \cap B|$, and this may introduce errors in the values returned by this algebraic function. In fact, if a trajectory is fast and agile, it can be found in both $C_{(x,y),t,p}$ and $C_{(x+1,y),t,p}$ without crossing the X border (since it can reach $C_{(x+1,y),t,p}$ by crossing the Y borders of $C_{(x,y),t,p}$ and $C_{(x+1,y),t,p}$ as shown in Fig. 6(a)). In the following figures we illustrate the two main kinds of error that the algebraic aggregate function can introduce in the roll-up phase due to the agility of trajectories. In Fig. 6(a), if we group together the cells C_1 and C_2, we obtain that the number of distinct trajectories is $C_1.Pres + C_2.Pres - C_2.CrossX = 1 + 1 - 0 = 2$. This is an overestimate of the number of distinct trajectories. On the other hand, in Fig. 6(b), if we group together C_1 and C_2 we correctly obtain $C_1.Pres + C_2.Pres - C_2.CrossX = 1 + 1 - 1 = 1$, similarly by aggregating C_3 and C_4. However, if we group $C_1 \cup C_2$ with $C_3 \cup C_4$ we obtain $C_1 \cup C_2.Pres + C_3 \cup C_4.Pres - C_1 \cup C_2.CrossY = 1 + 1 - 2 = 0$. This is an underestimate of the number of distinct trajectories.

Note that in order to face the distinct count problem when aggregating cells with different profiles, analogously to what we did for the spatial and temporal dimensions, it could be helpful to consider a measure *crossP*, specifying the number of distinct trajectories changing their profile from one cell to an adjacent one. However, since profile changes are rather rare in real-world scenarios and

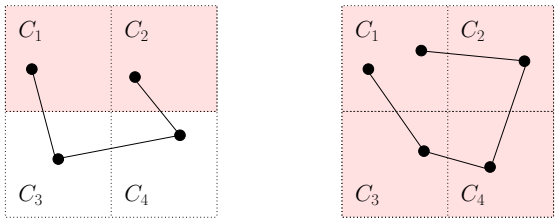


Fig. 6. (a) Overestimate of Presence, and (b) underestimate of Presence during the roll-up.

only appear in long term situations, we omit computing *crossP* and we simply use the distributive aggregate function *sum* for this kind of aggregations. (In any case, when needed, *crossP* can be added in our framework without additional difficulty.)

3 OLAP and Visualisation

A TDW serves two core needs: to provide the appropriate infrastructure for advanced reporting capabilities and to facilitate the application of trajectory mining algorithms on the aggregated data. According to their needs, end users could have access either to basic reports or OLAP-style analysis. What-if scenarios and multidimensional analysis are typical examples of analytics that can be supported by a TDW. Some interesting questions in the context of traffic monitoring, that an analyst may want to answer via the functionalities offered by TDW, are "When and in which area of the town does the most intense traffic appear?", if we consider the road network, "'which are the most trafficated roads?", "Is there any difference in traffic between the working days and the week-end?, "How does the movement propagate from place to place?".

Even if standard, table based OLAP operations could be used to answer this kind of queries, the interpretation of results, and the consequent refinement of queries and exploration of results, is not immediate. Integrating OLAP tools with Geographical Information Systems (GISs) provides advanced analysis capabilities. For instance, trajectory data can be georeferenced in a map, combined with several layers (such as topographic, demographic, thematic). Finally, performing OLAP operations on TDW specialised measures in a visual way makes the exploration of the data cube more rapid and intuitive.

We developed OLAP visual operations, by using the Visual Analytics Toolkit [3], an interactive Java™ based geographical information system. This toolkit permits a user to view georeferenced data over a map. It also offers functionalities to handle temporal data, by using graphs or animations, according to the type of data to analyse.

By using our system, it is simple to handle and visualise the spatio-temporal grids of the TDW at various levels of granularities. If the roll-up operation

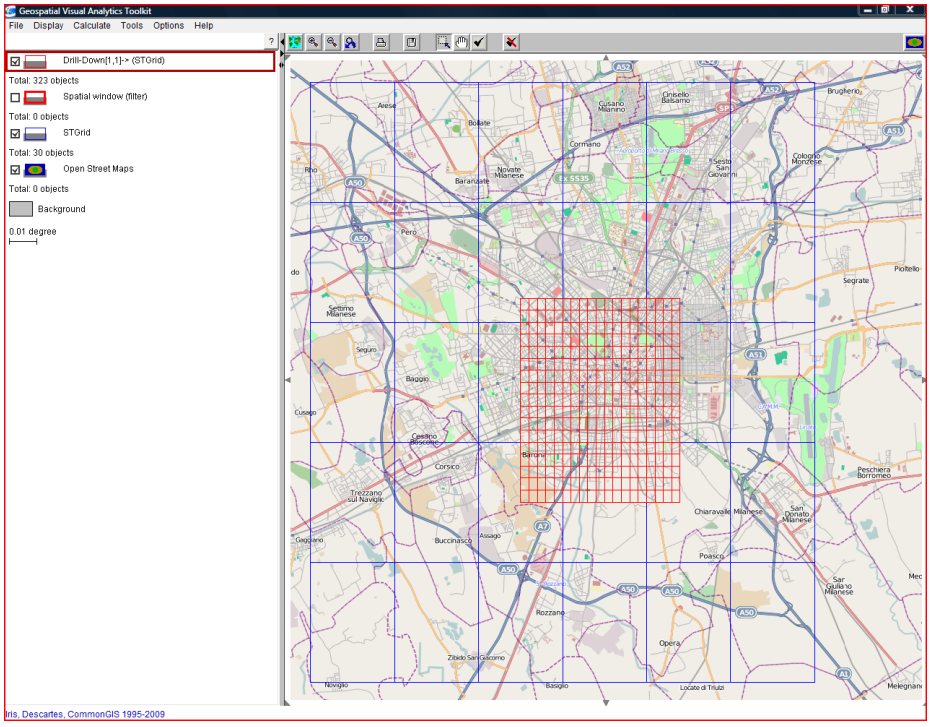


Fig. 7. Drill-down

involves the spatial dimension, visually this affects the granularity of the grid which becomes larger. The inverse operation is the drill-down which increases the level of detail of data; it allows the user to descend into the hierarchies. In this case, we can select the spatial area we are interested in and if we reduce the spatial dimension of the cells, a smaller grid is visualised as shown in Fig. 7.

Starting from this visualisation of the space, one can then decide to highlight some measures, which can be visualised according to several methods. The *unclassified choropleth map* technique fills the grid cells with colour shades so that the degree of darkness is proportional to the value of a selected measure. For building a *classified choropleth map*, the value range of the selected measure is divided into intervals, also called *classes*. Each class is assigned a particular color. These colors are then used for filling the grid cells on the map. In the *Triangle visualisation* technique, a triangle is drawn in each grid cell at a chosen level of the TDW hierarchy. The base and the height of such a triangle correspond to the values of two selected measures that the user wants to analyze. The *Line thickness* visualisation style draws linear symbols whose thickness is proportional to the value of a given TDW measure. These visualisation meth-

ods can be used in animated displays, where each frame represents the selected measure(s) in one time interval from the period of interest.

Cartographic visualization techniques offer limited opportunities for the examination of the temporal variation of the data. This weakness needs to be compensated by using additional visualisations appropriately representing the temporal aspect, such as the composite time series display demonstrated in Fig. 9 and 10. The display consists of two parts with a common horizontal axis representing the time period under study divided into intervals. The upper part is a generalized time graph. The vertical axis represents the value range of the selected measure. Instead of the lines showing the variation of the measure in each grid cell at a given granularity over time, there is a polygon enclosing all the lines. The lower and upper boundaries of the polygon show the ranges of the values in each time interval. Additional details are provided by dividing the polygon area into 10 parts. The division is done as follows. For each time interval, the range of values of the measure is divided into deciles, i.e. 10 parts containing approximately equal number of values. The positions of the corresponding deciles in consecutive time intervals are connected by lines and the areas between the lines are filled in two different shades of grey. On top of this, a thick black line represents the temporal variation of the mean value from all grid cells, which is computed for each interval.

The lower part of the display is a temporal histogram. The vertical dimension represents the number of cells. Each segmented bar shows the statistical distribution of the values of the measure in one time interval. For this purpose, the overall range of the values is divided into intervals, or classes, and each class is given a particular color. According to the chosen color scale, shades of blue correspond to low values (the lower, the darker) and shades of red to high values (the higher, the darker). The division into the classes and the corresponding colors are shown in the upper part of the display by background painting of the time graph area. Each bar in the time histogram is divided into segments filled with the colors assigned to the classes. The heights of the segments are proportional to the numbers of the grid cells whose values belong to the respective classes. Grey-colored segments stand for the cells where the aggregate values are not defined. The upper and lower parts of the display provide two complementary overall views of the temporal variation of the data.

4 Applying T-Warehouse to traffic data

In this section, first, we quantitatively evaluate the roll-up accuracy of our T-Warehouse. In particular, we show the error in computing *Pres* since, as discussed in Section 2.3, it is an approximation of the exact value and this affects also the measures *Distance* and *Time*. Then, we illustrate the use of the visual OLAP functionalities offered by T-Warehouse through several examples. Both analyses are based on a large mobility dataset described below.

4.1 Dataset

We used a real world dataset containing the observations of GPS-equipped cars moving in the urban area of Milan (Italy). The dataset consists of two millions of raw location records that represent the movement of 17,000 objects (i.e. about 200,000 trajectories) moving during a week period from Sunday to Saturday. As base granularity, we set a grid of rectangles, of size $330m \times 440m$, and time intervals of 1 hour. The spatial hierarchy aggregates groups of 10-20-40-80 spatially adjacent base cells, whereas the temporal hierarchy is hour- 3-hours interval-day-week. Unfortunately, the dataset does not contain any details about the demographical profiles of the different objects. However, even in this case where the schema of the TDW consists just of a spatial and a temporal dimension, our framework does not loose in expressive power as it is demonstrated in Section 4.3.

4.2 Accuracy of spatio-temporal aggregates

Before presenting the results of our experiments, we first define the metric that we use to quantify the overall error for the measure *Pres*, generated by an aggregation operation. Then we describe the sketches based algorithm adopted in [21] and used in our experiments.

In order to compare the errors we chose to adopt as an aggregation accuracy metric the normalized absolute error defined as follows:

$$Error = \frac{\sum_C Error(C)}{\sum_C C.Pres} = \frac{\sum_C |\widetilde{C.Pres} - C.Pres|}{\sum_C C.Pres} \quad (1)$$

where C are cells at a coarser granularity than the base one, $C.Pres$ is the exact value of *Pres* in the cell C whereas $\widetilde{C.Pres}$ is the approximated value obtained using one of the discussed methods, i.e. $Pres_{FM}$ (sketches), $Pres_{Distr}$ or $Pres_{Alg}$.

FM sketches The FM algorithm is a bitmap-based algorithm devised by Flajolet and Martin [8] that can be used to estimate the number of distinct items in a set using a limited amount of memory. Each entry in the sketch used by FM is a bitmap of length $r = \log UB$, where UB is an upper bound on the number of distinct items. A hash function h maps every object ID i (trajectory identifiers in our case) to a pseudo-random integer $h(i)$ corresponding to a position in the r-bit sketch that will be set (the whole bitmap is initially unset). The values are mapped by h according to a geometric distribution, that is, the probability that a generic ID i will be mapped to a position v is $Prob[h(i) = v] = 2^{-v}$ for $v \geq 1$.

After processing all objects, the most simple version of FM approximates the overall object count with 1.29×2^k, where k is the position of the leftmost bit of the sketch that is still unset. Unfortunately, this approach may entail large errors in the count approximation. For this reason, the authors of [8] propose the adoption of m sketches that use different and independent hash functions.

Only one randomly selected sketch is modified on update, thus each sketch becomes responsible for approximately n/m (distinct) objects. Then, the count is computed by using all sketches.

Interestingly, FM sketches can be merged in a *distributive* way. Suppose that a pair of sketches are updated according to the IDs of the objects contained in a different set, and that the intersection of those sets is possibly not empty. The sketch obtained as the *bitwise-OR* of the corresponding bitmaps in the original sketches will be identical to the one directly updated using the union of the sets of items.

Quantitative evaluation In Fig. 8 we compare the accuracy of these different approximate aggregation methods. The graphs show the normalised absolute errors as functions of cell granularities. Cell granularities are reported as values relative to the base one. For example, $g = 2$ indicates that we are considering cells having double size w.r.t. the base cells along all dimensions.

Notice that we avoid plotting the error for $g = 1$, corresponding to base cells, because here we are interested in the aggregation error ($g > 1$). Further, we recall that at the base granularity the measure *Pres* is exact because by using the spatio-temporal operators offered by the MOD the base cells are loaded with the correct values.

As shown by the corresponding curves, the *distributive* aggregate function (the top curve) quickly reaches very large errors as the roll-up granularity increases. This is due to the fact that we simply sum the sub-aggregates and as a consequence trajectories crossing different cells are counted many times: the number of duplicates becomes higher and higher at coarser granularities. Conversely, we obtained very accurate results with our *algebraic* method, especially at small granularities where the error is less than 3%. One can observe that the cumulative error starts increasing when larger granularities g are considered, since the number of trajectories that visit the various cells several times gets larger but the error remains always smaller than 10%. Finally, we can remark that for all granularities the aggregate function $Pres_{Alg}$ outperforms sketches and we also save memory. We highlight that in order to obtain an accuracy around 10% 40 sketches have to be used, each $32 - bit$ long, that is ten times the memory allocated by the four counters used by our algebraic aggregation method.

4.3 Visual OLAP analysis

In this subsection we present the functionalities and the flexibility of T-Warehouse for the visual analysis of the Milan dataset.

First of all we want to study how the traffic varies along the week and answer the query: "When does the most intense traffic appear?" The time series display in Fig. 9 summarizes the temporal variation of the measure *Pres* over the whole territory (i.e. all grid cells). The time period of the data (one week from Sunday to Saturday) has been divided into hourly intervals. The territory has been

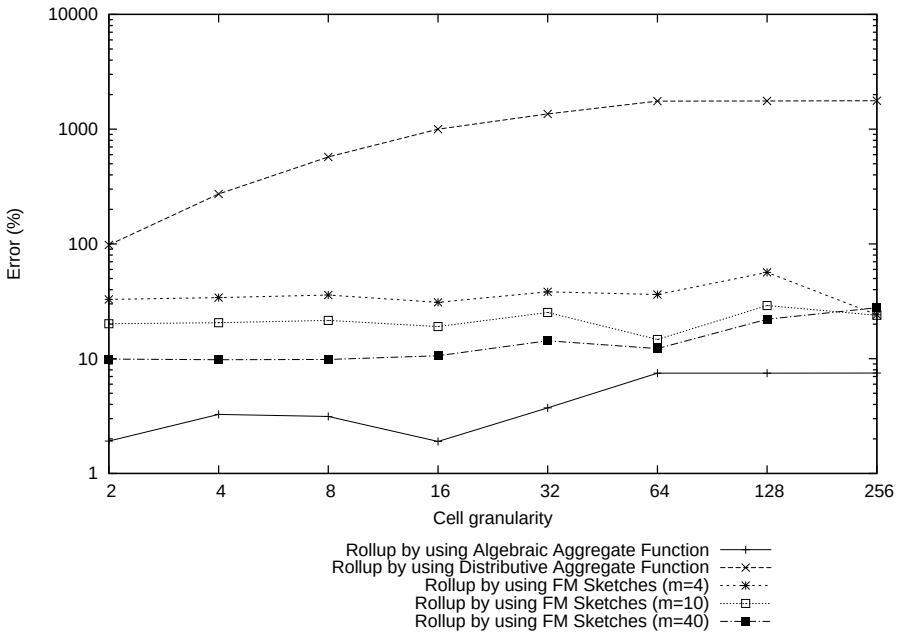


Fig. 8. Cumulative errors of roll-up phase.

divided into cells of the size $3.3km \times 4.4km$, i.e. 10 base cells are aggregated together along the x and y axes. The appearance of the display shows a clear subdivision of the whole time period into days. We can observe that the presence is much higher in the day hours than in the night and noticeably higher on the working days than on Sunday and Saturday. On each of the working days, there are two peaks of the number of cells with high presence, signified by the shades of red. These peaks correspond to the morning and afternoon rush hours, which occur in the intervals $6-9am$ and $3-6pm$. Interesting is the increase of traffic intensity on the Sunday afternoon. It is also visible that the traffic on Friday was less intense than on the previous working days: there were no cells with the values lying in the upper two classes of the values of presence.

Comparing the display of the presence with the display of the speed of the objects at the same granularity, shown in Fig. 10, one can immediately realise that presence and average speed are inversely proportional. During the early and late hours of the day the speed is high whereas from 6am up to 6pm the speed decreases significantly, exhibiting a dual behaviour with respect to the presence.

The composite time series displays representing the temporal evolution of the measures need to be combined with cartographic visualisations showing the data in the spatial context. For example, Fig. 11 is a screen-shot of the animation representing the values of the speed and the presence by triangular symbols.

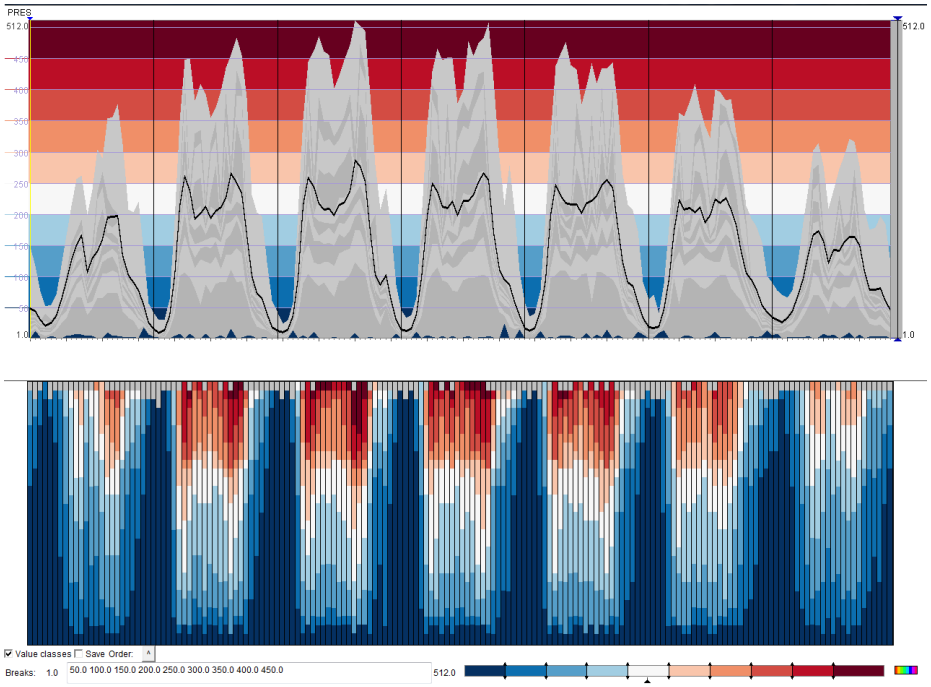


Fig. 9. The evolution of *Pres* during the week

The height of the triangle is proportional to the speed and the base to the presence. One animation frame corresponds to one hourly interval and the whole animation shows the variation of the presence and speed over the week. This reveals additional information with respect to the time series displays. Thus, the image in Fig.11 shows that the presence is higher in the centre and this has a strong impact on the speed of cars, which is very low. On the other hand, it highlights that along the ring roads, the speed is higher except in the north-east zone where the larger number of cars slows down the traffic.

Next, we compare the data of our DW at two different spatial granularities. We roll-up the data illustrated in Fig. 11 by aggregating two adjacent rectangles and 3 consecutive hours. Fig. 12(a) and 12(b) show the 8 screenshots of the data at 0-3am, 3-6am, ... taken on Tuesday and on Saturday, as representative of the situation on a working day and on the week-end. We chose an unclassified choropleth map, that gives us an overall view of the data: the denser is the traffic in a cell, the darker is its colour. During the working days, we can see that the traffic is concentrated in the centre and in the north-east areas of Milan and the rush hours are from 6am to 9am and from 3pm to 6pm, even though the centre is crowded also from 6pm up to 9pm. On the week-end, the densest area remains the centre but the peak of the traffic is reached in a different time

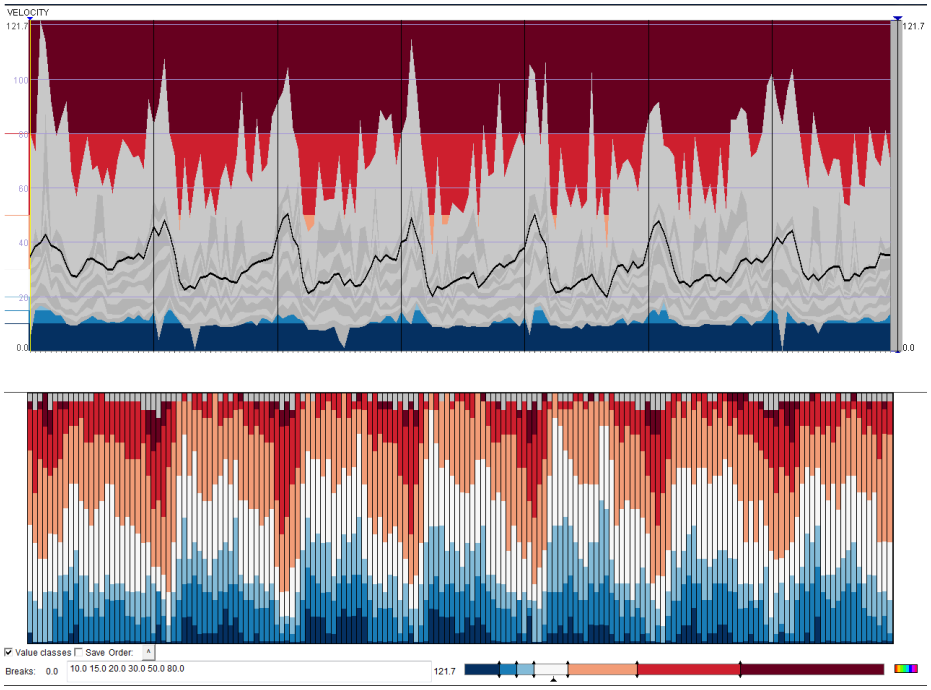


Fig. 10. The evolution of *Velocity* during the week

interval starting later, around 9am instead of 6am but remaining more sensibly intense for the whole night.

Now we apply to these data a drill-down operation in order to obtain the data at the base granularity for the spatial dimension. The result is visualised in Fig. 12(c) and Fig. 12(d) using the technique of classified choropleth map. Like in Fig.9, the shades of red represent high values of the presence and the shades of blue low values. At this level of detail, the information about the presence is strictly connected to the main roads. We can distinguish several rings around the centre and some radial streets that are used to enter/exit to/from the centre. This allows us to answer queries about the traffic conditions at the road network level and their evolution over time. It is interesting to notice that from 0am to 3am on Tuesday there are few cars moving around, and there is no dense area. Then the outer ring of the town becomes denser and later the inner rings and the radial roads. It may be concluded that in the morning there is a flow from the outside to the centre. An opposite pattern can be observed in the second part of the day (not illustrated in the figure). On Saturday (Fig. 12(d)) the situation is different. From 0am to 3am (the night from Friday to Saturday) there is traffic in some radial roads which reveals movement closer to the centre. From 3am to 9am, however, the traffic is not as intense as on Tuesday. Later it becomes denser, and there is traffic up to midnight also in the radial roads.

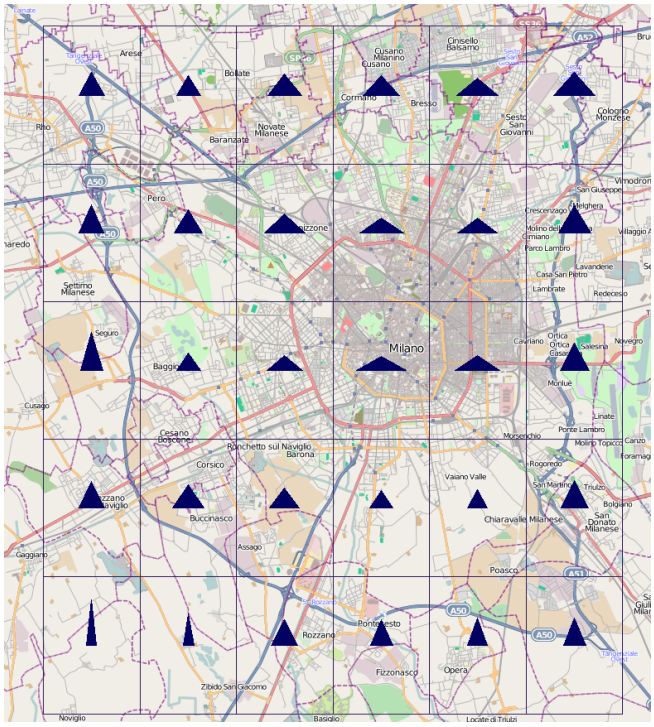


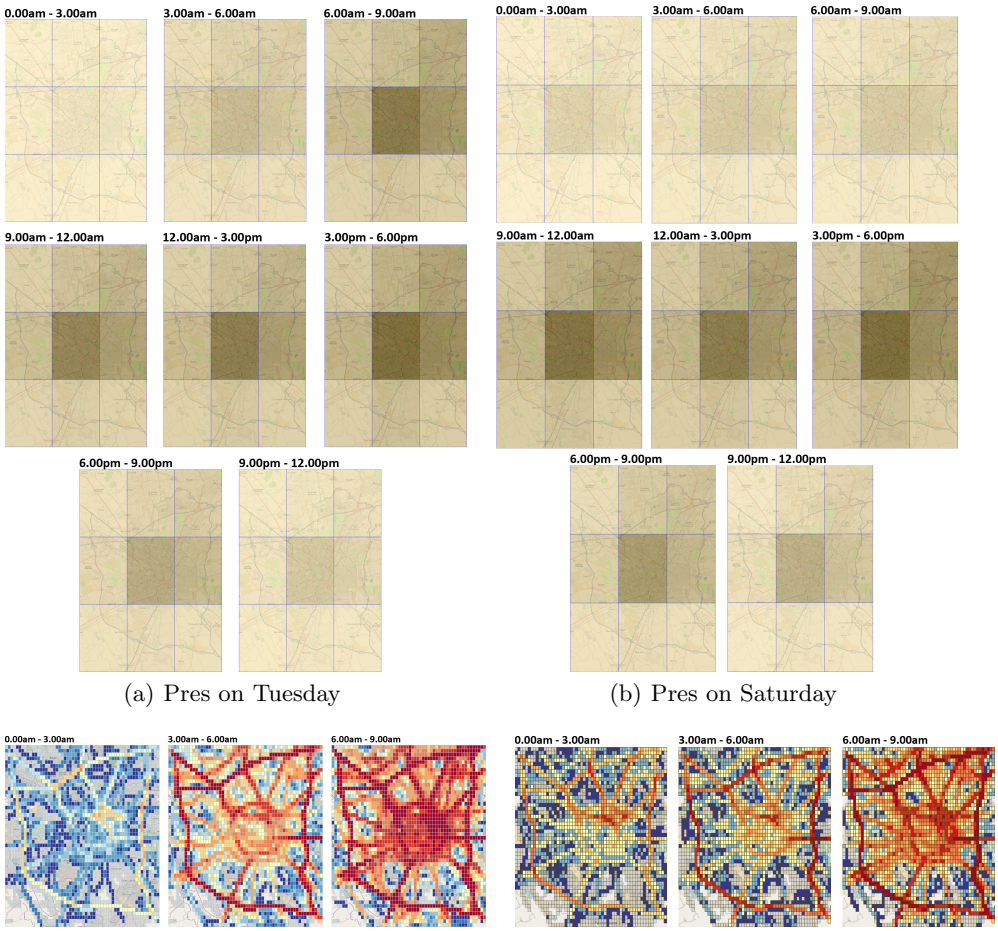
Fig. 11. Relationship between *Pres* (widths of the triangles) and *Velocity* (heights of the triangles).

In order to understand how the traffic flows from one cell to the other ones, we can use the *Cross Visualisation* operation which is intended to illustrate the cross measures, i.e. the number of trajectories traversing the x border and y border of a cell. The idea is that the thickness of the lines of the grid is proportional to the values of the cross, thus providing a qualitative representation of these measures. In Fig. 13 the measure *CrossX* (crossing of x border) is represented by vertical lines, whereas the measure *CrossY* (crossing of y border) by the horizontal lines.

Finally, T-Warehouse provides the user with an operation called *Cyclic Time analysis* which allows for a kind of cyclic aggregation. For instance, it is possible to capture what happens on Mondays, on Tuesdays and so on for the whole period of analysis, thus aggregating data concerning the same days of the week.

5 Related work

The research in TDW has intersections with two research fields extensively studied over the last decade, namely spatial data warehouses and moving object databases. In [11] the authors present a complete survey of both fields, as well



(a) Pres on Tuesday

(b) Pres on Saturday

(c) *Pres* on Tuesday at base granularity

(d) *Pres* on Saturday at base granularity

Fig. 12. *Pres* at different granularities

as a description of the emerging works on Spatio-Temporal Data Warehouses (STDW).

The pioneering work by Han et al. [14] introduces the concept of spatial data warehousing (SDW). The authors extend the idea of cube dimensions so as to include spatial and non-spatial ones, and of cube measures so as to represent space regions and/or calculate numerical data. One step further from modeling a SDW is modeling a STDW. As stated in [25] there is no commonly agreed definition of what a STDW is and what functionality such a data warehouse should support. In [25] the authors propose a conceptual framework for defining STDWs and a taxonomy for spatio-temporal OLAP queries through which they classify the approaches in literature. According to this classification, T-Warehouse is very expressive as it succeeds in supporting Spatio-Temporal OLAP queries.

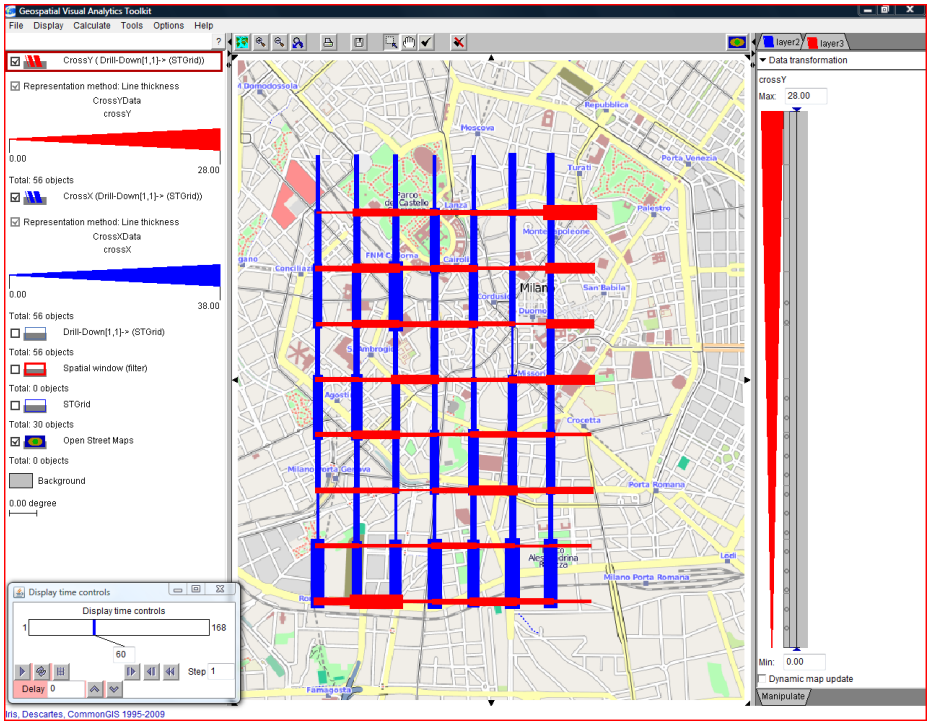


Fig. 13. Visualisation of CrossX and CrossY

Another major research direction concerns the efficient implementation of aggregate queries. Tao and Papadias [22] propose a technique based on the combined use of specialised indexes and materialisation of aggregate measures. Choi et al. [5] try to overcome the limitations of multi-tree structures by introducing a new index structure that combines the benefits of Quadtrees and Grid files. However, the above frameworks focus on calculating simple measures (e.g. count customers) and they do not cope with trajectories.

Traffic analysis is a topic that has been largely studied in the past, even if nowadays the large availability of trajectory data makes it possible to perform innovative and accurate analyses. To the best of our knowledge, however, this is the first work that leverages the depth of analyses allowed by a TDW, and the intuitive interaction obtained thanks to the visual spatio-temporal OLAP interface to support the decision making of traffic analysts.

Visual analysis of large collections of movement data is one of the research topics in the area of geographic visualisation. Starting from the work by Fredrikson et al. [10], spatial, temporal, and attributive aggregations have been applied to movement data. Temporally aggregated data are represented, for instance, by means of a temporal histogram where the bars correspond to time intervals and their heights are proportional e.g. to the number of locations visited or the

distance traveled [7]. Spatial aggregation produces a statistical surface, which is visualized on a map. Spatio-temporal aggregation produces a series of surfaces (one surface per time interval) visualized by means of an animated map display [7, 9]. In these works, movement data are treated as a set of independent points in space and time.

Another way of aggregating movement data is based on considering the data as a set of moves between predefined places (spatial compartments). Each move is treated as a vector characterized by its origin and destination places, start and end times, and, possibly, additional attributes such as duration and travelled distance. Moves with coinciding origins and destinations are united into aggregate moves, which are characterized by the count of the original moves and other statistics. The results may be visualized as a transition matrix where the rows and columns correspond to the places and symbols in the cells or cell coloring or shading encode the derived attribute values [13]. An obvious disadvantage is the lack of spatial context. Another technique is flow map, where aggregated moves are represented by bands or arrows connecting pairs of locations [24]. When this kind of spatial aggregation is combined with temporal aggregation, the result can be visualized by an animated matrix or flow map display or by a juxtaposed sequence of such displays. Drecki and Forer [6] use a three-dimensional representation to show aggregate moves corresponding to several consecutive time intervals (reproduced in [4]).

The work [2] surveys the methods that are used for aggregation of movement data and visualization of the resulting aggregates and proposes some novel techniques designed specifically for this kind of data. By this moment, there were no published works concerning visual analysis of movement data with the use of trajectory data warehouses.

6 Conclusions

This paper discussed the main design issues concerning a DW which stores aggregate measures computed over trajectories and allows performing OLAP analyses over both the temporal and spatial dimensions. In particular, we focused on issues related to storing and aggregating (rolling-up) the holistic measure *Pres*, which, along with other measures (speed, distance covered, etc.), is very useful to convey actionable knowledge to a traffic analyst. Moreover, we demonstrated how T-Warehouse can be used within a visual analytics environment for enabling interactive analysis and interpretation of the data.

Finally, we discussed the usage of T-Warehouse in the context of traffic analysis. In particular we presented a set of OLAP visual operations that permit answering interesting questions in the context of traffic monitoring. We showed a real use case which regards a large real dataset storing the trajectories of a fleet of cars moving in the metropolitan area of Milan (Italy).

References

1. S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In *VLDB*, pages 506–521, 1996.
2. G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *Proc. of IEEE Symposium on Visual Analytics Science and Technology (VAST 2008)*, pages 51–58. IEEE Computer Society Press, 2008.
3. G. Andrienko, N. Andrienko, and S. Wrobel. Visual Analytics Tools for Analysis of Movement Data. *ACM SIGKDD Explorations*, 9(2):28–46, 2007.
4. N. Andrienko and G. Andrienko. Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42(2):117–138, 2007.
5. W. Choi, D. Kwon, and S. Lee. Spatio-temporal data warehouses using an adaptive cell-based approach. *DKE*, 59(1):189–207, 2006.
6. I. Drecki and P. Forer. Tourism in new zealand - international visitors on the move (a1 cartographic plate). Technical report, Tourism, Recreation Research and Education Centre: Lincoln University, Lincoln, 2000.
7. J. A. Dykes and D. M. Mountain. Seeking structure in records of spatio-temporal behavior: visualization issues. *Computational Statistics and Data Analysis*, 43(4):581–603, 2003.
8. P. Flajolet and G. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.
9. P. Forer and O. Huisman. *Information, Place and Cyberspace: Issues in Accessibility*, chapter Time and Sequencing: Substitution at the Physical/Virtual Interface, pages 73–90. Springer Verlag, 2000.
10. A. Fredrikson, C. North, C. Plaisant, and B. Shneiderman. Temporal, geographical and categorical aggregations viewed through coordinated displays: a case study with highway incident data. In *Proc. Workshop on New Paradigms in information Visualization and Manipulation*, pages 26–34, 1999.
11. L. Gómez, B. Kuijpers, B. Moelans, and A. Vaisman. A survey on spatio-temporal data warehousing. *International Journal of Data Warehousing and Mining*, 5(3):28–55, 2009.
12. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–54, 1997.
13. D. Guo. Visual Analytics of Spatial Interaction Patterns for Pandemic Decision Support. *International Journal of Geographical Information Science*, 21(8):859–877, 2007.
14. J. Han, N. Stefanovic, and K. Kopersky. Selective materialization: An efficient method for spatial data cube construction. In *PAKDD'98*, 1998.
15. R. Kimball, M. Ross, W. Thornthwaite, J. Mundy, and B. Becker. *The Data Warehouse Lifecycle Toolkit, 2nd Edition: Practical Techniques for Building Data Warehouse and Intellingent Business Systems*. John Wiley & Sons, 2008.
16. G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, and Y. Theodoridis. Building real world trajectory warehouses. In *Proc. of 7th Int. ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, pages 8–15, 2008.
17. S. Orlando, R. Orsini, A. Raffaetà, A. Roncato, and C. Silvestri. Trajectory Data Warehouses: Design and Implementation Issues. *Journal of Computing Science and Engineering*, 1(2):240–261, 2007.

18. N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. HERMES: aggregative LBS via a trajectory DB engine. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1255–1258. ACM, 2008.
19. N. Pelekis, Y. Theodoridis, S. Vosinakis, and T. Panayiotopoulos. Hermes – a framework for location-based data management. In *EDBT'06*, pages 1130–1134, 2006.
20. D. Pfoser, C. S. Jensen, and Y. Theodoridis. Novel Approaches in Query Processing for Moving Object Trajectories. In *VLDB'00*, pages 395–406, 2000.
21. Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-temporal aggregation using sketches. In *ICDE'04*, pages 214–225, 2004.
22. Y. Tao and D. Papadias. Historical spatio-temporal aggregation. *ACM TOIS*, 23:61–102, 2005.
23. J. Thomas and K. Cook, editors. *Illuminating the Path: The Research and development Agenda for Visual Analytics*. IEEE Computer Society, 2005.
24. W. Tobler. Experiments in migration mapping by computer. *The American Cartographer*, 14(2):155–163, 1987.
25. A. Vaisman and E. Zimányi. What is spatio-temporal data warehousing? In *DaWaK '09: Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery*, pages 9–23, Berlin, Heidelberg, 2009. Springer-Verlag.