T-Warehouse: Visual OLAP Analysis on Trajectory Data

Luca Leonardi[#1], Gerasimos Marketos[*2], Elias Frentzos[*3], Nikos Giatrakos[*4], Salvatore Orlando[#5], Nikos Pelekis[*6], Alessandra Raffaetà[#7], Alessandro Roncato[#8], Claudio Silvestri[#9], Yannis Theodoridis[*10]

[#]*Dept. of Informatics, University Ca' Foscari Venezia, Italy*

[1]leonardi@dsi.unive.it, [5]orlando@dsi.unive.it, [7]raffaeta@dsi.unive.it, [8]roncato@dsi.unive.it, [9]silvestri@dsi.unive.it

[*]*Dept. of Informatics, University of Piraeus, Greece*

[2]marketos@unipi.gr, [3]efrentzo@unipi.gr, [4]ngiatrak@unipi.gr, [6]npelekis@unipi.gr, [10]ytheod@unipi.gr

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 on mobility data.**

I. 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, 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) [3].

The motivation behind a TDW is to transform raw trajectories to valuable information 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 extend traditional aggregation techniques to produce summarized trajectory information and provide OLAP style analyses.

One could mention an abundance of applications that would benefit from the aforementioned approach. 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 have to be considered:

- 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 [5], [4], 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 developed 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 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 II presents the architecture of our framework and its various components. Section III illustrates the functionalities that can be tested throughout the demonstration.

II. System Architecture

The overall architecture of T-Warehouse is illustrated in Fig. 1. Below, we present the main components accompanied by our contributions.

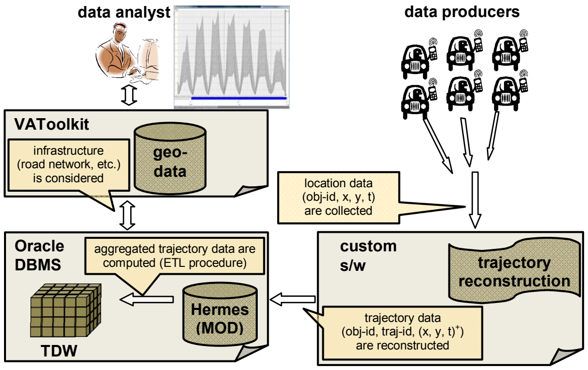

data analyst
data producers
VAToolkit
infrastructure (road network, etc.) is considered
geo-data
location data (obj-id, x, y, t) are collected
Oracle DBMS
aggregated trajectory data are computed (ETL procedure)
Hermes (MOD)
TDW
custom s/w
trajectory reconstruction
trajectory data (obj-id, traj-id, (x, y, t)+) are reconstructed

Fig. 1 T-Warehouse architecture

Trajectory Reconstruction. Sampled positions received by GPS-enabled devices need to be converted into trajectory data, describing the continuous movements of objects, and stored in a MOD. To this end, we use a *trajectory reconstruction* technique that transforms sequences of raw sample points, i.e., time-stamped positions along with object-id, into meaningful trajectories [4].

Due to the fact that the notion of trajectory can vary according to the application domain, our trajectory reconstruction algorithm relies on the following parameters: *temporal gap* (the maximum allowed time interval between two consecutive time-stamped positions of the same trajectory for a single moving object), s*patial gap (*the maximum allowed distance in 2D plane between two consecutive time-stamped positions of the same trajectory), *maximum speed (*the maximum allowed speed of a moving object), *maximum noise duration (*the maximum duration of a noisy part of a trajectory) and *tolerance distance* (the maximum distance between two consecutive time-stamped positions of the same object in order for the object to be considered as stationary).

TDW feeding. Let us consider as a running example a sample TDW schema, illustrated in Fig. 2, which includes a *spatial* (SPACE_DIM) and a *temporal* (TIME_DIM) *dimension* describing geography and time, respectively. Non spatio-temporal dimensions can be also considered. For instance, the schema in Fig. 2 contains the dimension OBJECT_PROFILE_DIM which collects demographical information, such as gender, age, job, of moving objects.

The TDW stores aggregations about trajectories of certain profiles that intersect the spatio-temporal cells. Hence, apart from the keys to dimension tables, the fact table also contains a set of measures representing aggregate information. The measures considered in the TDW schema of Fig. 2 include the *number of distinct trajectories* (PRES), the *average traveled distance* (DISTANCE), the *average travel duration* (TIME), the *average velocity* (VELOCITY) and some auxiliary measures (i.e. CROSSX, CROSSY, CROSST), for a particular group of people (having a certain profile) moving in a specific spatial area during a specific time period.

The TDW is to be fed with aggregate trajectory data; this is achieved by employing an efficient ETL process so as to fill in the measures of the TDW with the appropriate numeric values for each base cell. Our ETL process, thoroughly presented in [4], detects the trajectory portions that lie within the base cells. This step actually corresponds to spatio-temporal range queries that return not only the identifiers but also the portions of trajectories that satisfy the range constraints. To efficiently support the above described storage as well as trajectory-based query processing requirements, we use the HERMES MOD engine [6] that employs TB-tree [7], a state-of-the-art index for trajectory databases.

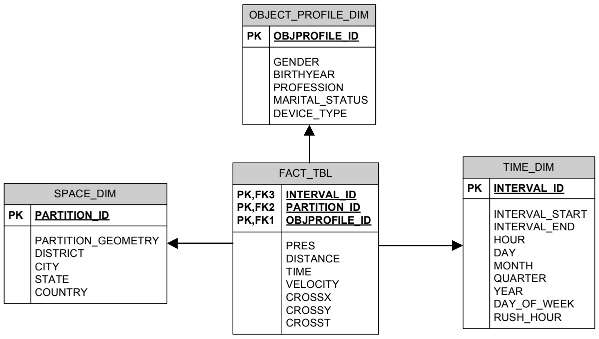


Fig. 2 An example of TDW.

Aggregation. Aggregation capabilities over measures are offered for OLAP purposes (i.e., how the measures at a lower level of the cube hierarchy can be exploited in order to compute the measures at some higher level of the hierarchy). A peculiarity with trajectory data is that a trajectory might span multiple base cells. Hence in the aggregation phase we have to cope with the so called *distinct count problem* [8]: 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 example in the computation of the measure PRES that should return the number of *distinct* trajectories of a certain profile crossing a spatio-temporal cell. This affects also the other measures (*average*) defined on top of PRES. In order to tackle this problem, we use an approximate solution, presented in [5], which turns out to perform effectively. Assuming that the PARTITION_GEOMETRY in Fig. 2 is a regular grid, in the auxiliary measures CROSSX, CROSSY and CROSST we store respectively the number of *distinct* trajectories of a certain profile crossing the spatial/temporal border of two adjacent cells along the x/y/t axis. Knowing the number of trajectories crossing the border between cells is helpful in correcting the errors due to the duplicates when aggregating such cells (see [4] for more details).

OLAP operations and Visualization. The usual OLAP operations provided by DWs [3], [2] are not completely adequate to analyze trajectories. In fact, since trajectories refer to information concerning space and time, the ordinary DW representation based on relational tables makes it very difficult for the user to grasp the relationships between areas in the same neighborhood, the evolution in time of spatial areas, or the correlations of different measures.

In order to overcome these limits we developed visual OLAP operations, by using the Visual Analytics Toolkit (VAToolkit) [1], an interactive Java-based geographical information system. This toolkit permits a user to see geo-referenced data over a map and, it also offers functionalities to handle temporal data, by using graphs or animations, according to the type of data to analyze.

The advantages of our system are manifold. First the user can visualize the partition of the spatial domain over the map which the spatial data refer to. Additionally, the user can graphically select an area and apply roll-up and drill-down operations in order to obtain, respectively, a more abstract or detailed view of such an area. Moreover, s/he can focus on a particular time interval, i.e. week-end or a selected month, and analyze how the measures vary in that period. To these views the user can apply a variety of specialized visualization techniques, which provide insightful understanding of the measures contained in the TDW.

In summary, the visual interface we implemented allows the user to easily navigate the data stored inside the TDW at different levels of the hierarchies, to have an overall vision of the data in time and in space or to focus on some particular measures, spatial areas or temporal intervals.

III. DEMO SPECIFICATIONS

For the purposes of the demonstration, we will use a large real dataset of a fleet of cars moving in the metropolitan 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. Throughout the demonstration users will be able to test all the components of T-WAREHOUSE architecture; i.e. a) the trajectory reconstruction technique so as to produce custom trajectories b) the ETL process in order to feed the TDW with new data and c) the Visual OLAP interface. To motivate the demonstration scenario of T-WAREHOUSE, below we pose some interesting questions that an analyst could interactively try to answer via the functionalities offered by T-WAREHOUSE:

- Where does the highest traffic appear? At what hour?
- What happens exactly at the road network level?
- How does the movement propagate from place to place?

Before showing the T-WAREHOUSE at work we describe the specific features of the TDW prototype used in this scenario.

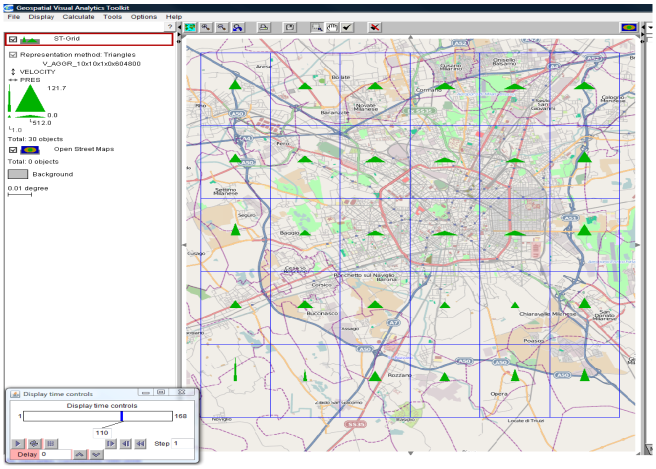
Design choices. The user can choose the base granularity both for the spatial and temporal dimension and the corresponding hierarchies. We set a grid of rectangles (PARTITION_GEOMETRY in Fig. 2), the size of which is 300×400 m^2, and time intervals of 1 hour, as base granularity. The spatial hierarchy consists of a set of grids aggregating groups of 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 lose in expressive power as it is demonstrated below. We implemented all the measures shown in Fig. 2.

GUI and Visual analytics. We present the functionalities provided to the analyst by the T-WAREHOUSE. The accompanying video [9] concretely shows the interaction with the system of a user studying the traffic in the centre of Milan.

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

Starting from the visualization of the space, one can then decide to highlight some measures, which can be visualized according to several methods.

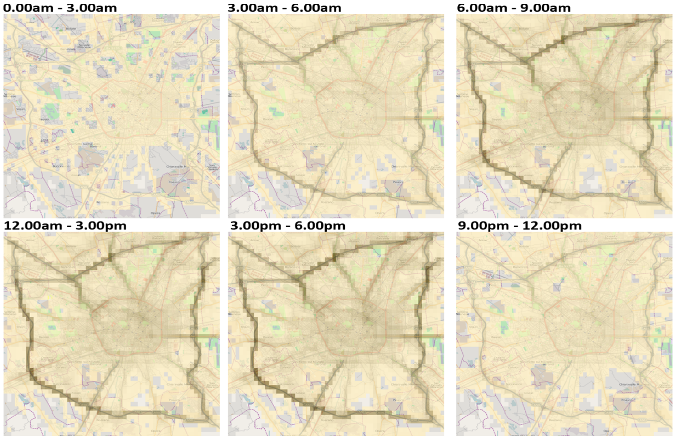
In the *Triangle* visualization style, a triangle is drawn in each grid cell at a given level of the TDW hierarchy. The base and the height of such a triangle correspond to the value of two selected measures that the user wants to analyze. As an example, Fig. 3 shows a screenshot of an animation that illustrates the variation of the speed and the presence along each hour of the whole week, using the triangle style. The height of the triangle represents the Velocity whereas the base represents the Presence. Note the underlying map of Milan, which allows us to better understand the traffic phenomenon. The presence is higher in the centre and this has a strong impact on the speed of cars that is very low. On the other hand, along the ring roads the speed is higher except in the north-east zone, where the larger number of cars slows down the traffic.



Representation method: Triangles
V_AGGR_10x10x1x0x604800
VELOCITY
PRES
Total: 30 objects
Total: 0 objects

Fig. 3 Relationship between Presence and Velocity

The *unclassified choropleth map* is a visualization style, in which all the grid cells are filled with a color shade, according to a classification of the value of a selected TDW measure. This style is illustrated in Fig. 4 which reports 6 screenshots respectively taken at 0-3 am, 3-6 am, 6-9 am, 12am-3pm, 3-6pm, 9-12pm of Tuesday (a working day). The images give us a qualitative view of the measure PRES: the denser is the traffic in a cell, the darker is its color. Compared to the grid of Fig. 3, this finer level of spatial granularity highlights the road network: several road rings around the centre, and some radial streets used to enter/exit to/from the centre. During the rush hours the traffic increases in the centre of the city, as well as in the main roads. From 0am to 3am there are few cars moving around since there are no dense areas; then the outer road ring of the town becomes denser, and after, the inner rings and the radial roads.


0.00am - 3.00am

3.00am - 6.00am

6.00am - 9.00am

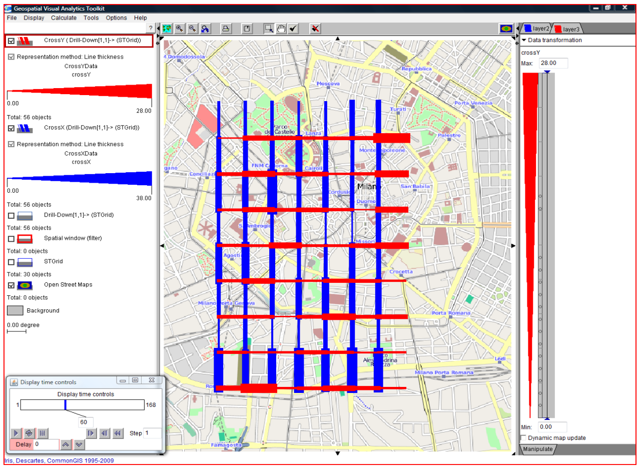
12.00am - 3.00pm

3.00pm - 6.00pm

9.00pm - 12.00pm

Fig. 4 Presence on Tuesday at base granularity

The *Line thickness* visualization style, instead, permits us to draw lines whose thickness is proportional to the value of a given TDW measure. In the screenshot of Fig. 5 these lines are used to visualize the *cross* measures. The measure CROSSX (crossing of X border) is represented by the vertical lines, whereas the measure CROSSY (crossing of Y border) by the horizontal lines.



☑ CrossY (Drill-Down[1,1]-> (STGrid))

☑ Representation method: Line thickness
CrossYData
crossY
0.00
28.00
Total: 56 objects

☑ CrossX (Drill-Down[1,1]-> (STGrid))

☑ Representation method: Line thickness
CrossXData
crossX
0.00
38.00
Total: 56 objects

☐ Drill-Down[1,1]-> (STGrid)
Total: 56 objects

☐ Spatial window (filter)
Total: 0 objects

☐ STGrid
Total: 30 objects

☑ Open Street Maps
Total: 0 objects

Background

0.00 degree

Display time controls
1 168
60
Step 1
Delay 0

Data transformation
crossY
Max: 28.00
Min: 0.00
☐ Dynamic map update

Iris, Descartes, CommonGIS 1995-2009

Fig. 5 Visualization of CROSSX and CROSSY

The described visualization methods can produce also animations, in which each frame represents the selected measure(s) in a time interval of the period of interest. In this way the user can obtain a visual representation of the variations of such measure(s) in different zones of the target space, and during different time intervals.

Another type of visualization is the *Time Graph* which generates a graph showing the temporal evolution of a selected measure. As an example, Fig. 6 reports the time graph representing the evolution of the measure PRES, along the week starting from Sunday up to Saturday at a granularity of 6 x 8 km^2 for the spatial dimension and of 3-hour interval for the temporal dimension. We can clearly recognize the week-days: the traffic grows during a day and decreases in the late hours of the same day. During the week-end the presence is definitely lower than in the working days. Remark that each curve of the graph is associated with a cell of our grid and this correspondence is highlighted by clicking on the curve.

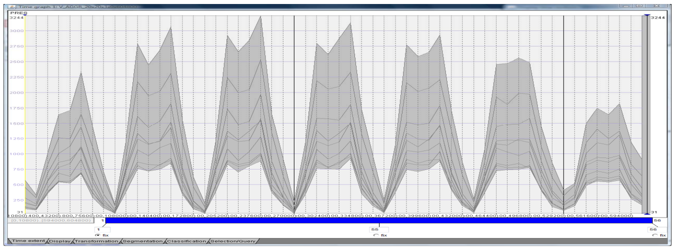


Fig. 6 The evolution of Presence during the week

ACKNOWLEDGMENT

Research partially supported by the EU FP6-14915 IST/FET Project GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery). Gerasimos Marketos was also supported by a PENED'2003 grant funded by the GSRT of the Greek Ministry of Development. The screenshots have been done by using the Geospatial visual analytics toolkit developed at the Fraunhofer Institute IAIS.

REFERENCES

[1] G. Andrienko, N. Andrienko, and S. Wrobel. Visual Analytics Tools for Analysis of Movement Data. ACM SIGKDD Explorations, 9(2), pp. 28-46, 2007.

[2] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing groub-by, cross-tab and sub-totals. *DMKD*, 1(1):29-54, 1997.

[3] 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.

[4] G. Marketos, E. Frentzos, I. Ntoutsi, N. Pelekis, A. Raffaetà, and Y. Theodoridis, Building Real World Trajectory Warehouses. *Proc. MobiDE*, 2008.

[5] S. Orlando, R. Orsini, A. Raffaetà, A. Roncato, and C. Silvestri. Trajectory Data Warehouses: Design and Implementation Issues. *JCSE*, 1(2):240-261, 2007.

[6] N. Pelekis, E. Frentzos, N. Giatrakos and Y. Theodoridis. HERMES: Aggregative LBS via a Trajectory DB Engine, *Proc. SIGMOD*, 2008.

[7] D. Pfoser, C.S. Jensen, and Y. Theodoridis. Novel Approaches to the Indexing of Moving Object Trajectories, *Proc. VLDB*, 2000.

[8] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-Temporal Aggregation Using Sketches. *Proc. ICDE*, 2004.

[9] http://www.dsi.unive.it/~leonardi/Video.html