Extracting Trajectories with the Map Cube Operator

Francisco Moreno[1]*, Juan Camilo Alzate*[2] *and Jesús Hernández*[1,*]

[1] Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia
[2] Facultad de Ingeniería, Universidad de Antioquia, Medellín, Colombia

Abstract: OLAP (On Line Analytical Processing) is a set of techniques and operators to facilitate the analysis of data that are usually stored in a specialized database known as data warehouse. Data is analyzed in order to provide information about a particular business subject called *fact*, for example, crimes, sales, shipments, among others. In this paper, we extend the functionality of an OLAP operator, known as map cube, with the definition and incorporation of a function called *Trajectory*. This function allows the analysts extract the trajectory followed by a set of facts. For example, the trajectory of crimes of each suspect, starting from his/her oldest crime to his/her more recent one. In order to prove the expediency of our proposal, we developed a prototype that was applied to a case study of crimes in Pennsylvania. Although more extensive experiments are required, our results show patterns that can help to understand the *modus operandi* of suspects.

Keywords: Data Warehouse, Map Cube, Trajectory, KML, OLAP operators.

1 Introduction

A data warehouse (DW) is a subject-oriented, integrated, time-variant, and non-volatile collection of data supporting a management decision making process [1]. A DW can be seen as a complete record of an organization's data, stored in a database designed to facilitate the analysis and integration of data.

A DW is usually modeled in a multidimensional way [2], [3]. A multidimensional model is a set of dimensions that are associated with a subject of analysis called *fact*. A dimension represents an organizational perspective to analyze the facts. A dimension has levels, which are organized in a hierarchy. For example, in a Time dimension, with levels day, month and year, there is a hierarchical relationship such that the years are grouped into months and the months are grouped into days. A fact has measures, i.e., indicators that allow the analysts evaluating a specific aspect of an organization. For example, consider a DW of crimes, the dimensions Suspect and Type of crime may be used to analyze the crimes (facts), where the number of victims may be the measure to evaluate (for details, see Section 2).

On the other hand, OLAP (On Line Analytical Processing) is a solution often used to speed up and facilitate the formulation of queries in a DW. OLAP allows users the manipulation and the aggregation of data using specialized operators like roll-up, drill-down, slice, dice, and drill-across [4].

Map cube is another OLAP operator. It was proposed by [5] as a spatial extension of the data cube operator [6], which in turn is a generalization of the GROUP BY clause of the SQL SELECT sentence. Given n grouping columns, the data cube operator creates subtotals for all possible combinations of these columns, i.e., 2^n subtotals. Each combination is called a cuboid. For example, if Suspect and Type of crime are grouping columns then four cuboids are generated. The generated subtotals (e.g., total of victims) are: i) for each suspect and each type of crime, ii) for each suspect, iii) for each type of crime, and iv) for all data, i.e., the whole total.

Now suppose that in addition to the number of victims, there is a measure that represents the locations, e.g., points (latitude and longitude), where the crimes were committed. The map cube operator generates the spatial aggregation of such measures. For example, a set of points can be spatially aggregated with a geometric union [7]. Map cube then associates a map with each cuboid generated by data cube displaying the spatial and non-spatial aggregations.

The map cube operator only provides spatial aggregation using the geometric union function. To

---

* Corresponding author e-mail: jahernan@unal.edu.co

overcome this limitation, Moreno et al. [8] extended the map cube operator in order to support different spatial aggregation functions such as center of mass, convex hull, Voronoi diagram, among others.

However, in its current state, map cube does not offer spatio-temporal aggregation functions [8], [9], [10], [11], [12]. For example, suppose a crimes analyst has data about dates and places (geographical coordinates) where crimes were committed. The analyst is interested in the trajectory of crimes of each suspect, starting from his/her oldest crime to his/her more recent one. The trajectory is formed by a set of tuples (x, y, t), where x and y represent the coordinates and t the time. To support this requirement, we propose a spatio-temporal trajectory aggregation function and its incorporation to the map cube operator.

Next, we present some related works. Balzer et al. [13] propose OLAP operators to group trajectories and present a tool for their visualization. In [14] the authors define a DW to store entire trajectories as measures and address the problem of their aggregation and visualization [15], e.g., the amount of trajectories that passed through an area. This allows users the visualization of high traffic areas. Because the simultaneous visualization of multiple trajectories can lead to confused and crowded deployment, Demsar [16] proposes a method to facilitate that visualization. Bakshev et al. [17] propose a framework for trajectory aggregation queries (e.g., show all trajectories that passed through the downtown area of a city) and their 3D visualization. Biadgilgn et al. [18] present four methods to visualize trajectories: on a single map, several maps which show gradual temporal evolution of the trajectories, animation, and 3D using a spatio-temporal cube. Sack et al. [19] present a survey about the representation, analysis and visualization of trajectories of moving objects.

The identified works include aspects such as trajectory aggregation (to identify high traffic areas), storage of entire trajectories in a DW, and tools for trajectory visualization. However, none of them allows us reconstructing the trajectories from the DW according to the grouping columns (dimensions) that the user selects. In this paper, we propose a function to meet this requirement.

On the other hand, GIS (geographic information systems) such as ArcGIS [20], Kosmo [21], Capaware [22], among others, offer functions, e.g., to locate a particular place, to calculate optimal routes between two points, to calculate the distance between two points, among others. However, GIS often lack advanced analytics tools [23], [24] and usually work with geographic data separate from other business data [25]. Indeed, in [24] the authors outline the features that should support a spatio-temporal DW, including the aspect of aggregation.

A case study of crimes, with real data taken from the Division of Public Safety University of Pennsylvania [26] was used to validate our proposal. The Trajectory function was implemented in a WAMP environment (Windows, Apache, MySQL, and PHP). The graphic capabilities of Google Earth [27], were used for visualization. The KML markup language (Keyhole Markup Language) [28] based on XML was used to represent the trajectories in a web environment. PHP was used to query the database data, generate the respective cuboids, and implement the Trajectory function.

The paper is organized as follows: in Section 2, we present a motivating example for our proposed function. In Section 3, we define and propose the Trajectory function. In Section 4, we present some experiments with real data of crimes. In Section 5, we conclude and present future work.

2 Illustrative Example

Consider a DW of crimes whose conceptual model is shown in Figure 1. We use the notation of [7]; its essential symbols are shown in Figure 2. For each crime (fact) we record the day of occurrence, the suspect, the type of crime (robbery, murder, rape, among others), the number of victims and the spatial coordinates (latitude and longitude) of the place of occurrence of the crime. Table 1 presents a sample of data. Note that in this DW, each fact represents the set of the same type of crimes committed by the same suspect on the same date (day). For example, according to Table 1 the suspect Sp_2 stole in two places on 10-Feb-2009.

Suppose the analyst wants to get the total number of victims and the center of mass of the crimes under the following classifications: by type of crime and suspect, by type of crime, by suspect, and by the whole total. To achieve this we may use the map cube operator [5] with the proposed extension from [8] as follows:

Base Map	Crimes Map
Base Table	Crimes
Aggregate by	SUM: #Victims AS
	Total victims,
	CENTER OF MASS: Place AS
	Center of mass of crimes
Reclassify by	Type of crime, Suspect
	Cartographic preference: Overlay:
	(Meta Map)

A brief description of the terms of the operator is: i) Base Map indicates the referenced map by the points of the measure Location (Crimes Map), ii) Base Table indicates the fact table (Crimes), iii) Aggregate by indicates the measures to be aggregated with the corresponding aggregation function, iv) Reclassify by indicates the grouping columns (dimensions), and v) Cartographic preferences indicates the preference of visualization, here a map called Meta Map, where the results generated by the operator are overlapped. Table 2

and Figure 3 show a part of the results generated by the previous sentence, in particular results from the suspect cuboid.

Table 1: Sample 1 of our data warehouse of crimes.

Levels			Measures	
Day	Type_of _Crime	Suspect	# Victims	Location (Lat - Long)
20-Jul-2009	Robbery	Sp_1	1	{(2. 4815, 56.1600)}
10-Aug-2009	Murder	Sp_1	1	{(2.1505, 55.2230)}
29 Oct 2009	Murder	Sp_1	2	{(3.3755, 55.4500)}
02-Jan-2009	Rape	Sp_2	1	{(3.0505, 57.1500)}
30-Jan-2009	Rape	Sp_2	1	{(3.1340, 58.1350)}
10-Feb-2009	Robbery	Sp_2	3	{(4.0205, 57.4400), (3.5550, 57.2730)}

Table 2: Data of the suspect cuboid.

Suspect	Total _ victims	Center _of mass _ of _ crimes
Sp_1	4	{(2.5345, 55.475)}
Sp_2	5	{(3.3410, 57.3950)}

Suppose now, that instead of the center of mass, the analyst wants the trajectory followed by the crimes. That is, the total number of victims and the trajectory followed by the crimes by type of crime and suspect; by type of crime, by suspect and by the whole total. For example, consider the trajectory of all crimes attributed to the suspect Sp_1. His oldest crime was on 20-Jul-2009 at location (2.4815, 56.1600), his next crime on 10-Aug-2009 at location (2.1505, 55.2230) and his most recent crime on 29-Oct-2009 at location (3.3755, 55.4500). Figure 4 shows the corresponding trajectory.

However, the current map cube operator does not offer a function to achieve this. Note that this function must consider spatial (the crime location) and temporal (time of occurrence of the crime) aspects. The trajectory is then drawn from the oldest fact in the corresponding set to the most recent one.

3 Trajectory function

Consider again the example of Section 2. The incorporation of the Trajectory function to the map cube operator involves some problems. For example, consider the data shown in Table 3 and suppose the analyst wants the trajectory of the crimes of the suspect Sp_3.

Table 3: Sample 2 of our data warehouse of crimes.

Levels			Measures	
Day	Type_of _Crime	Suspect	# Victims	Location (Lat - Long)
20-Jul-2009	Robbery	Sp_3	2	{(3.0550, 55.3340)}
25-Jul-2009	Robbery	Sp_3	3	{(3.4000, 55.5650), (2.4715, 56.1535)}
30-Jul-2009	Robbery	Sp_3	1	{(3.3915, 57.0310)}

Note that Sp_3 starts his/her criminal trajectory on 20-Jul-2009 at location (3.0550, 55.3340), then on 25-Jul-2009 Sp_3 steals in two places: (3.4000, 55.5650) and (2.4715, 56.1535). Therefore, from the location (3.0550, 55.3340), the first place of the crimes of Sp_3, the trajectory can be drawn towards (3.4000, 55.5650), see Figure 5(a), or to (2.4715, 56.1535) see Figure 5(b).

To resolve this problem the date of occurrence of the facts could be handled with a lower granularity level, e.g., hours, minutes or seconds. Unfortunately, in real life applications the data are not always in this level of detail. Indeed, usually the date of occurrence of a crime is just an estimated value. As a consequence, we consider the following criteria:

1) a) **Randomness:** randomly choose (if the facts occurred on the same date) the fact that follows in the temporal sequence of the trajectory. We may also generate and show to the user all possible alternative trajectories. However, note that if there are many temporal ties, the number of alternative trajectories resulting could be very high.

2) b) **Proximity:** select within the set of facts with the same date, the one closest (in distance) to the fact preceding it temporarily. For example, the point of the first crime of Sp_3 (3.0550, 55.3340) is closer to the point (3.4000, 55.5650) than the point (2.4715, 56.1535). Therefore, considering this criterion, the resulting trajectory in our running example is shown in Figure 5 (b). If there are ties by proximity, the choice is random.

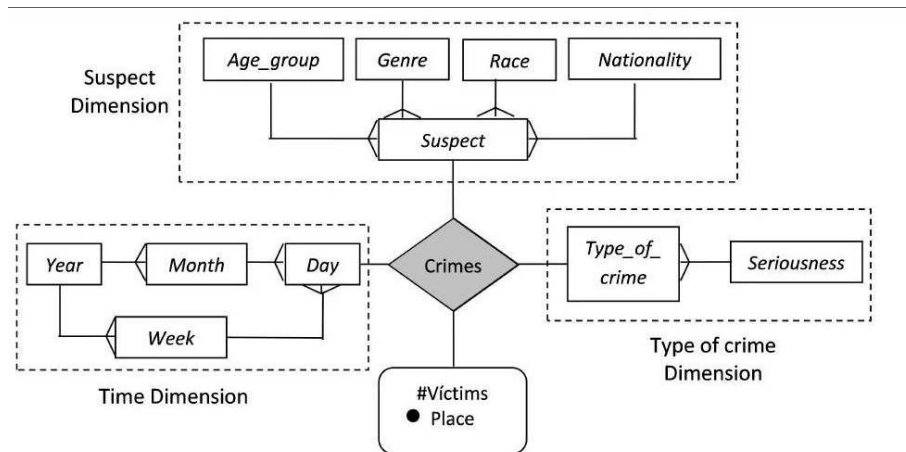3) c) **Center of mass:** consider the center of mass of the locations corresponding to the facts that occurred on

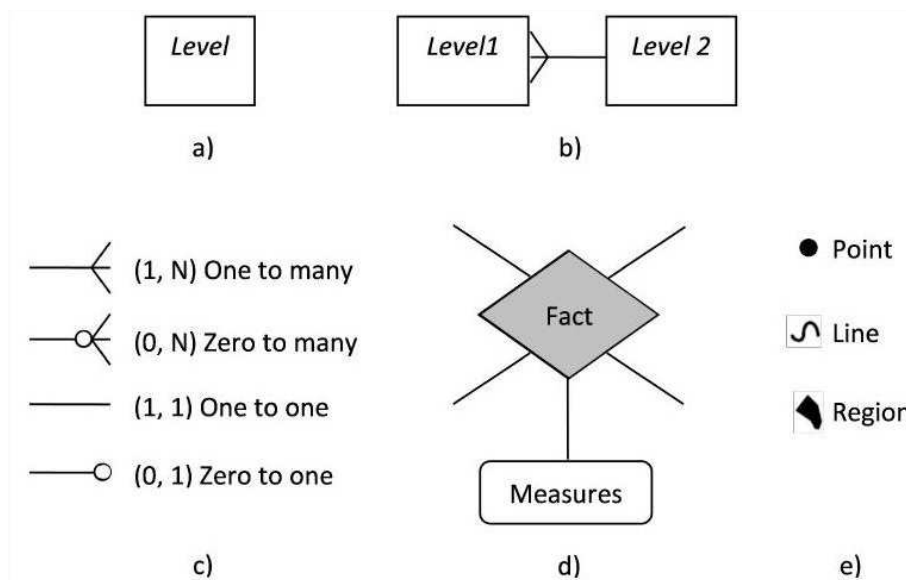**Fig. 1:** Conceptual model of a data warehouse of crimes.

Fig. 2: Symbols used to represent multidimensional models: a) level, b) hierarchy, c) cardinalities, d) fact, and e) geometries. Source (Malinowski and Zimanyi, 2008).

the same date. For example, the center of mass of the points (3.4000, 55.5650) and (2.4715, 56.1535) is (3.1427, 56.0612); therefore, the trajectory is plotted as shown in Figure 6.

The above problem is even more evident when an analyst wants to find the trajectory of all crimes in a given date (regardless of the type of crime or the suspect). Because all the places are associated with the same date, any of the above criteria is equally applicable. Note that if the alternative c) is applied, using the criterion of center of mass, the result would be a point.

According to the above analysis, the Trajectory function should provide the analysts the ability to choose the most appropriate criterion according to their needs. Center of mass is the default criterion of the Trajectory function.

Another issue to consider is that a fact can be associated with multiple temporal dimensions and several spatial measures. For example, consider the DW in Figure 7 and the data shown in Table 4.

In this example, a crime is associated with two dates: one represents the date of occurrence and the other one represents the date of recording in a police station. This
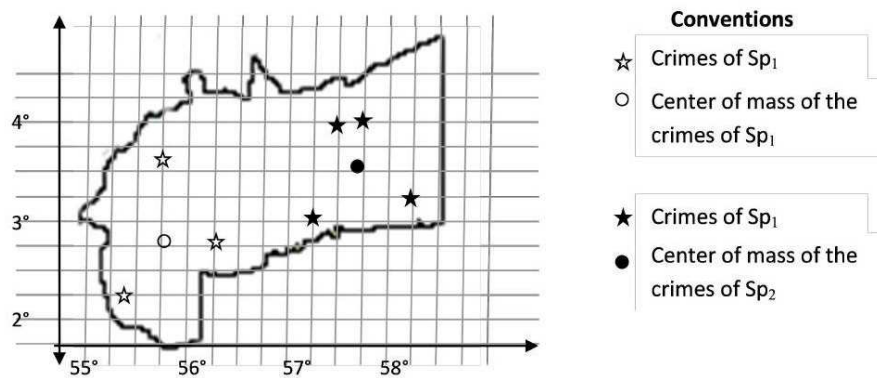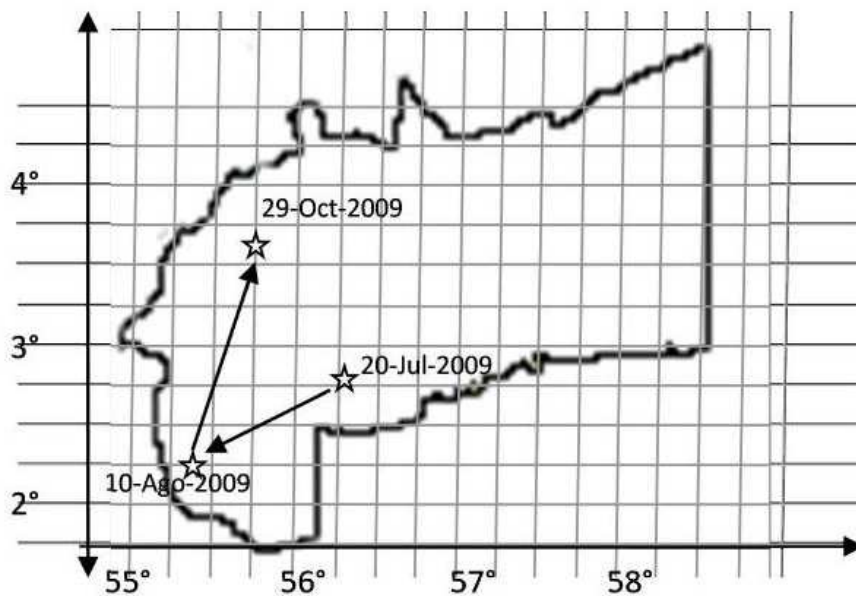
Fig. 3: Map for the suspect cuboid.

Fig. 4: Trajectory of crimes of the suspect Sp_1.

Table 4: Sample data from the data warehouse of crimes.

Levels				Measures	
Day of ocurrence	Day of recording	Type of Crime	Suspect	# Victims	Location (Lat - Long)
20-Jul-2009	22-Jul-2009	Robbery	Sp_1	1	{(2.4815, 56.1600)}
10-Aug-2009	11-Aug-2009	Murder	Sp_1	1	{(2.1505, 55.2230)}
29-Oct-2009	10-Nov-2009	Murder	Sp_1	2	{(3.3755, 55.4500)}
02-Jan-2009	02-Jan-2009	Rape	Sp_2	1	{(3.0505, 57.1500)}
30-Jan-2009	28-Feb-2009	Rape	Sp_2	1	{(3.1340, 58.1350)}
10-Feb-2009	15-Feb-2009	Robbery	Sp_2	3	{(4.0205, 57.4400)}

implies that in the Trajectory function the user must specify the temporal level and the spatial measure from which the corresponding trajectory will be generated. Note that in this DW, each fact represents the set of crimes (possibly a unit set) of the same type committed by the same suspect on the same date and also recorded in a police station on the same date.
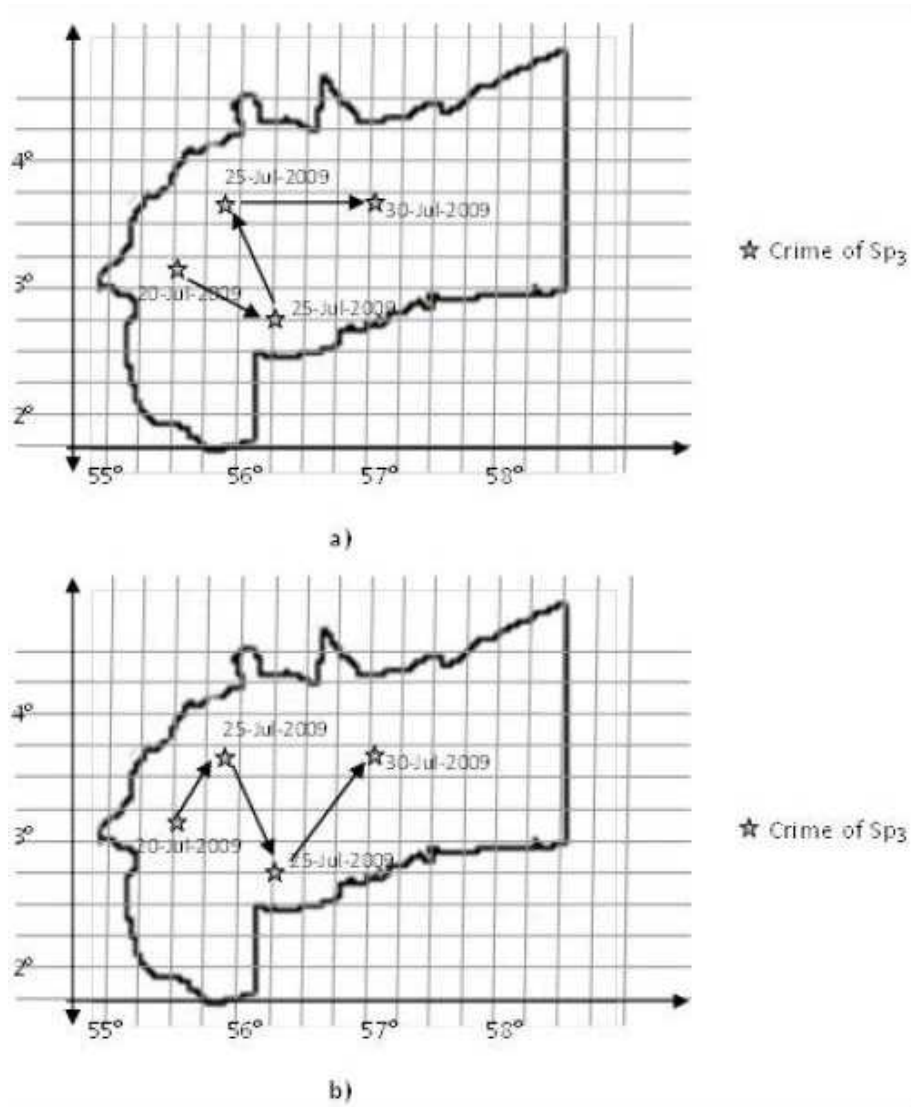
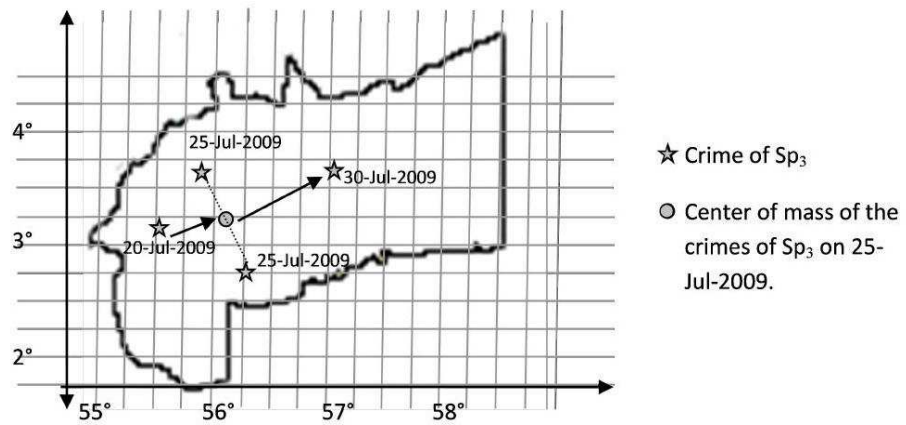**Fig. 5:** Two possible trajectories for crimes of suspect Sp3: a) trajectory 1, and b) trajectory 2.

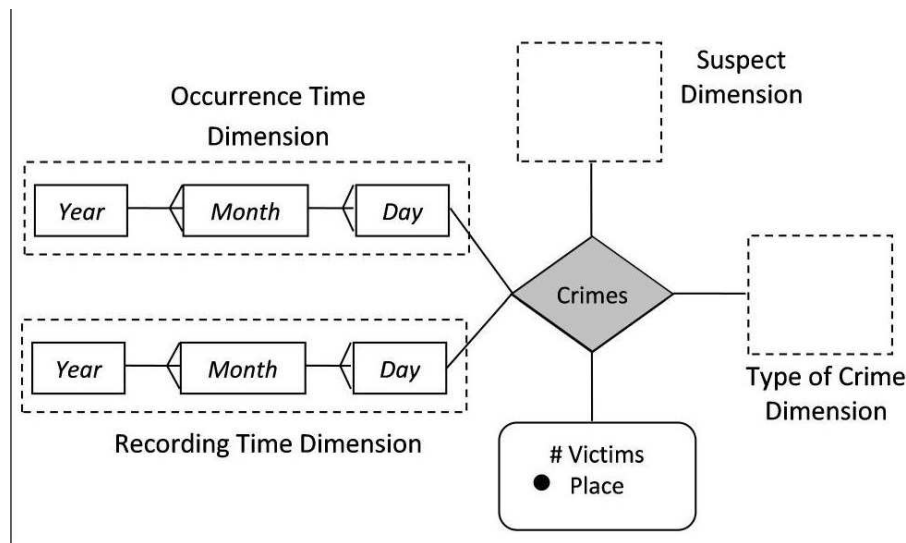**Fig. 6:** Center of mass criterion to break ties.

Fig. 7: Conceptual model of a data warehouse of crimes with two temporal dimensions.

Considering these aspects, we incorporate the Trajectory function as a new spatio-temporal aggregation function into the map cube operator. The Trajectory function receives as parameters a temporal level, a spatial measure, and a criterion to resolve the exposed problem on the plotting. The syntax of the function expressed in Backus-Naur notation [29] is:

$TRAJECTORY : (< time\_level >,$
$< spatial\_measure >) < tiebreakercriterion >$
$< time\_level > ::= < name >$
$< spatial\_measure > ::= < name >$
$< tiebreakercriterion > ::= NULL \,|\, CENTER$
$\_OF\_MASS \;|\; NEAREST\_NEIGHBOR \;|\; RANDOM \;|$
$ALL \,|\, < usertiebreakercriterion >$
$< usertiebreakercriterion > ::= < name >$
$<\quad name\quad > \qquad\qquad ::=$
$< letter > (< letter >|< digit >|< symbol >)*$
$< letter > ::= A \,|\, B \,|\, \ldots \,|\, Z \,|\, a \,|\, b \,|\, \ldots \,|\, z$
$< digit > ::= 0 \,|\, 1 \,|\, 2 \,|\, 3 \,|\, 4 \,|\, \ldots \,|\, 9$
$< symbol > ::= \# \,|\, \_ \,|\, .$

The possible values for criterion $< tiebreakercriterion >$ are:

- NULL: when the user does not specify the criterion. The default criterion is used.
- CENTER _ OF _ MASS: uses the center of mass criterion. It is the default criterion.
- NEAREST _ NEIGHBOR: uses the criterion of proximity.
- RANDOM: uses the criterion of randomness.
- ALL: generates all possible trajectories resulting from facts that occurred on the same date.
- $< usertiebreakercriterion >$: Allows the user to specify a personal criterion.

For example, the sentence which generates the results of Figure 5 (b) is:

Base Map	Crimes Map
Base Table	Crimes
Aggregate by	TRAJECTORY: (Day of occurrence, Location) NEAREST _ NEIGHBOR
Reclassify by	Suspect
Cartographic preference	Overlay: (Meta _ Map)

With regard to the execution time of the function, this will depend on the number of trajectories to plot (and the length of each one). This number depends on two factors: i) the number of cuboids, that is 2^n, as explained in Section 1 and ii) the number of trajectories that are plotted in each cuboid, which depend on the data in the fact table. For example, the previous sentence generates two cuboids: suspect and whole total. For the suspect cuboid, m trajectories are plotted; where m is the total number of different suspects. For the whole total cuboid, a single trajectory is drawn; which corresponds to the trajectory of crimes of all the suspects.

Note that the length and the number of segments of each trajectory also depend on the data in the fact table. For example, if a suspect in this table has five facts, then his/her trajectory will have four segments. In a real scenario, a data base of crimes from Pennsylvania was considered. Recurrent criminals were selected (20 in total with an average of 12 crimes per criminal) for a semester. With these values and an environment such as the described in Section 4, our results were obtained in less than 30 seconds, which is a reasonable time for users.

4 Experiments

Next, we present the essential components for the implementation of the Trajectory function. An HTML interface was designed and the Google Earth plugin was used to plot the resulting trajectories in a 3D environment. For this, we used the KML markup language based on XML to represent 3D geographical data (latitude, longitude, and altitude). MySQL 5.05 was used as database management system. The system architecture is shown in Figure 8. The experiments were run on a computer with operating system Windows 7 Home Premium 64-bit, Intel Core i5 2.40 GHz and 4.00 GB RAM.

From reports of [26] we analyzed crimes during a semester. The total number of crimes was 100. We considered four types of crimes and 20 criminals with an average rate of 12 crimes by criminal. Table 5 presents a sample of the analyzed data.

We obtained the trajectory for each type of crime, considering the date of occurrence. For this, we used the following map cube sentence:

Base Map	Crimes Map
Base Table	Crimes
Aggregate by	TRAJECTORY:
	(Day _ of _ occurrence, Location)
	RANDOM AS Traj_type_crime
Reclassify by	Type_of_crime
Cartographic preference	Overlay:
	(Pennsylvania_Map)

The results for the type of crime robbery are shown in Figure 9. It can be seen that most offenses occur in areas in the northeast and in the southeast of Pennsylvania.

We also obtained the trajectory for each criminal, considering the date of occurrence. For this, we used the following map cube sentence.

Base Map	Crimes Map
Base Table	Crimes
Aggregate by	TRAJECTORY:
	(Day _ of _ occurrence, Location)
	RANDOM AS Traj_suspetcs
Reclassify by	Suspect
Cartographic preference	Overlay:
	(Pennsylvania_Map)

The results for the criminal Schuyler Crossman are shown in Figure 10. It can be seen that the criminal has two areas where he repeats offenses, the Central and the Northeast area where he committed about 75% of his crimes. This information could help establish the *modus operandi* of criminals and draw a tracing network around certain areas.

Table 6 presents some consolidated results. For example, in the case of criminals, (1) indicates that if we consider a region with a radius of 500 m around a point, the criminal commits three crimes on average in this region, in probabilistic terms the value was 0.27, see (2), because the crime rate was 12 per criminal. (3) indicates that the next crime of a criminal will be on average in a region of radius 2.3 km. Analogous results were obtained for the case of the types of crimes.

Below is a part of the KML code generated by the prototype, in particular the data of the corresponding trajectory of the type of crime robbery from previous example (Figure 9).

$< LineString >$
$< extrude > 1 < /extrude >$
$< tessellate > 1 < /tessellate >$
$< altitudeMode > absolute < /altitudeMode >$
$< coordinates >$
$38.0337, -76.3356, 1500$
$39.5713, -75.1140, 1500$
$39.1916, -76.4339, 1500$
$39.1916, -76.4339, 1500$
$\dots$
$< /coordinates >$
$< /LineString >$

The label $< LineString >$ specifies a straight line; the coordinates of its points (latitude, longitude, and altitude) are specified in the label $< coordinates >$. The value 1 (Boolean) of the label $< extrude >$ indicates that the vertices of the line are connected to earth (ground). The value 1 (Boolean) of the label $< tesellate >$ indicates that the line fits the curvature of the earth (and thus avoiding some of the segments of the line are hidden under the ground). The label $< altitudeMode >$ specifies how to interpret altitude; the value absolute indicates values in meters above sea level (1500 m in the shown code). Other options and further details on these labels are explained in [28].

5 Conclusions and future work.

In this paper, we extended the functionality of the map cube operator with the incorporation of a function for plotting the trajectory followed by a set of facts. Because several facts in the set may have occurred on the same date, this may lead to different trajectories. To resolve this problem we proposed four criteria: NEAREST_NEIGHBOR, RANDOMNESS, CENTER_OF_MASS, and ALL. We presented the syntax of the function and we analyzed the behavior of each of these criteria. In addition, the possibility for the user to specify its own criteria in the grammar of the function was left open.
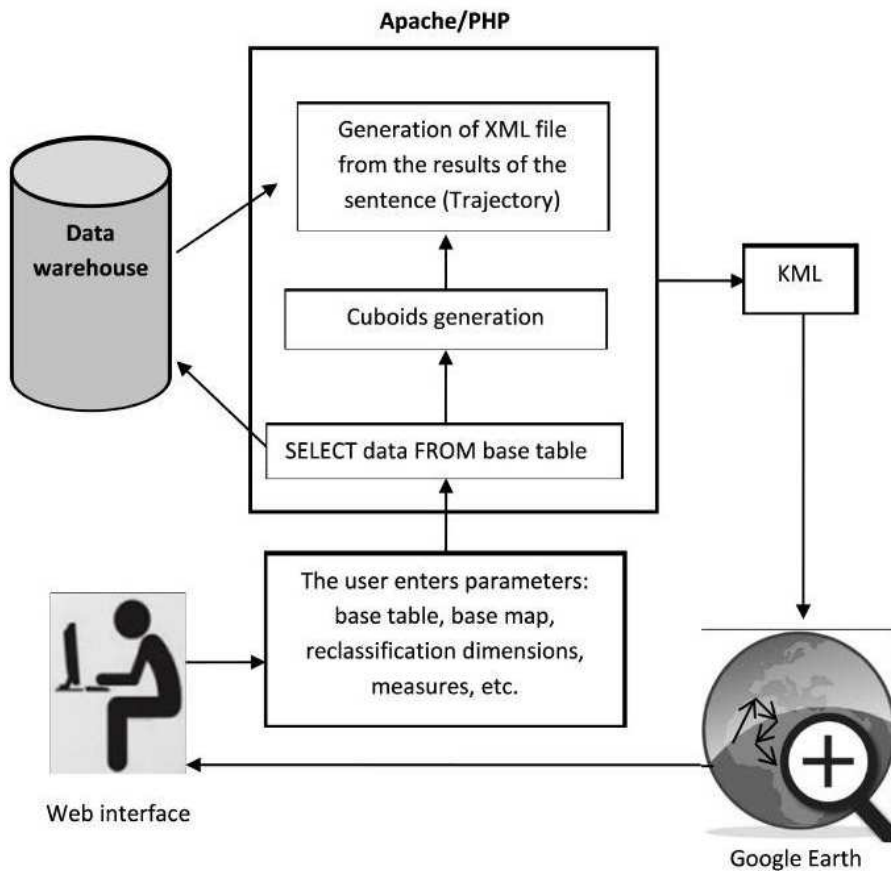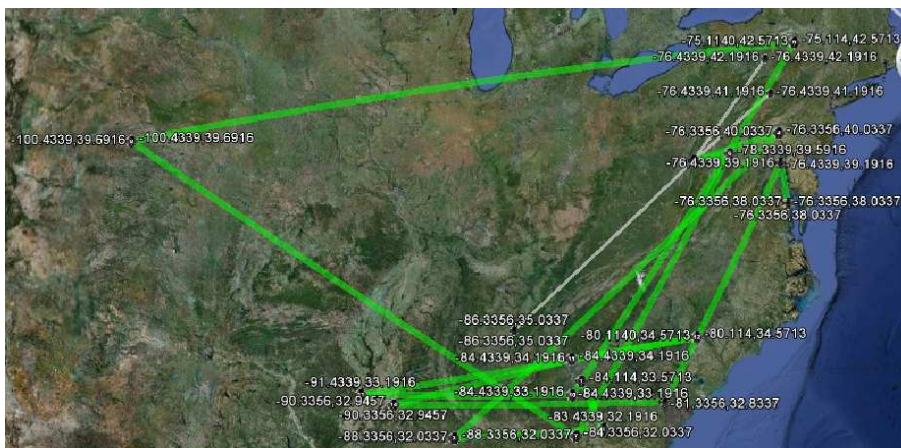
Fig. 8: System Architecture.

Fig. 9: Results in Google Earth for the map cube sentence: Trajectory of Type _ of _ crime cuboid, type = "Robbery ". Conventions: Green Line (robbery) and gray (Tiebreaker route for robberies on the same date).

For the layout of the trajectories, Google Earth and KML were used, allowing its visualization in a real context. To validate and show the expediency of our proposal, we presented a case study about crimes in Pennsylvania.

Table 5: Sample data from the data warehouse of the DPSUP crimes.

Levels				Measures
Day _ of _ ocurrence	Day _ of _ recording	Type _ of _ Crime	Suspect	Location (Lat - Long)
20-Jan-2010	09-Jun-2010	Robbery	Schuyler Crossman	4100 LOCUST ST (38.0337, -76.3356)
20-Jan-2010	09-Jun-2010	Other offenses	Schuyler Crossman	4100 LOCUST ST (38.0337, -76.3356)
08-Jun-2010	08-Jun-2010	Driving under the influence	Darryl Dove	3900 BALTIMORE (38.5902, -78.5615)
10-Jun-2010	09-Jun-2010	Driving under the influence	Farise Overton	3300 SOUTH ST (39.1320, -76.4213)
07-Jun-2010	07-Jun-2010	Other offenses	Schuyler Crossman	4006 MARKET ST (39.0017, -76.5414)
08-Jun-2010	08-Jun-2010	Robbery	Lajoy Redmon	3651 WALNUT ST (39.1916, -76.4339) (39.1916, -76.4339)
05-Jun-2010	05-Jun-2010	Robbery	Candice Copes	3651 WALNUT ST (39.1916, -76.4339) (39.1916, -76.4339)
04-Jun-2010	04-Jun-2010	Robbery	Candice Copes	3600 SANSOM ST (39.5713,-75.1140) (39.5713, -75.1140)
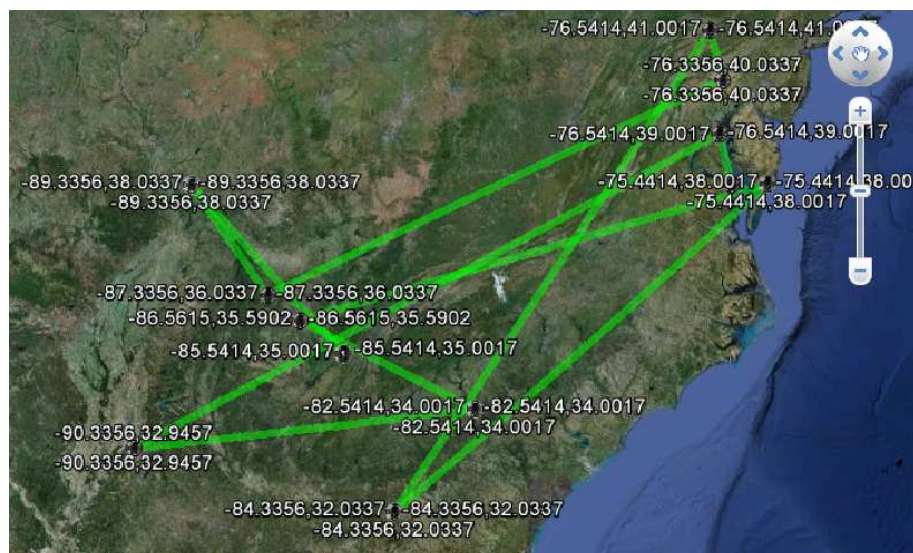
Fig. 10: Results in Google Earth for the map cube sentence: Trajectory of Suspect cuboid, Suspect = "Schuyler Crossman".

Table 6: Experimental results.

Grouping element	(1) Average number of reoffendings in a region (500 m radius)	(2) Average probability of committing a crime in a region (radius 500 m)	(3) Average distance between a crime and the next one (according to their order of occurrence)
Suspect	3	0.27	2.3 km
Type of crime	7	0.35	0.8 km

The tracing of trajectories considering obstacles is planned for future work. The current connecting points of a trajectory generate straight lines. However, some trajectories may cross streets and mountains, which may

be unrealistic in certain domains, e.g., for land vehicle traffic. To do this, the user may specify a set of constraints or the function might try to discover them. Furthermore, we could consider complex issues as follows: given two points and a set of obstacles between them, what is the shortest route to connect these points without crossing barriers? A starting point is to use the taxicab geometry [30]. We also plan to incorporate a function such that given the trajectory of an object, find the set of facts in a DW that form a trajectory that best fits the given trajectory, and present its degree of similarity. This should consider aspects of similarity in shape, time of occurrence, orientation, among others [31]. Finally, we plan to incorporate a function that based on descriptive data sets, it detects facts that may form a trajectory belonging to a single object or individual. For example, based on the descriptions of the crimes, obtain trajectories that suggest the same *modus operandi*.

References

[1] W. Inmon, Building the data warehouse. Wiley, New York, (2005).

[2] M. Golfarelli and S. Rizzi, Data warehouse design: modern principles and methodologies. McGraw-Hill Osborne Media, New York, (2009).

[3] R. Kimball, M. Ross, W. Thornthwaite, J. Mundy and B. Becker, The data warehouse lifecycle toolkit. Wiley, New York, (2008).

[4] R. Laberge, The data warehouse mentor: Practical data warehouse and business intelligence insights, McGraw Hill Osborne Media, New York, (2011).

[5] S. Shekhar, C. Lu, X. Tan, and S. Chawla, Map cube: A visualization tool for spatial data warehouses. H. Miller and J. Han, (Eds), Geographic data mining and knowledge discovery. Taylor and Francis, New York, (2001).

[6] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. Naughton, R. Ramakrishnan and S. Sarawagi, On the computation of multidimensional aggregates. 22nd International Conference on Very Large Data Bases (VLDB). Mumbai, India, (1996).

[7] E. Malinowski, and E. Zimanyi, Advanced data warehouse design: from conventional to spatial and temporal applications, Springer, New York, (2008).

[8] F. Moreno, F. Arango and R. Fileto, Extending the map cube operator with multiple spatial aggregate functions and map overlay. 17th International Conference on Geoinformatics (Geoinformatics). Fairfax, USA, (2009).

[9] C. T. Lu, Y. Kou, H. Wang, S. Shekhar, P. Zhang and R. Liu, Two web-based spatial data visualization and mining systems: mapcube & mapview, International Workshop on Next Generation Geospatial Information, Cambridge, USA, (2003).

[10] R. Santos Mello, R. Fileto, C. Dorneles and V. Bogorny, Research on complex data management and analysis at UFSC. Journal of Information and Data Management, **2**, 2 (2011).

[11] N. Prat, J. Akoka , I. Comyn-Wattiau, Transforming multidimensional models into OWL-DL ontologies, IEEE Sixth International Conference on Research Challenges in Information Science, Valencia, Spain, (2012).

[12] Map cube, http://www.spatial.cs.umn.edu/mapcube.htm

[13] O. Baltzer, F. Dehne, S. Hambrusch and A. Rau-Chaplin, OLAP for trajectories. 19th International Conference Database and Expert Systems Applications (DEXA). Turin, Italy, (2008).

[14] L. Leonardi, G. Marketos, E. Frentzos, N. Giatrakos, S. Orlando, N. Pelekis, A. Raffaeta, A. Roncato, C. Silvestri, and Y. Theodoridis, T-Warehouse: Visual OLAP analysis on trajectory data. 26th International Conference on Data Engineering (ICDE). Long Beach, USA, (2010).

[15] G. Marketos, Y. Theodoridis, Ad-hoc OLAP on Trajectory Data. 11th International Conference on Mobile Data Management (MDM). Kansas City, USA, (2010).

[16] U. Demsar, Geovisual analytics for trajectories-3D kernel density. 2nd Seminar on Representation, analysis and visualization of moving objects. Dagstuhl, Germany, (2010).

[17] S. Bakshev, J. de Macedo, L. Spisanti, M. Casanova, V. Vidal and C. Vidal Semantic visualization of trajectories. 13th International Conference on Enterprise Information Systems (ICEIS). Beijing, China, (2011).

[18] D. Biadgilgn, C. Blok and O. Huisman, Assessing the cartographic visualization of moving objects. Momona Ethiopian Journal of Science, **3**, 22, (2011).

[19] J. Sack, B. Speckmann, E. Van Loo and R. Weibel, Abstracts collection: Representation, analysis and visualization of moving objects. 2nd Seminar on Representation, analysis and visualization of moving objects. Dagstuhl, Germany, (2011).

[20] ArcGIS, http://www.arcgis.com

[21] Kosmo, http://www.opengis.es

[22] Capaware, http://www.capaware.org

[23] S. Rivest, Y. Bedard and P. Marchand, Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP), Geomatica, **55**, 4 (2001).

[24] A. Vaisman, E. Zimanyi, What is spatio-temporal data warehousing? 11th International Conference on Data Warehousing and Knowledge Discovery (DaWaK) Linz, Austria, (2009).

[25] G. Pestana and M. Mira da Silva, Multidimensional modeling based on spatial, temporal and spatio-temporal stereotypes, 25th ESRI International User Conference (ESRI), San Diego, USA, (2005).

[26] DPSUP. Division of Public Safety University of Pennsylvania, http://www.publicsafety.upenn.edu.

[27] Google Earth,http://earth.google.com

[28] KML., http://code.google.com/intl/es-ES/apis/kml/documentation

[29] D. Grune, Parsing techniques: A practical guide, Springer, New York, (2010).

[30] E. Krause, Taxicab Geometry. Dover, New York, (1987).

[31] N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos and y Y. Theodoridis, Mining trajectory databases via a suite of distance operators. 23rd International Conference on Data Engineering (ICDE). Istanbul, Turkey, (2007).

F. J. M. Arboleda holds a PhD in Engineering Systems from Universidad Nacional de Colombia. He currently works as an associate professor at Universidad Nacional de Colombia. His research areas are datawarehouses and spatio-temporal databases.

Juan Camilo Alzate holds a Bachelor degree in Systems Engineering from Universidad Nacional de Colombia. He is currently pursuing his MSc. in Engineering at Universidad de Antioquia, Colombia and works as assistant professor in the same university. His research areas are datawarehouses and spatio-temporal databases.

J. A. H. Riveros earned his Ph.D. Riveros with honors from the University of A Coruña, Spain. He has served as a consultant, researcher and professor. He has developed research projects with UNESCO and other institutions. Author of diverse scientific publications. Nowadays, he is a professor at the Faculty of Engineering, Universidad Nacional de Colombia, Medellín. Computational intelligence in engineering and modeling of multi-domain systems are some of his areas of interest.