OLAP for Trajectories

Oliver Baltzer[1], Frank Dehne[2],
Susanne Hambrusch[3], and Andrew Rau-Chaplin[1]

[1] Dalhousie University, Halifax, Canada
obaltzer@cs.dal.ca, arc@cs.dal.ca
http://www.cs.dal.ca/~arc
[2] Carleton University, Ottawa, Canada
frank@dehne.net
http://www.dehne.net
[3] Purdue University, West Lafayette, IN, USA
seh@cs.purdue.edu
http://www.cs.purdue.edu/people/faculty/seh/

Abstract. In this paper, we present an OLAP framework for trajectories of moving objects. We introduce a new operator GROUP_TRAJECTORIES for group-by operations on trajectories and present three implementation alternatives for computing groups of trajectories for group-by aggregation: *group by overlap*, *group by intersection*, and *group by overlap and intersection*. We also present an interactive OLAP environment for resolution drill-down/roll-up on sets of trajectories and parameter browsing. Using generated and real life moving data sets, we evaluate the performance of our GROUP_TRAJECTORIES operator. An implementation of our new interactive OLAP environment for trajectories can be accessed at http://OLAP-T.cgmlab.org.

1 Introduction

Global positioning (GPS) and RFID systems are creating vast amounts of spatio-temporal data for *moving objects*. Consider N moving objects on a 2D spatial grid. Each object is identified by a unique *tag* number (similar to EPC in RFID). Object movements are recorded through a set of readings $((x, y), i, t)$ indicating that object (tag) i was detected at time t within the grid cell located at (x, y). The N moving objects are represented by a relational table *objects* with N records. Each record contains values *tag, name, size, color,* etc. describing one object according to a star schema. Among them is a value *trajectory* representing the movement of the respective object as a sequence $[(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots (x_m, y_m, t_m)]$ of positions at time $t = t_1, t_2, \ldots t_m$. In order to efficiently *analyze* large scale data sets representing moving objects, it is important to have available the well established set of tools for OLAP analysis. In order to apply OLAP tools towards moving object datasets, it is necessary to aggregate with respect to *trajectory* as a *feature* dimension as well as a *measure* dimension.

We illustrate this with the example shown in Figure 1. Consider the trajectories shown in Figure 1a. We observe a number of individual objects that move

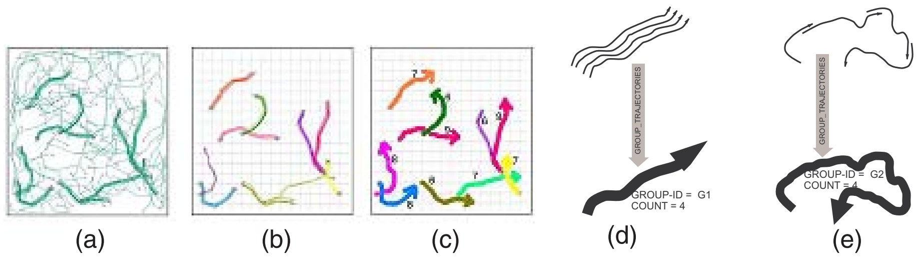

Fig. 1. *OLAP For Trajectories* Example. (a) Input data. (b) Groups with minimum support. (c) Aggregate results reported (aggregate trajectories and counts). (d) Illustration of operator GROUP_TRAJECTORIES:Group by Intersection. (e) Illustration of operator GROUP_TRAJECTORIES:Group by Overlap.

on random paths plus 10 *groups* of objects that move together on similar paths. Each group consists of more then five objects moving on similar paths which, taken together, appear to the human eye as "bold" paths.

Consider the following SQL query where *trajectory* is both, a *feature* dimension as well as a *measure* dimension:

```
SELECT AGGREGATE(trajectory) AS trajectory
       COUNT(trajectory) as count
FROM objects
GROUP BY GROUP_TRAJECTORIES(trajectory,
         resolution)
HAVING COUNT(*) >= 5
```

For this example, the aim of the GROUP BY operation with respect to *feature* dimension *trajectory* is to group similar trajectories and eliminate groups with less than minimum support (less than 5 similar trajectories). The resulting set of groups is shown in Figure 1b. Once the groups of trajectories have been determined, we report for each group an *aggregate trajectory* representing the trajectories in the group. In this example, the aggregate trajectory is the average trajectory computed by calculating for each time t_i the average of the locations (x_i, y_i) of the trajectories in the group. The result is shown in Figure 1c, where each group is represented by the aggregate trajectory and size of the group (count).

The goal of *OLAP analysis for trajectories* is to answer aggregate queries with respect to the spatial movements of a set of objects represented in a relational table *objects*. The main problem arising is how to aggregate with respect to feature dimension *trajectory*. It is very unlikely that any two trajectories are exactly the same. Hence, standard aggregation of records with equivalent *trajectory* values is not very useful in most cases. We propose to partition the given trajectories into disjoint *groups* of trajectories using a new operator which we term GROUP_-TRAJECTORIES. This operator returns for each trajectory a *group identifier*, and then OLAP can proceed with standard aggregation according to the group identifiers instead of the trajectories themselves.

The main problem addressed in this paper is how to define and compute the operator GROUP_TRAJECTORIES such that the resulting groups allow for a meaningful analysis of object movements via OLAP. We propose *three different versions* of the operator GROUP_TRAJECTORIES which compute groups of trajectories that are appropriate for OLAP analysis of trajectories for different circumstances and applications: *Group by Overlap*, *Group by Intersection* and *Group by Overlap and Intersection*.

Section 3 will show in detail how these three different versions of our GROUP_-TRAJECTORIES operator are defined and computed. Our *Group by Intersection* method aggregates subsets of trajectories that correspond to similar or synchronous movements; see Figure 1d. Our *Group by Overlap* method aggregates subsets of trajectories that correspond to sequences of movements with sufficient overlap between subsequent trajectories; see in Figure 1e. The *Group by Overlap and Intersection* method aggregates subsets of trajectories that correspond to a combination of sequences of movements and similar or synchronous movements.

In Section 4, we present an *interactive OLAP environment* for the analysis of trajectories that allows resolution drill-down and roll-up as well as parameter browsing. An experimental evaluation is outlined in Section 5. An implementation of our new interactive OLAP environment for trajectories can be accessed at `http://OLAP-T.cgmlab.org`.

2 Related Work

There is a wealth of literature on spatiotemporal data analysis and aggregation. See e.g. [9] for a survey. This work studies aggregation by specific temporal dimensions such as "by day" or "by year", or by strict topological association such as "by location square" or "within 10 km of" (e.g. [11]). In our case, we wish to aggregate entire trajectories. For the detection of relationships among trajectories in a moving object database we found in the literature five groups of approaches: variations of frequent pattern or association rule mining (e.g. [4,5,6,16]), clustering techniques (e.g. [8,12]), Computational Geometry techniques (e.g [7]), neural network based techniques (e.g. [15]), and edit distance, warping techniques and longest common subsequence (LCSS) extraction (e.g. [13,14,17,18]). A comparison of our work with these approaches is omitted due to page restrictions. It can be found in the extended version of this paper [2].

3 Computing Groups of Trajectories

In this section we present three different implementations of the operator GROUP_-TRAJECTORIES which compute groups of trajectories that are appropriate for OLAP analysis of trajectories for different circumstances and applications: Group by Overlap, Group by Intersection, and Group by Overlap and Intersection. We first apply a time and space resolution mapping of our initial set $\mathcal{T}$ of trajectories. This allows for the resolution drill-down and roll-up within our interactive OLAP framework for trajectories to be discussed in Section 4. Next, we compute frequent itemsets for the mapped set of trajectories and then apply a reverse mapping step. Here, we determine for each frequent itemset f, the corresponding *original* group c of trajectories and create a set $\mathcal{C}$ of resulting (f, c) pairs. A more detailed presentation of our method is contained in the extended version of this paper [2].

The most important part of our method is the group merging phase. In this paper, we present three different methods: (a) Group by Overlap (Sections 3.1), (b) Group by Intersection (3.2), and (c) Group by Overlap and Intersection (3.3).

3.1 Group by Overlap

Our *Group By Overlap* method introduces a tunable parameter *overlap ratio threshold ORT* which controls the strength of the grouping process. The interactive OLAP framework for trajectories discussed in Section 4 will allow for an interactive tuning of this parameter.

Our *Group By Overlap* method is based on an *overlap graph* Γ, where each vertex corresponds to a trajectory. For each frequent item set f and corresponding set c of trajectories, we consider all pairs of trajectories $t_i, t_j \in c$ and add for each pair an edge (t_i, t_j) with label *overlap ratio* $OS = \frac{2\cdot|f|}{|t_i|+|t_j|}$. The *overlap ratio* measures the size of the overlap relative to the sizes of the trajectories. We then remove all edges where the *overlap ratio* OS is smaller than the chosen *overlap ratio threshold* ORT and compute the connected components of the remaining graph. These components correspond to the groups of trajectories that are reported. A more detailed presentation is contained in the extended version of this paper [2].

The nature of the obtained groups of trajectories is determined by two factors. (1) The *overlap ratio threshold* ORT determines how much two neighboring trajectories within a group have to overlap. (2) The graph connected component construction allows for an "adding up" of trajectories corresponding to a "relay" type of movement. Depending on the chosen *overlap ratio threshold* ORT, the "relay" parties will have to move in unison for more or less of their own individual movements.

3.2 Group by Intersection

Our *Group By Intersection* method introduces a tunable parameter *intersection ratio threshold IRT* which controls the strength of the grouping process. The interactive OLAP framework for trajectories discussed in Section 4 will allow for an interactive tuning of this parameter.

Our *Group By Intersection* method first creates an initial set $\mathcal{G}$ of groups of trajectories, where each group c corresponds to a frequent itemset f determined in the reverse matching in Section 3. Each group c is assigned a *group strength* $GS(c)$ which is initially set to the size of the respective frequent itemset. The remainder of our method merges groups in $\mathcal{G}$ by iterating the following loop. We compute for each pair $g_i, g_j \in \mathcal{G}$ a value *intersection ratio* $AS(g_i \cup g_j) = \min\left(\frac{|g_i \cap g_j|}{|g_1|}, \frac{|g_i \cap g_j|}{|g_2|}\right)$ which represents the number of trajectories that occur in both g_i and g_j, relative to the sizes of g_i and g_j. We will consider as candidates for merging all pairs g_i, g_j whose intersection ratio is larger than our input parameter *intersection ratio threshold* IRT and compute for each such pair a value *merge strength* $MS(g_i \cup g_j) = \frac{GS(g_i)+GS(g_j)}{2}$ which is the average of their *group strength* values. All candidate pairs are ranked by their *merge strength* and we will merge the pair g_i^*, g_j^* with maximum merge strength, or one of the maximal pairs if there are multiple. The *group strength* $GS(g_{i^*} \cup g_{j^*})$ of the new merged group will be the *merge strength* $MS(g_i^* \cup g_j^*)$. This process is repeated until there are no more pairs of groups with non zero *merge strength*, that is, until there are no more pairs of groups with *intersection ratio* larger than the *intersection ratio threshold* IRT. A more detailed presentation of our method is contained in the extended version of this paper [2].

Our *Group by Intersection* method aggregates subsets of trajectories that correspond to "marching band" style parallel movements. The nature of the obtained groups of trajectories is determines by two factors. (1) The *intersection ratio threshold* IRT determines how many shared trajectories between two groups are "sufficient" for them to be merged. (2) The merging process which is similar in nature to a minimum spanning tree calculation. We merge first the largest groups with sufficient shared trajectories and then work our way down to the smaller groups. Unlike the *Group by Overlap* method which combines sequences of movements, the *Group by Intersection* method combines parallel of movements.

3.3 Group by Intersection and Overlap

The goal of our *Group by Intersection and Overlap* method is to group both, sequences of movements and parallel movements. It is a combination of our methods in Sections 3.1 and 3.2. We create the same set $\mathcal{G}'$ of groups of trajectories as in Section 3.2 and the same overlap graph Γ as in Section 3.1. Then we add to Γ a clique for each $g \in \mathcal{G}'$ (i.e. edges between all pairs of trajectories $t_1, t_2 \in g$) and compute the connected components of the modified graph Γ. Each connected component corresponds to a group of trajectories.

The resulting groups are sequences of overlapping trajectories as in our *Group by Overlap* method to which we add parallel trajectories as in our *Group by Intersection* method. The aggregation is guided by two parameters, the *intersection strength threshold* IRT and the *overlap ratio threshold* ORT, which control the width and length, respectively, of the generated groups.

4 Interactive OLAP for Trajectories

The algorithms for the three different versions of operator GROUP_TRAJECTORIES presented in Section 3 are guided by the following parameters: space resolution, time resolution, minimum support, intersection ratio threshold and overlap ratio threshold. This allows to analyze groups of trajectories for various levels of resolution or connectedness, and provides another opportunity for

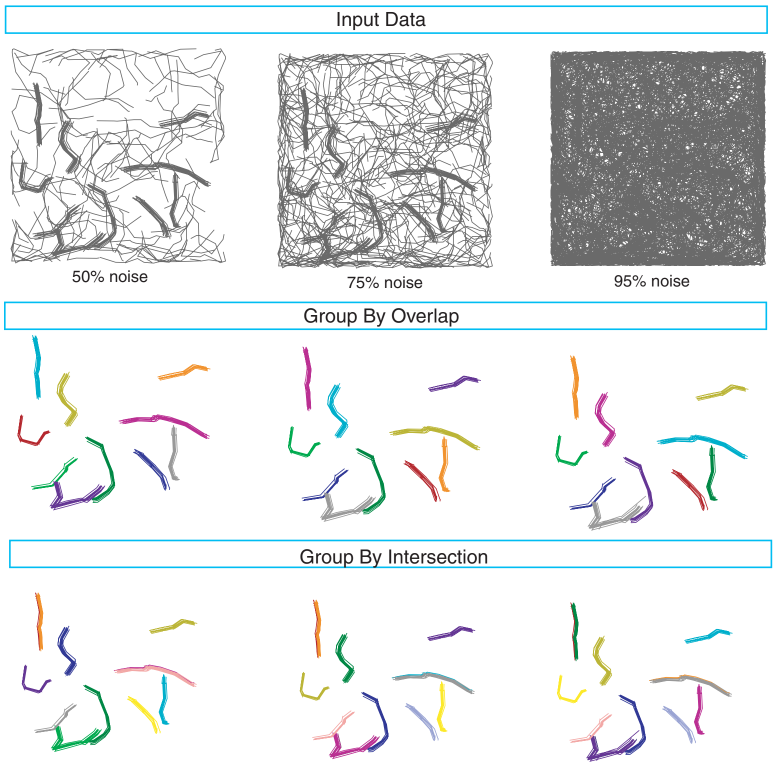

Fig. 2. Test of robustness against noise. Top row: input data consisting of 10 groups with 10 similar trajectories each and three levels of noise: 50%, 75% and 95%. Center row: Groups computed by GROUP_TRAJECTORIES: *Group By Overlap* ($ORT = 0.5$, $min\_support = 4$). Bottom row: Groups computed by GROUP_TRAJECTORIES: *Group By Intersection* ($IRT = 0.5$, $min\_support = 4$). Groups are identified by color (*group identifier* = color).

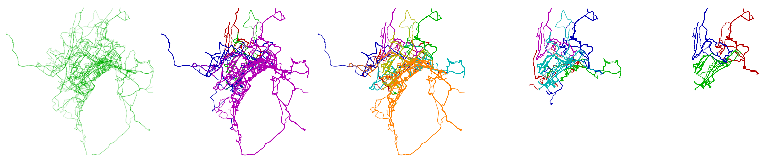

Fig. 3. School Buses Dataset and Groups reported (identified by color) using *Group by Overlap* and $ORT = 0.4, 0.5, 0.6, 0.7$, respectively ($min\_support = 5$, $min\_length = 30$)

OLAP analysis of trajectories. For example, for a high level analysis of GPS data for the movement of a fleet of ships, time granularity "day" may be sufficient. However, a drill-down to viewing the paths taken by a group of ships when entering a port may require a time granularity "minute". As an example for browsing a parameter like *overlap ratio threshold*, consider a set of trajectories representing movements of people who pass on a disease virus. The aggregate, using our *Group by Overlap* method, could be used to analyze the total movement of the virus. In this example, our parameter *overlap ratio threshold* would represent the amount of interaction between individuals required to pass on the virus. Changing the threshold value allows to evaluate how far the virus will spread based on different assumption about its transmission.

We have built a prototype *interactive environment for the analysis of trajectories* that allows resolution drill-down and roll-up as well as parameter browsing. It can be accessed at `http://OLAP-T.cgmlab.org`.

5 Experimental Evaluation

Our *Group by Overlap* and *Group by Intersection* methods have a surprising resilience against background noise. On the example shown in Figures 2, as well as many other examples that we tested, they have no trouble reporting the correct result for noise levels of 50%, 75% and even as high as 95%. At a noise level of 95%, the human eye can no longer visually detect the original groups of parallel paths but our methods have no problem reporting the correct result.

For the evaluation of our methods on real world data, we have chosen the school buses dataset that can be freely obtained from [1]. The dataset contains 145 trajectories of buses that are moving in and around an urban area. Due to page restrictions, we can not show the dataset here. It can be viewed by going to http://OLAP-T.cgmlab.org and selecting the dataset "buses".

Frequent itemsets mining without aggregation, as e.g. in [4,10,3] (plus a minimum length cutoff as used in our methods), would result in 76 groups being identified. This large number of groups reported by frequent itemsets mining based methods is often a disadvantage because it does not lead to signifficant aggregation in an OLAP setting. Figure 3 shows the results obtained with our *Group by Overlap* method for *ORT* values 0.4, 0.5, 0.6, and 0.7. We observe that the parameter *ORT* in our *Group by Overlap* method allows for a much finer control over the grouping of trajectories reported and that the *Group by Overlap* method reports a considerably smaller number of groups.

A more detailed presentation of experimental results for our method is contained in the extended version of this paper [2].

References

1. R-tree Portal (Last accessed, November 16, 2007), `http://www.rtreeportal.org/`
2. Baltzer, O., Dehne, F., Hambrusch, S., Rau-Chaplin, A.: Olap for trajectories. Technical Report TR-08-11, School of Computer Science, Carleton University, `http://www.scs.carleton.ca`
3. Cao, H., Mamoulis, N., Cheung, D.W.: Mining frequent spatio-temporal sequential patterns. icdm, 82–89 (2005)

4. Gidófalvi, G., Pedersen, T.B.: Mining Long, Sharable Patterns in Trajectories of Moving Objects. In: STDBM 2006: Proceedings of the 3rd Workshop on Spatio-Temporal Database Management (2006)
5. Hwang, S.Y., Liu, Y.H., Chiu, J.K., Lim, E.P.: Mining mobile group patterns: A trajectory-based approach. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 713–718. Springer, Heidelberg (2005)
6. Kim, D., Kang, H., Hong, D., Yun, J., Han, K.: STMPE: An Efficient Movement Pattern Extraction Algorithm for Spatio-temporal Data Mining. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3981, pp. 259–269. Springer, Heidelberg (2006)
7. Laube, P., van Kreveld, M., Imfeld, S.: Finding REMO–detecting relative motion patterns in geospatial lifelines. In: Developments in Spatial Data Handling: Proceedings of the 11th International Symposium on Spatial Data Handling, pp. 201–214 (2004)
8. Li, Y., Han, J., Yang, J.: Clustering moving objects. In: KDD 2004: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 617–622. ACM, New York (2004)
9. López, I.F.V., Snodgrass, R.T., Moon, B.: Spatiotemporal Aggregate Computation: A Survey. IEEE Transactions on Knowledge and Data Engineering 17(2), 271–286 (2005)
10. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.W.: Mining, indexing, and querying historical spatiotemporal data. In: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 236–245 (2004)
11. Marchand, P., Brisebois, A., Bédard, Y., Edwards, G.: Implementation and evaluation of a hypercube-based method for spatiotemporal exploration and analysis. ISPRS Journal of Photogrammetry and Remote Sensing 59(1-2), 6–20 (2004)
12. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. 27(3), 267–289 (2006)
13. Sclaroff, S., Kollios, G., Betke, M.: Motion mining: discovering spatio-temporal patterns in databases of human motion. In: Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (2001)
14. Shim, C.B., Chang, J.W.: A new similar trajectory retrieval scheme using k-warping distance algorithm for moving objects. In: Dong, G., Tang, C.-j., Wang, W. (eds.) WAIM 2003. LNCS, vol. 2762, pp. 433–444. Springer, Heidelberg (2003)
15. Sumpter, N., Bulpitt, A.: Learning spatio-temporal patterns for predicting object behaviour (1998)
16. Verhein, F., Chawla, S.: Mining spatio-temporal patterns in object mobility databases. Data Mining and Knowledge Discovery (2007)
17. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings. 18th International Conference on Data Engineering, 2002, pp. 673–684 (2002)
18. Zeinalipour-Yazti, D., Lin, S., Gunopulos, D.: Distributed spatio-temporal similarity search. In: CIKM 2006: Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 14–23. ACM, New York (2006)