MULTITEMPORAL GEOSPATIAL QUERY GROUPING USING CORRELATION SIGNATURES

Giorgos Mountrakis Peggy Agouris Anthony Stefanidis

Department of Spatial Information Science and Engineering
University of Maine, Orono, ME, 04469-5711, USA
{giorgos, peggy, tony}@spatial.maine.edu

ABSTRACT

With recent advances in temporal and spatiotemporal databases, user demands are becoming more complex. As a result, simple queries are replaced by complex multitemporal query scenarios. In this paper we propose a novel image-based approach to temporally group together multidimensional geospatial queries. Correlation signatures act as a powerful raster mapping that visualizes multi-dimensional similarity of multiple queries and expresses it in a temporally referenced manner. Convolution of our raster representation with discrete weight masks can express arbitrary temporal preference (e.g. relative, cyclic queries). Furthermore, our multi-query grouping in the temporal domain can also support temporal relations between queries (e.g. alternative scenarios, AND/OR operators). By transforming the problem in the image domain, the expressiveness of our method allows an intuitive visual interaction to assist non-expert database users.

1. INTRODUCTION

Advances in airborne and earth-based sensors have resulted in the availability of substantial volumes of spatiotemporal datasets to support remote sensing and Geographic Information Systems (GIS) applications. The necessity to organize and retrieve this information in a temporally ordered fashion is well understood. In this paper we introduce a novel approach to group multiple complex queries based on arbitrary temporal rules and preferences. We do so by transforming the problem from the temporal domain to the image one and by applying well-established image processing techniques.

In order to support information dissemination, GIS datasets are typically indexed according to their metadata information. For example, an image may be indexed according to its common metadata parameters (e.g. location, scale, resolution, time, sensor). The result is a multidimensional indexing space, corresponding to the metadata parameters. For an overview of high-dimensional similarity search methods the reader is referred to [1] and for recent advances to [2]. In this paper we assume that a degree of similarity is established by searching in every dimension except the temporal one.

Our approach incorporates mapping techniques of database content. Mapping can be done on the schema of the database [3]. Alternatively mapping is used to aggregate dimensions and visualize them for further processing (e.g. a cube-based representation [4]). In the GIS field a working digital library with collections of geographically referenced materials and services for accessing such collections is presented in [5]. Additionally in [6] exploration of spatiotemporal data is performed with 3D visualizations using SELES as a simulation tool. Our work is similar to the last three approaches in the sense that we are also visualizing the content of a GIS database to support data mining. Our work differs from existing solutions as our mapping is based on a query-database correlation and it is performed exclusively in the temporal domain. In addition we introduce the use of robust image processing techniques to support database management.

Relevant work can also be found in time series analysis and specifically in temporal pattern matching. This type of matching is typically performed using graph-based approaches. In our case the underlying concept is to extract temporally ordered information based on predefined rules/associations. In this regard works like [7, 8] address problems similar to ours in that they make use of a sliding window to identify patterns in a database. Therefore, they are solving the inverse problem that we are facing, trying to extract patterns, instead of projecting given temporal behavior. Multiple granularities are only supported in [8], but neither of the approaches incorporates temporal preference within each event. Temporal grouping is also addressed in the context of aggregating similar information in the temporal domain [9]. In our research we make use of temporal aggregation methods to segment the database time line into query-specific temporal intervals, but grouping appears in the query level using time as a constraint.

By transforming the problem in the image domain, the functionality and expressiveness of our method allows:

- Insertion of arbitrary temporal preference (e.g. relative, cyclic) in the query process.
- Support of temporal relations (e.g. alternative scenarios, AND/OR operators).
- An intuitive user interaction for non-expert database users. Visualizing database content to the user allows a semi-automated query grouping. Temporal areas of interest are manually defined and a significant computational gain is achieved.
- Application of image enhancement methods such as threshold and histogram manipulation to improve temporal grouping. Creating image pyramids with different resolutions and then grouping them together accordingly can result in a multi-resolution approach.

The rest of the paper is organized as follows. Section two introduces the correlation signatures as a temporal raster mapping of the query set on the database. The next section shows a convolution-type approach using correlation signatures. Section 4 provides some examples expressing the functionality of our environment. Finally section 5 presents conclusions and benefits as a result of this work.

2. QUERY-DATABASE MAPPING SIGNATURES

Our goal is to return the highest temporally correlated datasets to a multi-query scenario, for example the best combination of aerial photograph and satellite imagery to the query "*satellite imagery with snow coverage in 1999 and aerial photograph from 1999 until 2000 preferably January 2000".* In order to do so we create a 2D mapping representation (i.e. image) for query grouping. We begin with a simple case of a single query and extend it to multiple ones.

2.1 Single source mapping

In order to extract a dataset requested by a query we perform a correlation analysis in the temporal domain. A degree of correlation (DoC) is computed showing how related each dataset is to each query in every dimension except time. For example if the request is for a satellite image covering area A, with resolution B, and C band channels, (ideal) parameters A, B and C would be compared with parameters of existing imagery and DoC would express their similarity, usually through a percentage.

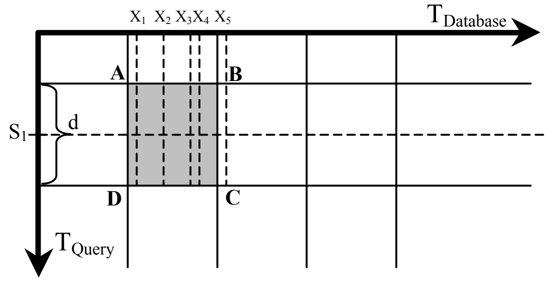

Fig. 1. Query Mapping on Database Timeline

Our analysis starts directly after that. We create a 2D mapping of the query on the database in the temporal domain. This way we create a content mapping space of the degree of correlation registered in time (fig. 1). In the horizontal axis we project the temporal footprints of the database sources and on the vertical the temporal footprints of the query source(s) that were requested. An important note is that the two temporal axes have the same scale. The DB axis starts with the oldest available information increasing to the right and the Query axis begins with the oldest requested query source going down to the most recent one. So if we have a query for source S_1 that lasts time d (*"a satellite image from 01/01/1999 to 12/31/1999"*) and X_s as DB information sources we would have the 2D representation of figure 1. It is important to clarify the role of distance d. It shows that ANY information within that interval would be acceptable. So if we would like an exact answer to the question: "satellite image from 01/01/1999 to 12/12/1999" the result would be given by putting A in (01/01/1999,0) coordinates and assigning d to be one year long. Within the shaded square (pixel) we would have all candidate responses. For each pair (S_1, X_k) the DoC is computed. The maximum DoC will provide the winning pair of (S_1, X_w) and a pixel value that corresponds to the degree of correlation.

2.2 Stacking multiple queries

In order to expand the applicability of the above approach, we combine the results in a multi-source query. Each query source is treated independently and the overall results are combined in one raster representation creating *Correlation Signatures*.

We make use of the above single source example and we create similar correlation profiles over the DB temporal axis. For each query a corresponding row of pixels is created. For optimization and visualization purposes a stacking and shifting process takes place. We allow each query (row) to shift on the X axis such that all corresponding queries are temporally referenced to the database axis. By arranging the queries in a temporal order the relative temporal sequence is preserved. Then we stack all rows together, eliminating gaps. This resolves the temporal correlation in the query axis but these relations are already captured by row shifting.

Stacking facilitates two purposes. By eliminating redundant information (unused temporal cells), we speed up the process. Most importantly we allow the system to support temporally overlapping queries. In figure 2 a 2D representation of our approach is shown. The gray values reflect a normalized representation of the degree of correlation (black =100% correlation, white = 0%). The final pixel size is a combination of the temporal extent of each query in conjunction with the complexity of the temporal preference and the requested accuracy.

3. TEMPORAL QUERY GROUPING USING CORRELATION SIGNATURES

In this section we describe how correlation signatures can support multiple query grouping. The underlying idea is to convolve the once-created correlation signature with different image masks expressing user needs. Each mask row corresponds to a discrete weighting function on a specific query from the query set. The 2D masks are then created as a collection of temporally referenced 1D masks. They are applied on the image through a nearest neighbor operation. The convolution direction is along the X axis, examining a selected portion or the complete temporal footprint of the database.

If we define I(x,y) as the DoC results of the signature (i.e. gray values), M(x,y) as the mask weight expressing temporal preference, W(i) as the weight of each query within the scenario, ($\otimes$) as the special convolution function, and n as the number of queries composing the query scenario, the temporal grouping result on time t is expressed as:

$$Correlation(t) = W * \left| I(x,y) \otimes M(x,y) \right| \tag{1}$$

$$Correlation(t) = \sum_{i=1}^{n} W(i) * \max \left| I(x,y) \otimes M(x,y) \right| \tag{2}$$

There is a difference between the convolution performed above and the traditional one used in image processing. We convolve each image row with the corresponding mask row but we return the maximum convolution value from each row, instead of a summation of all the colvolved elements. In doing so we allow only one winner for each query.

The major applicability of the masks is due to their natural ability to express temporal preference in our query process. Masks form a discrete representation of arbitrary continuous functions. Based on the underlying function and chosen resolution complex queries can be addressed in an efficient way. Temporal preference can be applied to a single query, multiple sets of queries, and/or provide alternative scenarios depending on the choice of mask's setup.

4. FUNCTIONALITY EXAMPLES

Here we introduce examples showing the applicability and the expressiveness of our method.

· **Absolute time queries**

As we mentioned our temporal masks have the ability to express temporal preference. This is accomplished through a discrete representation of arbitrary continuous functions. Within our environment the user can create/insert functions and customize the resolution step. Temporal masks can be static or adaptive to the signature content. In their simplest form they are one-dimensional. For example the user might request "satellite image from 01/01/1999 to 12/31/1999, preferably in early June". This type of query is not a binary one since it has an internal gradual temporal preference. Temporal preference can be expressed using a discrete gaussian distribution, inserted in the mask weights. Other examples that can be naturally expressed through our method are cyclic and/or step queries.

We expand the application of temporal preference from isolated queries to a set of them. In this case we create 2D masks, where each mask row is colvolved with the corresponding signature row. By inserting 2D masks we allow the user to temporally group queries together. For example a multi-query scenario might be formulated as *"return a DEM in 2000 (1999 is acceptable too), a satellite imagery with snow coverage from 1999 until 2000, and an aerial photograph from 1999 until 2000 preferably January 2000".* Such a scenario would result in the convolution mask of fig. 3. The corresponding correlation signature would be created and the discrete version of the mask would be overlaid (fig. 2).

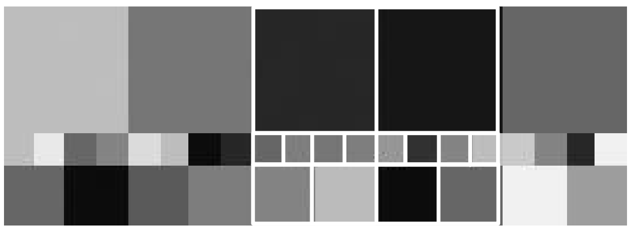

Fig. 2. Convolution mask applied on Correlation Signature

· **Relative time queries**

An important characteristic of our approach is the support for temporal grouping based on relative times. If a temporal distance is established between queries then the convolution mask can be created and overlaid. This time since there is no fixed timestamp the algorithm can scan parts or the whole database timeline for successful results. The scanning process is done simply by sliding the mask along the X axis and evaluating the results.

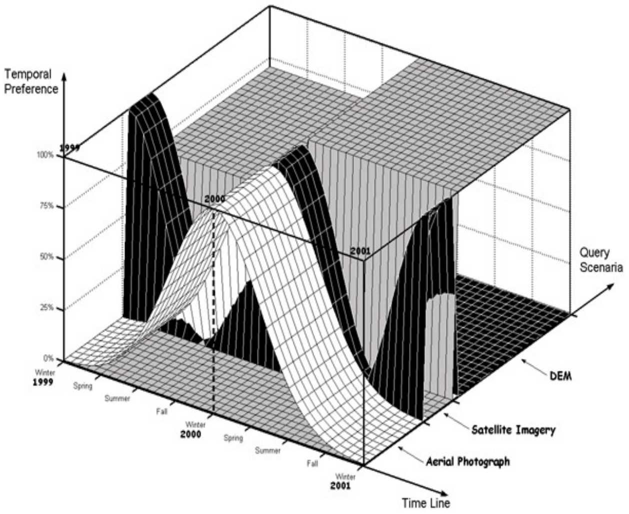

Fig. 3. Multi-temporal continuous Convolution Mask

· **Combining multiple scenarios**

In order to accommodate multiple scenarios in our solution we introduce a three-dimensional version of the masks. The third dimension shows alternative masks (scenarios) that might satisfy user demands. In other words this type of mask is composed of multiple layers of 1D or 2D (sub)masks. Alternative scenarios can be weighted (i.e each mask layer has a weight) and OR/AND operators are used to group them together. In more advanced operations mask choice can be dynamically adjusted based on the signature values. In this last category we allow the system to make adjustments during the signature scanning process (dynamic convolution). A rule-based approach can be incorporated to relate masks together. By using this adaptive method we scan our signature only once, thus reducing retrieval times.

For example a complex temporal query might be: Search for an aerial photograph taken in time t *and* a satellite image at t+2. Depending on the results obtained from the first layer of the mask the algorithm adapts dynamically to the following two choices. IF overall DoC>80% AND satellite image correlation>70% (meaning a good dataset), THEN search for vector data at t+4 (for further processing). IF the overall correlation is high, but the satellite image is not that good, THEN extract a DEM from time t-2 until t+2, preferably in time t (so the aerial photograph can be draped on top). Each choice is represented by an additional layer in the mask.

5. CONCLUSIONS

Temporal data mining is attracting primary attention as the temporal information availability and nature of datasets are increasing. In this paper we introduce correlation signatures as a powerful raster mapping that visualizes multi-dimensional similarity of multiple queries and expresses it in a temporally referenced manner. By convolving the above raster representation with discrete weight masks we can introduce arbitrary temporal preference (e.g. relative, cyclic) in our queries. Convolution masks function as statistical maximum operators and as group results. They can also express temporal relations (e.g. alternative scenarios, AND/OR operators) between queries allowing multi-query grouping in the temporal domain.

This is the rationale for our adoption of an image-processing approach for what is in essence a traditional database problem. This allows us to manipulate raster techniques such as convolution for information grouping and retrieval. The raster representation is used for semi-automated grouping as well as for database content visualization. Other techniques like image enhancement and image pyramids facilitate retrieval efficiency.

6. ACKNOWLEDGMENTS

This work is supported by the National Science Foundation through grants DG-9983445, EPS-9983432, and ITR-0121269, and the Advanced Research and Development Activity (ARDA) through a subaward by BAE Systems.

7. REFERENCES

[1] R. Weber, H. Schek, and S. Blott. A quantitative analysis and performance study for Similarity Search Methods in High Dimensional Spaces, In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB)*, pp. 194-205,1998.

[2] F. Korn, B.U. Pagel, C. Faloutsos. On the Dimensionality Curse and the Self-Similarity Blessing, *IEEE Trans. Knowledge and Data Engineering,* 13(1), pp. 96-111, 2001.

[3] L. Yan, R. J. Miller, L. M. Haas and R. Fagin. Data-Driven Understanding and Refinement of Schema Mappings, Proc. of ACM SIGMOD, pp. 485-496, 2001.

[4] 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, pp. 29-54, 1997.

[5] The Alexandria Digital Library Project, http://www.alexandria.ucsb.edu/, Current as of April 7, 2003.

[6] S. T. Carpendale, D. J. Cowperthwaite, M. Tigges, A. Fall and F. D. Fracchia. The Tardis: A Visual Exploration Environment for Landscape Dynamics, *SPIE Conference on Visual Data Exploration and Analysis VI*, 1999.

[7] H. Mannila, H. Toivonen, and A. I.Verkamo. Discovering frequent episodes in sequences, In *Proc. Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95),* pp. 210-215, 1995.

[8] C. Bettini, X. S. Wang, S. Jajodia, and J. Lin. Discovering Frequent Event Patterns with Multiple Granularities in Time Sequences, *IEEE Transactions on Knowledge and Data Engineering,* 10(2), pp. 222-237, 1998.

[9] D. Zhang, A. Markowetz, V. Tsotras, D. Gunopulos, and B. Seeger. Efficient Computation of Temporal Aggregates with Range Predicates, *Proc. of PODS*, 2001.