Spatial Cone Tree: An Index Structure for Correlation-based Similarity Queries on Spatial Time Series Data

Pusheng Zhang, Shashi Shekhar, Vipin Kumar
Computer Science & Engineering Department, University of Minnesota
Emails: [pusheng,shekhar,kumar]@cs.umn.edu

Yan Huang
Department of Computer Science
University of North Texas
Email: huangyan@cs.unt.edu

October 1, 2003

1 Introduction

A spatial time series dataset [15] is a collection of time series [3], each referencing a location in a common spatial framework [14]. Finding highly correlated time series from spatial time series datasets collected by satellites, sensor nets, retailers, mobile device servers, and medical instruments on a daily basis is important for many application domains such as epidemiology, ecology, climatology, and census statistics. For example, such queries were used to identify the land locations where the climate was often affected by El Nino [13]. However, correlation queries are computationally expensive because large spatio-temporal frameworks contain many locations and time points. The design of efficient access methods to facilitate correlation-based query processing[1, 7] on spatial time series data, the focus of this work, is crucial to organizations which make decisions based on large spatio-temporal datasets.

The problem of designing an efficient indexing method for spatial time series data can be defined as follows. Given a set of operations, e.g., finding most correlated time series with a query time series, finding all time series with a correlation above a given threshold for a query time series, insert, delete, and bulk load. Our goal is to find an access method, which provides efficient support for the frequent operations in spatial time series datasets.

Previous work [1, 5, 7] on indexing time series data has focused on dimensionality reduction followed by the use of low dimensional indexing [8, 10, 11] in the transformed space. Unfortunately, the efficiencies of these approaches deteriorates substantially when a small dimensions of subspace cannot represent enough information in the time series data. Many spatial time series datasets fall in this category. For example, finding anomalies is more desirable than finding well-known seasonality in the knowledge discovery process of spatial time series datasets. Therefore, data used in anomaly detection is usually data whose seasonality has been removed. After transformations are applied on deseasonalized data, the power spectrum spreads all over most dimensions. Furthermore, in most spatial time series datasets, the number of spatial locations is much greater than the length of time series. This makes it possible to improve the performance of query processing of spatial time series data by exploiting spatial proximity in the design of access methods.

In this paper, we develop the spatial cone tree, an index structure for spatial time series data. The spatial cone tree groups similar time series together based on spatial proximity. Correlation-based similarity queries are facilitated using spatial cone trees. Our approach is orthogonal to dimensionality reduction solutions. The spatial cone tree structure preserves full length of time series, and therefore it is insensitive to the distribution of the power spectrum after data transformations. Algebraic analyses using cost models and experimental evaluations are carried out to show that the proposed access method saves a large portion of computational cost, ranging from 40% to 98%.

2 Proposed Access Method

2.1 Spatial Cone Tree Structure

A normalized time series with m time points is located on the surface of an m-dimensional unit sphere [15]. The correlation of two time series is directly related to the angle between the two normalized time series vectors in the multi-dimensional unit sphere. A cone [15] is a set of normalized time series in a multi-dimensional unit sphere.

A *spatial cone tree* is a search tree in which each node represents a cone. Each node includes multiple normalized time series in a multi-dimensional unit sphere. The root has at least two children unless it is a leaf. A leaf node contains an array of leaf entries. A leaf entry consists of a normalized time series. A non-leaf node contains an array of node entries. Every cone node x has the following common fields:

- ◇ axis(x), the mean of all normalized time series in cone x
- ◇ span(x), the maximal angle between any normalized time series vector in node x and axis(x) vector

Let M denote the maximum number of entries that fit in one node and max_span denote the maximum span threshold. In a cone node, the number of normalized time series and the span should be no more than M and max_span respectively.

2.2 Spatial Cone Tree Construction

Both space-partitioning methods [11] and data-partitioning methods [8, 2] can be applied to the spatial cone tree construction. For simplicity, a top-down quad-tree-like [11] spatial cone tree construction method is used.

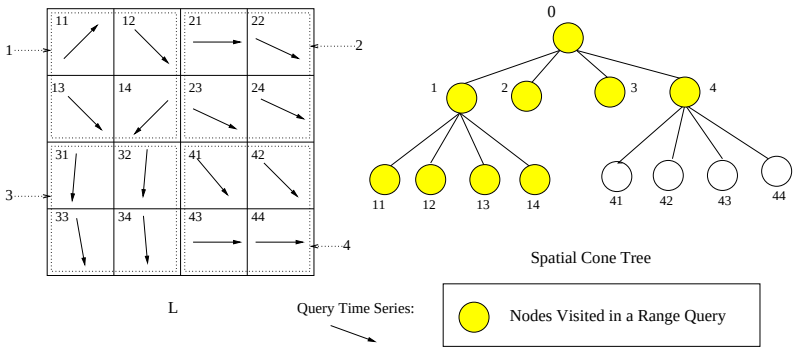

Figure 1: Spatial Cone Tree Construction

We begin with the whole space as the root of the cone tree. When the cone span exceeds max_span or the number of time series exceeds M, a cone is split into four sub-cones based on

spatial proximity. The time series are re-distributed into the sub-cones, and the axis and span are calculated for each sub-cone. Each sub-cone is checked and split recursively until its cone span is no more than max_span and its number of time series is no more than M. Figure 1 illustrates the spatial cone tree construction for a small spatial time series dataset. The spatial framework consists of 16 locations, and each location contains a time series of length 2. Each arrow in a location represents the normalized time series vector. The whole space L was divided into four disjoint quadrants 1, 2, 3, and 4, and each quadrant corresponds to a cone node in the spatial cone tree. $M = 4$ and $max\_span = 10°$. Cones 1 and 4 are further split because their spans exceed max_span.

The spatial cone tree structure allows cone overlapping in multi-dimensional unit spheres. Thus it cannot guarantee that only one search path is required for an exact match query. However, the overlapping-cones technique does not hurt the performance of correlation-based similarity query processing. We will show the proposed access method saves a large portion of computational cost in Section 3.

2.3 Operations

The spatial cone tree can efficiently support a set of operations including finding most correlated time series with a query time series(point query), finding all time series with a correlation above a given threshold for a query time series(range query), finding all pairs of time series with a correlation above a given threshold between two spatial cone trees(join), insert, delete, and bulk load. Due to page limits, we only discuss the operation of a range query, finding all time series with a correlation over a given threshold for a query time series.

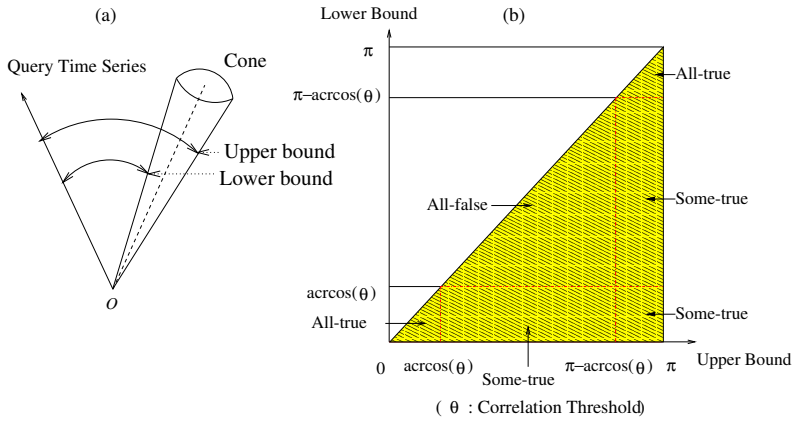

Figure 2: (a)Upper Bound and Lower Bound (b)Filtering Lemmas

The key idea of range query processing in our approach is processing the query in a filter-and-refine style on the cone level instead of on the time series level. The *filtering* step traverses the spatial cone tree, applying filtering lemmas on the cones. The filtering lemmas are developed to eliminate cones with all times series satisfying/dissatisfying the correlation threshold. As shown in Figure 2, the possible relationships between a cone and the query time series consist of all-true, all-false, or some-true. All-true means that all times series with a correlation over the correlation threshold; all-false means all time series with a correlation less than the correlation threshold; some-true means only part of time series with a correlation over the correlation threshold. Therefore, the cones satisfying all-true or all-false are filtered out. The cones satisfying some-true are traversed recursively until all-true or all-false is satisfied or a leaf cone is reached. The *refinement* step

manually checks the some-true leaf cones.

For example, a range query is carried out on the spatial cone tree shown in Figure 1. The search begins with the root of the spatial cone tree. The root cone is a some-true cone, and therefore its children are traversed. Cones 2 and 4 are all-true cones, and cone 3 is a all-false cone. Thus only cone 1 is traversed further. All time series in cones 2 and 4 and time series 12 and 13 are identified to be highly correlated with the query time series.

The traversal strategies in the filtering step can use any traversal strategy [4, 9] used in tree-based spatial index structures [10, 11, 12]. The processing algorithms are proved complete and correct, i.e., there are no false dismissals or false admissions. Formal proofs will be available in the full paper.

3 Evaluations

We provide algebraic cost models for processing the operations on spatial cone trees. These operations are efficiently supported. The proposed range query processing algorithm outperforms sequential scan, and the proposed join query processing algorithm outperforms the nested-loop strategy.

We evaluate the performance of the proposed query processing algorithms using a dataset from NASA Earth science data. Correlation-based range queries and join queries were carried out on Sea Surface Temperature (SST) data in the eastern tropical region of the Pacific Ocean and on Net Primary Production (NPP) data in the United States. The SST data contain 11556 ocean cells of the Pacific Ocean and the NPP data contain 2901 land cells of the United States. The records of SST and NPP were monthly data from 1982 to 1993. The preliminary experimental results show that the proposed query processing algorithms often save a large fraction of computational cost, e.g., saving 97.6% of the computational cost for a join query using minimal correlation threshold 0.9 between the SST and NPP data.

4 Conclusion

In this paper, we developed the spatial cone tree, an index structure for correlation-based similarity queries. Correlation-based query processing was efficiently facilitated using the spatial cone tree. Analytical and experimental evaluations were carried out to show the efficiency of proposed query processing algorithms. In future work, we would like to investigate the generalization of spatial cone trees to non-spatial index structures using spherical k-means [6] to construct cone trees.

Acknowledgments

This work was partially supported by NASA grant No. NCC 2 1231 and by Army High Performance Computing Research Center contract number DAAD19-01-2-0014. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by the AHPCRC and the Minnesota Supercomputing Institute.

We are particularly grateful to Prof. Kyu-Young Whang for his helpful comments and valuable discussions. We would also like to express our thanks to Kim Koffolt for improving the readability of this paper.

References

[1] R. Agrawal, C. Faloutsos, and A. Swami. Efficient Similarity Search In Sequence Databases. In *Proc. of the 4th Int'l Conference of Foundations of Data Organization and Algorithms*, 1993.

[2] S. Berchtold, D. Keim, and H. Kriegel. The X-tree: An Index Structure for High-Dimensional Data. In *The Proc. of 22nd VLDB Conference*, 1996.

[3] G. Box, G. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control.* Prentice Hall, 1994.

[4] T. Brinkhoff, H. Kriegel, and B. Seeger. Efficient Processing of Spatial Join Using R-trees. In *ACM SIGMOD*, 1993.

[5] K. Chan and A. W. Fu. Efficient Time Series Matching by Wavelets. In *Proc. of the 15th ICDE*, 1999.

[6] I. Dhillon, J. Fan, and Y. Guan. Efficient Clustering of Very Large Document Collections. In R. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu, editors, *Data Mining for Scientific and Engineering Applications.* Kluwer Academic Publishers, 2001.

[7] Christos Faloutsos. *Searching Multimedia Databases By Content.* Kluwer Academic Publishers, 1996.

[8] A. Guttman. R-Trees: A Dynamic Index Structure For Spatial Searching. In *ACM SIGMOD*, 1984.

[9] Y. Huang, N. Jing, and E. Rundensteiner. Spatial Joins using R-trees: Breadth-First Traversal with Global Optimizations. In *The Proc. of 23nd VLDB Conference*, 1997.

[10] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases: With Application to GIS.* Morgan Kaufmann Publishers, 2001.

[11] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley Publishing Company, Inc., 1990.

[12] S. Shekhar and S. Chawla. *Spatial Databases: A Tour.* Prentice Hall, ISBN:0130174807, 2003.

[13] G. H. Taylor. Impacts of the El Nio/Southern Oscillation on the Pacific Northwest. http://www.ocs.orst.edu/reports/enso_pnw.html.

[14] Michael F. Worboys. *GIS - A Computing Perspective.* Taylor and Francis, 1995.

[15] P. Zhang, Y. Huang, S. Shekhar, and V. Kumar. Correlation Analysis of Spatial Time Series Datasets: A Filter-and-Refine Approach. In *the Proc. of the 7th PAKDD*, 2003.