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

Pusheng Zhang
Shashi Shekhar
Vipin Kumar
Department of Computer Science and Engineering
University of Minnesota

Yan Huang
Department of Computer Science
University of North Texas

Illustrative Application Domain

⋆ Teleconnection: Ocean affects lands

- NASA Funded Project: Discovery of Changes from the Global Carbon Cycle and Climate System Using Data Mining
- E.g.: El Nino
  - Anomalous warming of tropical Eastern Pacific leads to:
    - Heavy Rain in Peru and Drought in Australia

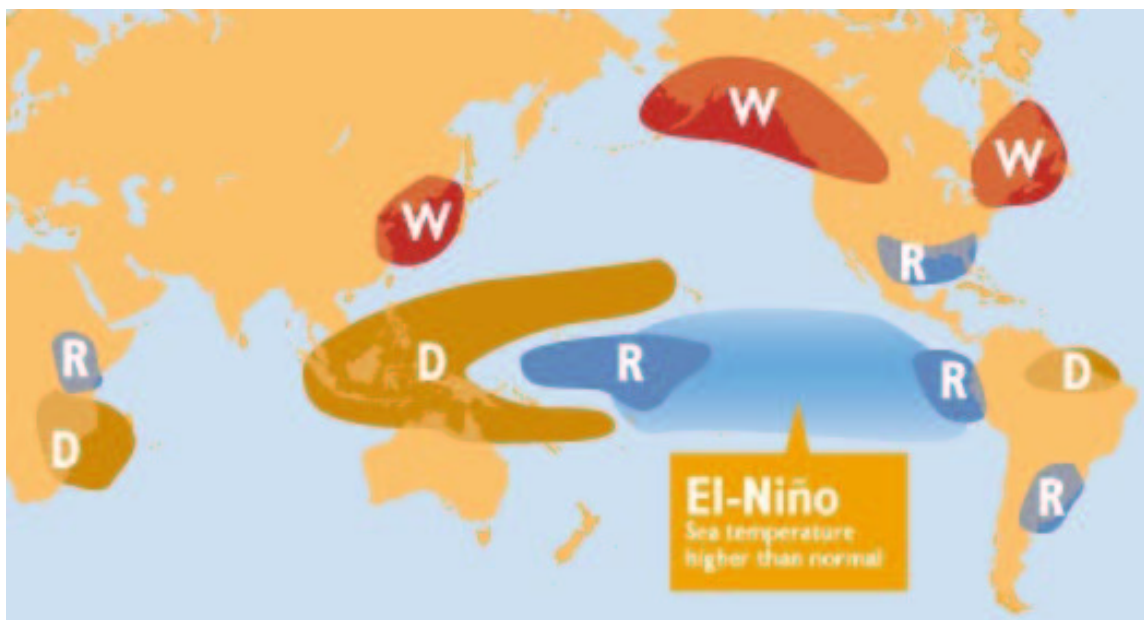


Figure 1: Global Influence of El Nino during the Northern Hemisphere Winter

  - D indicates drought
  - R indicates unusually high rainfall
  - W indicates abnormally warm periods

Spatio-temporal Data Mining

⋆ The process of discovering

- interesting, potentially useful, non-trivial patterns
- from large spatio-temporal datasets

⋆ Application Domains

- E.g.: Climatology, Earth science, and Epidemiology
- Spatial time series data
  - a spatial framework: set of locations
  - k attributes per location, each a time series of length m

⋆ Analysis Questions

- Classification: predict long-term (3-6 months) rainfall
- Clustering: find spatio-temporal homogeneous regions
- Relationship: find locations influenced by El Nino
- Anomaly: find unusual years in ocean temp. data

Relationship Mining in Spatial Time Series Datasets

⋆ Interest Measure of Earth Scientists

- Correlation(time series f, time series g, length m)

$$r = \frac{1}{m-1} \sum_{t=1}^{m} (\frac{(f_t - \overline{f})(g_t - \overline{g})}{\sigma_f \sigma_g}$$

- Correlation Significance Test
  - Fisher's Z test: $Z = \frac{1}{2} \log \frac{1+r}{1-r}$
  - confidence level $\Rightarrow r_{min}$
    - E.g.: confidence level 95% $\Rightarrow r_{min} = 0.46$
  - Student-t test for short time series

Example of Correlation Queries

⋆ Correlation Queries:

- Datasets: $D^1(location, f)$ and $D^2(location, g)$
- Range Query: select $D^1.location$ from D^1
  where corr($D^1.f$, SOI) $\geq \theta$
- Join Query: select $D^1.location, D^2.location$ from D^1, D^2
  where corr($D^1.f$,$D^2.g$)$\geq \theta$

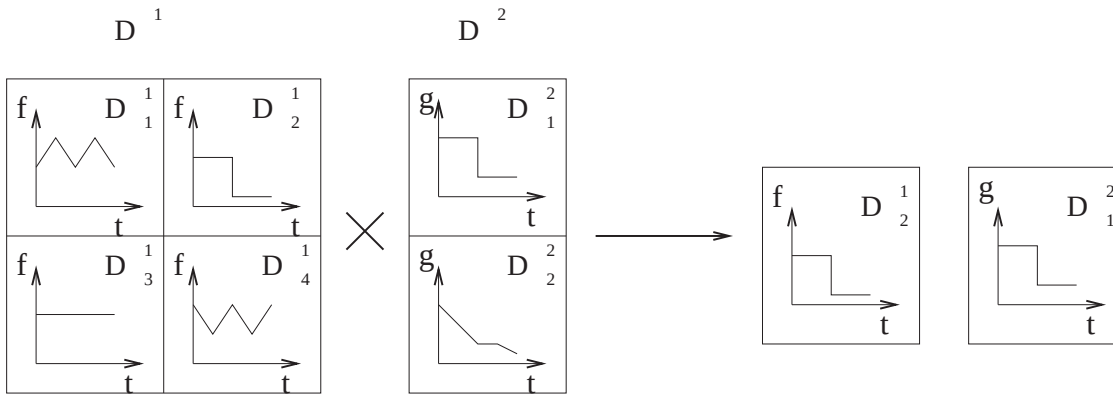


Figure 2: Illustration of Join Query in Spatial Time Series Data

Problem Definition

⋆ Given:

- Spatial time series datasets
- A set of operations
  - Correlation-based Queries:
    - range query, join, and nearest-neighbor query
  - Maintenance Operations: insertion, deletion, bulk-loading

⋆ Find: A disk-based data structure

⋆ Objectives: computational efficiency

⋆ Constrains:

- Correctness and Completeness
  - No false admissions and false drops
- Time shift and smoothing: part of preprocessing
- Spatial autocorrelation in attribute time series
- Size(spatial framework) $\gg$ Size(temporal framework)

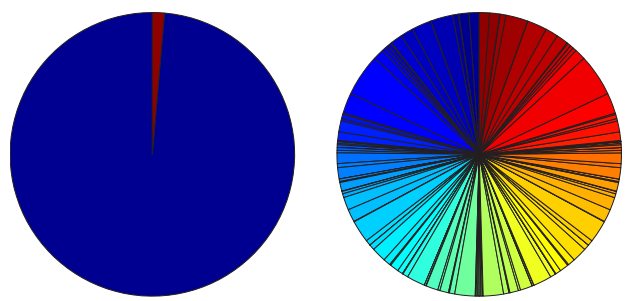
Related Work

⋆ Reduce Time Dimensionality

- Transformations: DFT, DWT
- Low-dim indexing: R-tree, Grid file, Quad-tree
- e.g., F-index[Agrawal et al. 1993], [Chan et al. 1999]

⋆ Limitations

- Effectiveness ↓ for non-skewed power spectrum
  - Non-skewed power spectrum: removing seasonality for time series



(a) (b)

Figure 3: Power Spectrum of DFT for: (a) Raw Time Series (c) Removing Seasonality

- F-index is not efficient for (b)
- Room to improve using spatial properties

Our Contributions

- ⋆ Spatial Cone Tree
  - An auxiliary search structure on high-dim normalized time series
  - spatial autocorrelation facilitates cheap bulk load
  - orthogonal to time series dimension reduction
- ⋆ Speed up correlation-based similarity queries
  - Range Queries
  - Join Queries

Overview

✓ Motivation and Problem Definition

✓ Related Work and Contributions

⇒ Proposed Approach

⋆ Evaluation of Proposed Approach

⋆ Conclusions & Future Work

Overview of Proposed Approach

- ⋆ Normalization of time series [slide 10 – 11]
- ⋆ Spatial Cone Tree Structure[slide 12 – 14]
  - Concept of Cone [slide 12]
  - Spatial Cone Tree [slide 13 – 14]
- ⋆ Operations [slide 15 – 18]
  - Range Query [slide 16 – 17]
  - Bulk Loading [slide 18]

Normalization of Time Series to Unit Sphere

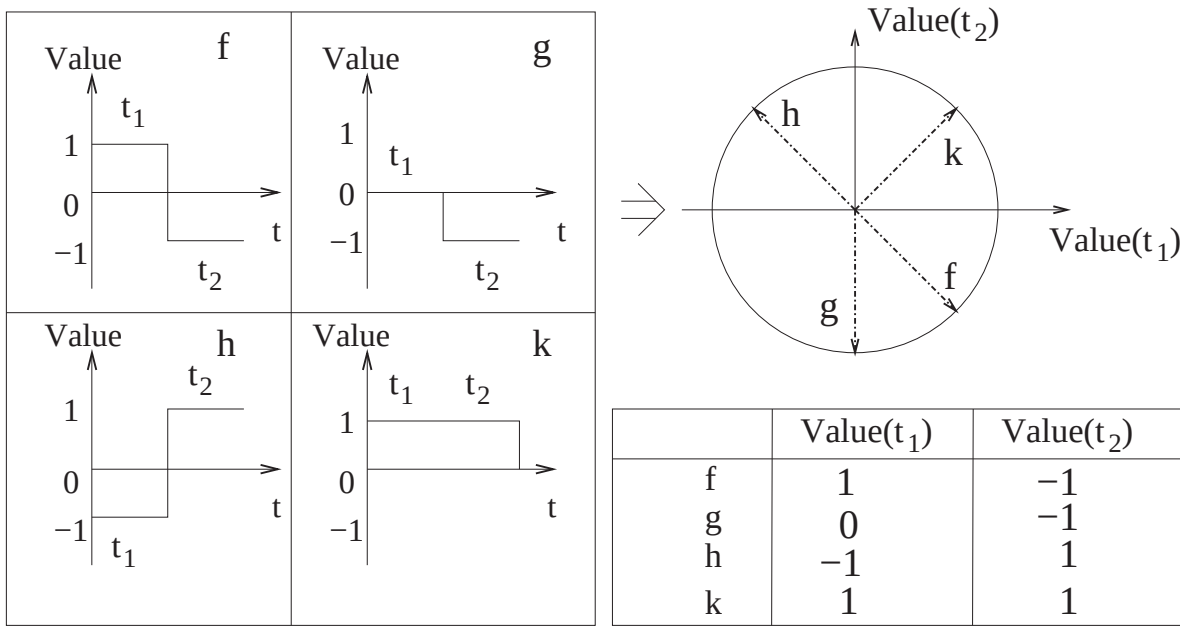
⋆ Time Series of length m , $f = \{f_1, f_2, \ldots, f_m\}$

- Normalized time series: unit vector $\widehat{f}$

$$\widehat{f} = \frac{f - \overline{f}}{\sigma_f}$$

⋆ Fact 1: $\widehat{f}$ is on surface of a m-dimensional unit sphere

⋆ Fact 2: $corr(f, g) = cos(\widehat{f}, \widehat{g}) = \widehat{f} \cdot \widehat{g}$ for f, g



	Value(t_1)	Value(t_2)
f	1	−1
g	0	−1
h	−1	1
k	1	1

Figure 4: Time Series in Unit Sphere

Correlation in Unit Sphere

⋆ Given:

- Time Series f, g and correlation threshold θ

⋆ Fact 3: $corr(f, g) > \theta \Rightarrow arccos(\widehat{f} \cdot \hat{g}) \in (0, arccos(\theta))$

⋆ Fact 4: $corr(f, g) < -\theta \Rightarrow arccos(\widehat{f} \cdot \hat{g}) \in (\pi - arccos(\theta), \pi)$

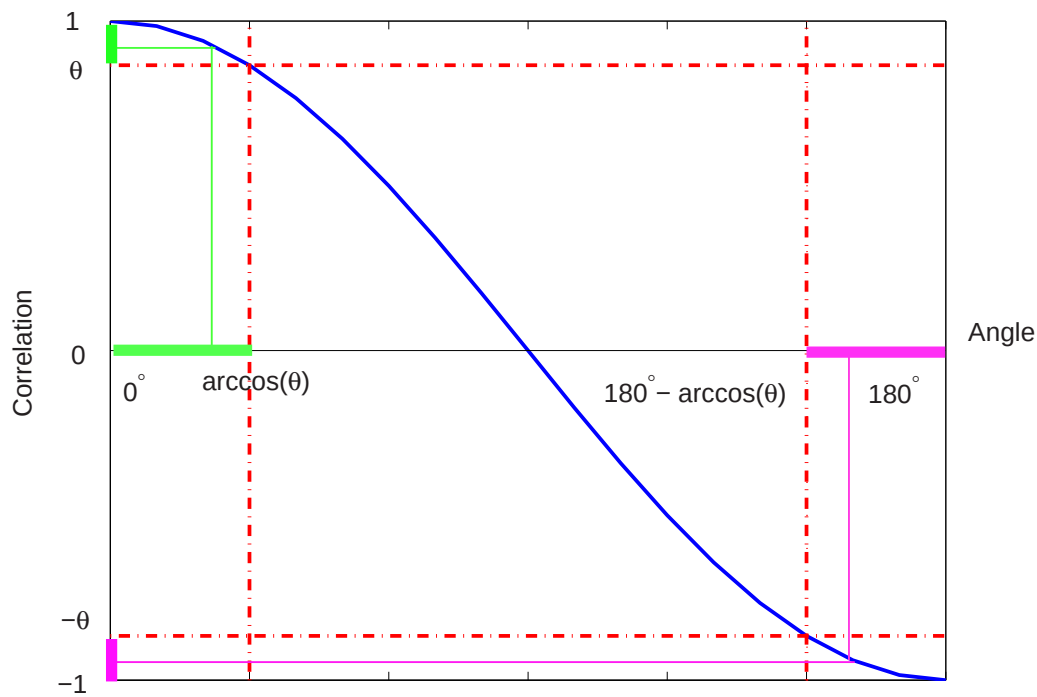


Figure 5: Correlation in Multi-dimensional Unit Sphere

Concepts of Cones

⋆ Cone

- Group of time series unit vector
- Specifications
  - axis unit vector
  - span: largest angle between axis and any unit vector

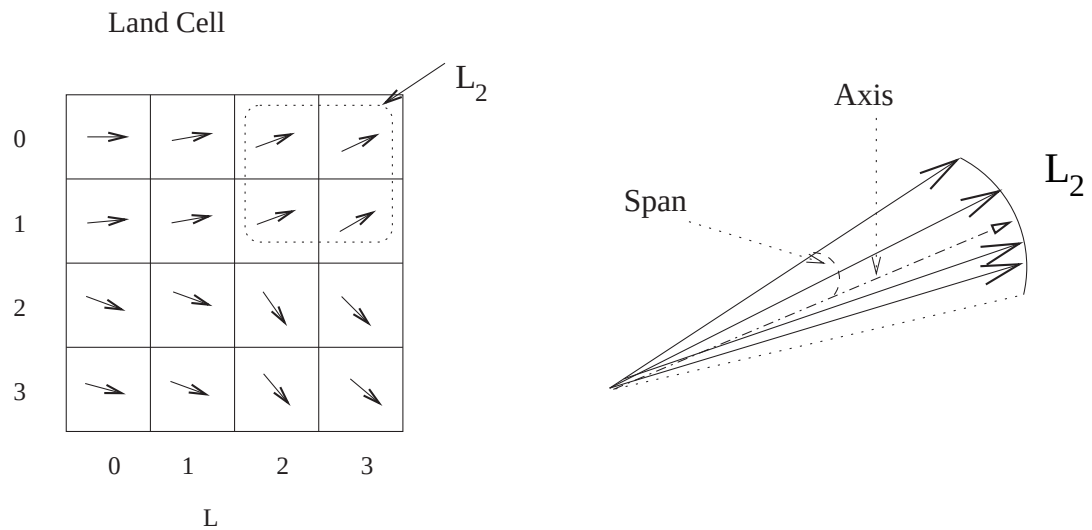


Figure 6: Illustration of Cone

Spatial Cone Tree

⋆ Spatial Cone Tree

- Auxiliary search structure on normalized time series data
- Leaf node: cone and pointer to disk page containing one or more normalized time series
- Internal node: cone, pointer to index page

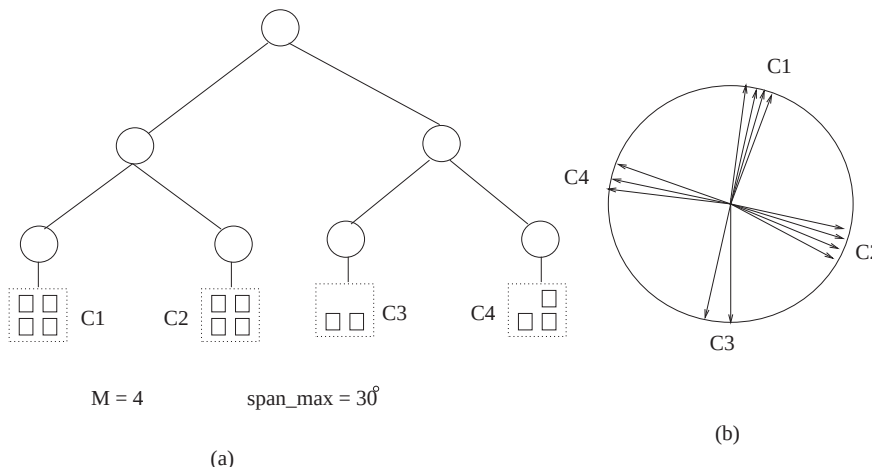


Figure 7: (a)Illustration of a Spatial Cone Tree (b)Normalized Time Series Vectors in a Circle

Design Issues

- ⋆ Blocking Factor
  - the number of index records per disk page
  - Depend on length of time series(cone axis)
    - index record includes cone span, cone axis, and pointer
  - Challenges on long time series
    - Dimensionality reduction: reduce length of time series
    - Divide long time series into fixed-length smaller chunks
- ⋆ Balancing Issue
  - Balanced tree: all leaf nodes are on the same level
  - Balancing is desirable
  - Overheads of keeping balancing are extensive
  - Begin with unbalanced tree structure

Operations on Spatial Cone Trees

- ⋆ Query Operations:
  - Point Query: find exact time series
  - Range Query:
    - find highly correlated time series with query time series
  - Nearest-neighbor query:
    - find closest time series with query time series
- ⋆ Maintenance Operations:
  - Insertion
  - Deletion
  - Bulk-loading

Range Query Processing

⋆ Filter-and-Refine

- Filtering lemmas: All-True and All-False lemmas
- filter(cone1, cone2) ∈ [all-true, all-false, some-true]
- if some-true then refinement: check the candidates

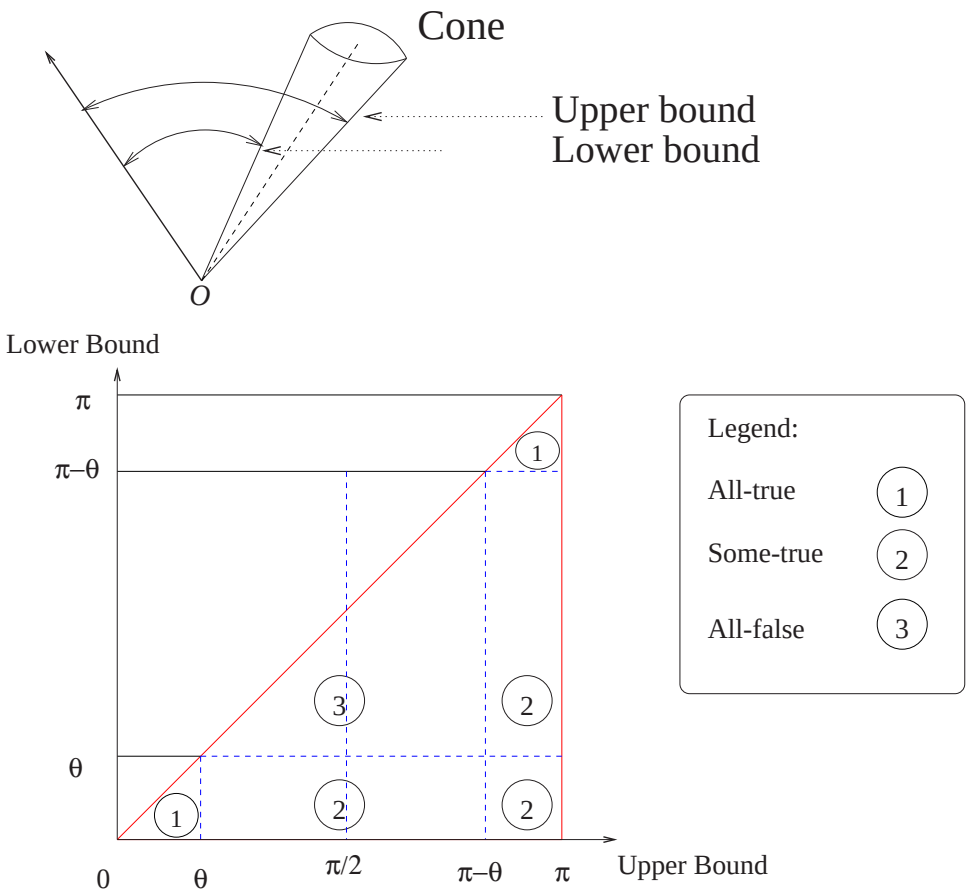


Figure 8: Filtering Lemmas

An Example of Range Query Processing

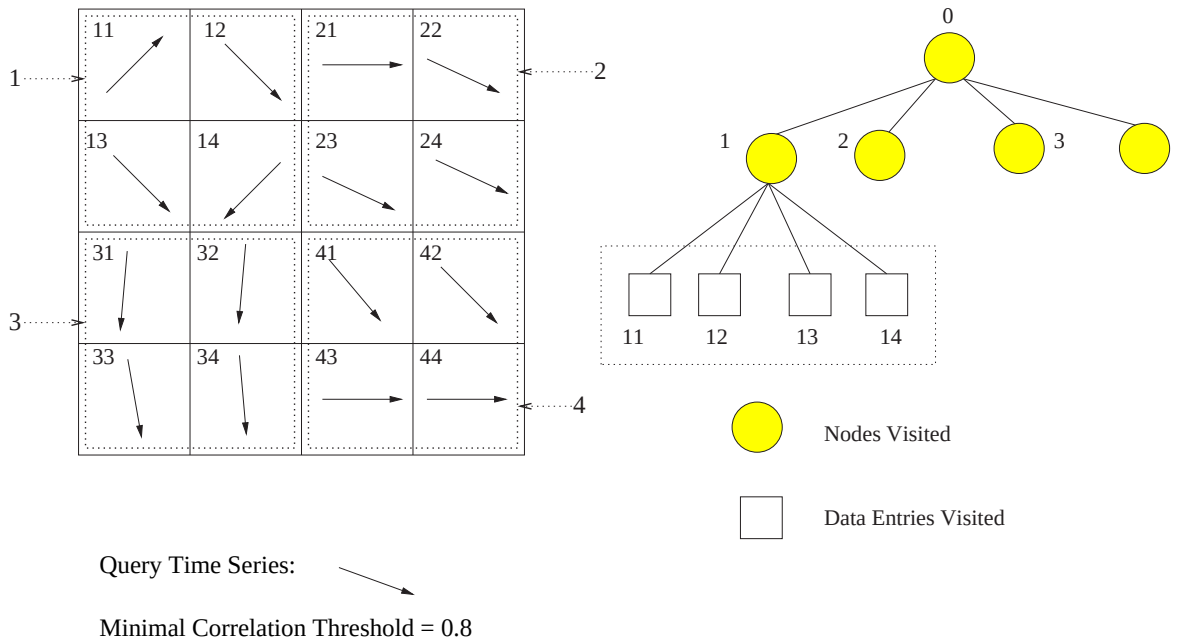


Figure 9: Spatial Cone Tree

Nodes visited	Filtering	Refinement
1	Some-True	4
2	All-True	0
3	All-False	0
4	All-True	0

Table 1: Part of Range Query Processing in Example Data

Illustration of Bulk-Loading

⋆ Rationale and Strategy

- Spatial autocorrelation
  - Nearby objects are related
- Space-partition based bulk loading: e.g., quad-tree like

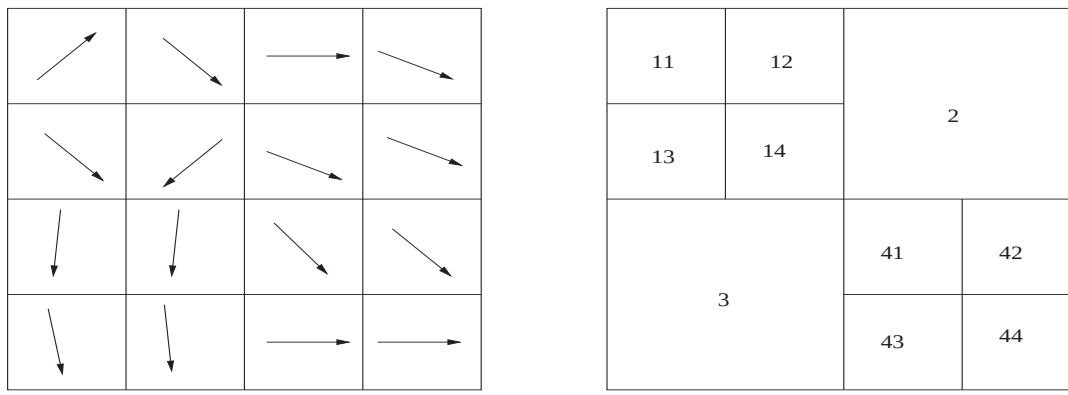


Figure 10: A Quad-tree like Bulk-loading

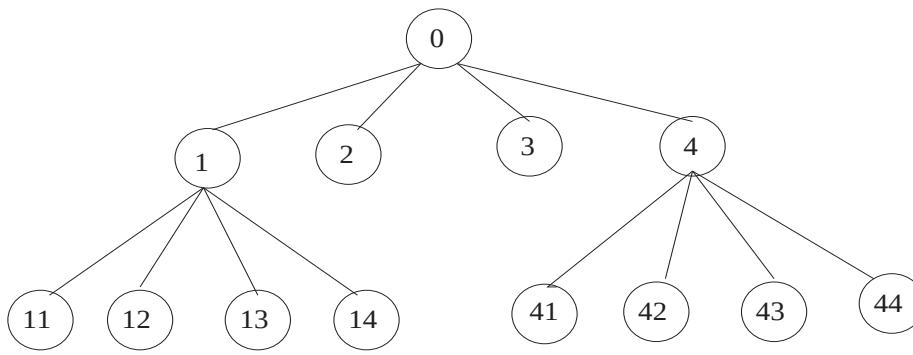


Figure 11: Spatial Cone Tree

Overview

✓ Motivation and Problem Definition

✓ Related Work and Contributions

✓ Proposed Approach

⇒ Evaluation of Proposed Approach

⋆ Conclusions & Future Work

Overview of Evaluations

- ⋆ Analytical evaluation with cost models
  - Range queries
    - Depth-First traversal
  - Join queries
    - Nested loops
    - Depth-first traversal is used in inner loop
    - Leaf scanning is used in outer loop
- ⋆ Experimental evaluation [slide 21 –23]
  - Data: real Earth science data [slide 21]
  - Performance evaluation on range queries and join queries [slide 22 – 23]

Experimental Evaluation

⋆ Workload

- S: Monthly Sea Surface Temperature of Pacific
- N: Monthly Net Primary Production of USA
- Temporal Span: 1982-1993 (12 * 12 = 144)
- Spatial Resolution: $0.5° \times 0.5°$, #S: 11556, #N: 2901

S: SST of Pacific

Longitude	Latitude	SST (82-93)
120.5W	5.0N	2560,2567, ..., 2787
121.0W	5.0N	2567,2456, ..., 2789
...		
179.5W	5.0S	2034,2175, ..., 2445

N: NPP of USA

Longitude	Latitude	NPP (82-93)
97.0W	33.5N	4.56,5.67, ..., 6.90
97.0W	34.0N	4.34,6.29, ..., 7.56
...		
97.0W	38.0N	2.34,3.23, ..., 4.34

Table 2: Tables Schema and Sample Contents of Table S and Table N

Results of Range Query Processing

⋆ Variable Parameters

- Minimal correlation threshold θ: 0.3-0.9

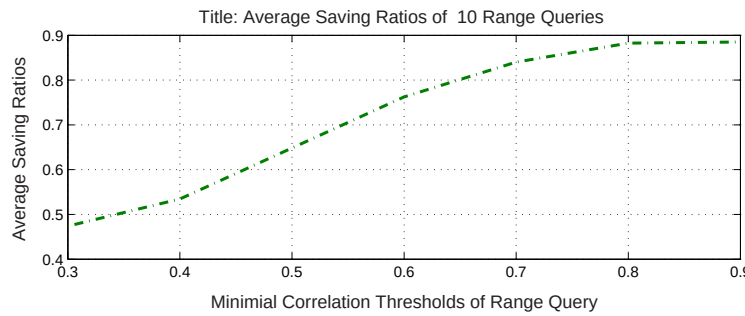


Figure 12: Average Saving Ratios of 10 Queries

⋆ Computational savings range 45% – 89%

⋆ Increase $\theta \Rightarrow$ savings $\uparrow$

Results of Join: Effect of Minimal Correlation Threshold

⋆ Variable Parameters

- Minimal correlation thresholdθ: 0.3, 0.5, 0.7, 0.9

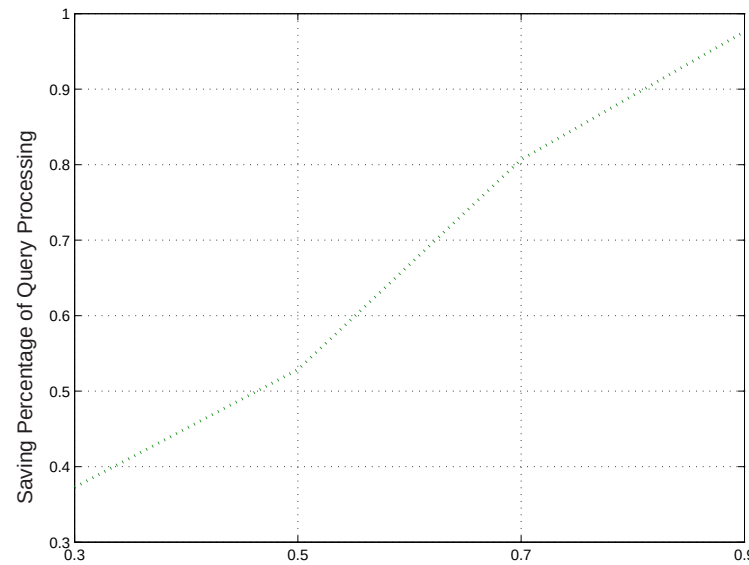


Figure 13: Savings of Join Processing

⋆ Computational savings range 37% – 98 %

⋆ Increase $\theta \Rightarrow$ savings $\uparrow$

Conclusions

- ⋆ Spatial time series data and correlation-based queries are abundant in many applications
  - e.g., NASA Earth science data, epidemiology, climatology
- ⋆ Spatial Cone Tree
  - Auxiliary search data structure on high-dim normalized time series data
  - facilitate correlation-based queries
  - spatial autocorrelation facilitated cheap bulk load
  - orthogonal to time series dimension reduction

Future Work

- ⋆ Generalize cone tree to generic time series data
- ⋆ Compare proposed approach with temporal dimension reduction approaches
  - E.g.: our approach vs. F-index
- ⋆ Propose and develop query languages to facilitate queries on spatial time series

Thank You!

More details are available online at:
http://www.cs.umn.edu/research/shashi-group
Email: pusheng@cs.umn.edu

Filtering Lemmas (Backup)

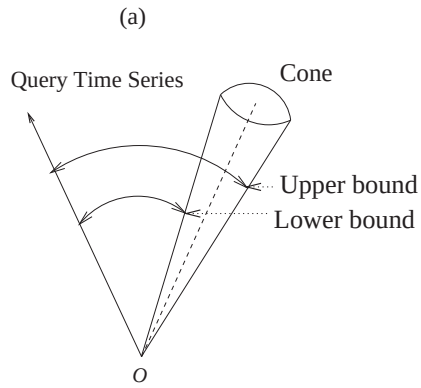


Figure 14: Upper Bound and Lower Bound

⋆ Angles between query time series and cone

- Upper Bound Angle: γ_{max}
- Lower Bound Angle: γ_{min}

1. **If** $\gamma_{max} \in (0, arccos(\theta))$**, then** $\angle(\overrightarrow{T_q}, \overrightarrow{T}) \in (0, arccos(\theta))$**;**
2. **If** $\gamma_{min} \in (180° - arccos(\theta), 180°)$**, then** $\angle(\overrightarrow{T_q}, \overrightarrow{T}) \in (180° - arccos(\theta), 180°)$**;**
3. **If** $\gamma_{min} \in (arccos(\theta), 180°)$ **and** $\gamma_{max} \in (\gamma_{min}, 180° - arccos(\theta))$**, then** $\angle(\overrightarrow{T_q}, \overrightarrow{T}) \in (arccos(\theta), 180° - arccos(\theta))$**.**

⋆ Filtering Lemmas

- All-True Lemma: case 1 or case 2
- All-False Lemma: case 3

Algorithm of Range Query Processing(Backup)

⋆ Range Query Processing

Input: 1) TR: a spatial autocorrelation-based search tree;
2) T_q: the query time series;
3) θ: minimal correlation threshold;

Output: all time series whose correlations with T_q are above θ;

Method:

```
traverse TR; for each cone c on the route do                          (1)
 // any tree traversal algorithm TR
  Filter_Flag = Cone-level_Join(T_q, c, θ);                           (2)
  if (Filter_Flag == ALL_TRUE)                                        (3)
    output all time series in the cone c                              (4)
  else if (Filter_Flag != ALL_FALSE)                                  (5)
    if c is a leaf node                                               (6)
      for all pair T_q and s from c do                                (7)
        High_Corr_Flag = Instance-level_Join(T_q,s, θ);               (8)
        if (High_Corr_Flag) output s;                                 (9)
    else for each c' of c's children do                               (10)
      Similarity_Range_Query(c', T_q, θ)                              (11)
```

Bulk-loading Algorithm(Backup)

⋆ Cone Formation

Input:: 1) $S = \{s_1, s_2, \ldots, s_n\}$:n spatial referenced time series;

2) a maximum threshold of cone angle τ_{max}

Output: Similarity Search Tree with Threaded Leaves

Method:

divide SF into a collection of disjoint cells C;
$index$ = 1;
while ($index$ < $C.size()$)
  $C(index).cener$ = average(time series in $C(index)$);
  $C(index).angle$ = max angle(any time series in $C(index)$, $C(index).cener$));
  if ($C(index).angle$ > τ_{max})
    split cell $C(index)$ into four quarters C_{11}, C_{12},C_{13},C_{14};
    insert four quarters into C at position $index + 1$;
    set C_{11}, C_{12},C_{13},C_{14} as $C(index)$'s children;
  else
    $index$ ++ ;
    insert $C(index)$ at the end of the threaded leaf list;
return C;

Spatial Autocorrelation (Backup)

- ⋆ Tobler's first law of geography:
  - "Everything is related to everything else but nearby things are more related than distant things"
- ⋆ Spatial autocorrelation
  - Nearby objects tend to affect each other
  - Local continuity
- ⋆ Correlogram
  - Estimate of spatial continuity in data
  - Distance vs. Correlation

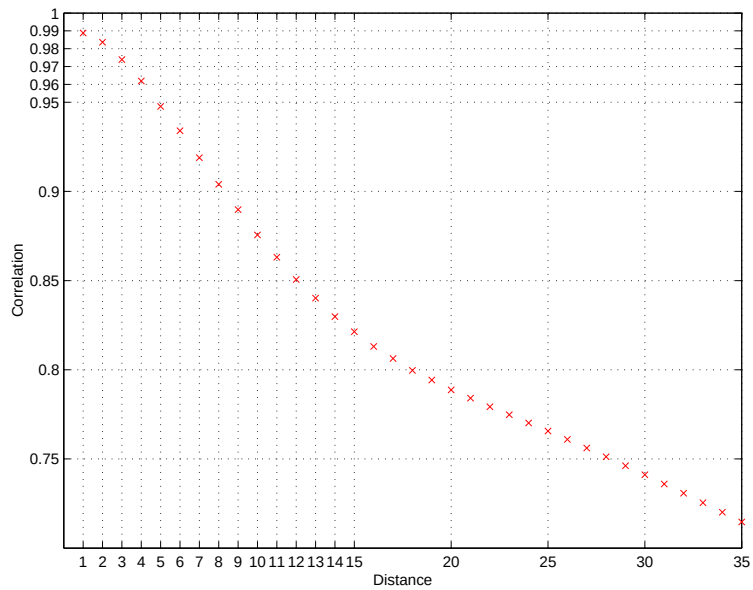


Figure 15: Correlogram for Ocean

Cost Model for Range Query (Backup)

- ⋆ Unit: number of correlation calculations
- ⋆ Number of leaf cones: $|L|$
- ⋆ Cone Selectivity Ratio j = $\frac{\#\,cones\ in\ some{-}true}{total\,\#\,cones}$
  - Assume cone selectivity ratio keep same at every level
- ⋆ Cost:
  - Filtering:
    - Depth-first traversal: $|L| * (1 + \frac{1}{p-1}) * j * t_1$
    - p is number of children
    - t_1 is a constant
  - Refinement:
    - $|L| * j * t_2$
    - t_2 is a constant
- ⋆ Cone selectivity ratio
  - plays an important role in performance
  - directly related to minimal correlation threshold for queries

Experimental Evaluation(cont'd)(Backup)

⋆ Correlogram

- Distance vs. Correlation
- Parameter Selection for Proper Spatial Cone Size

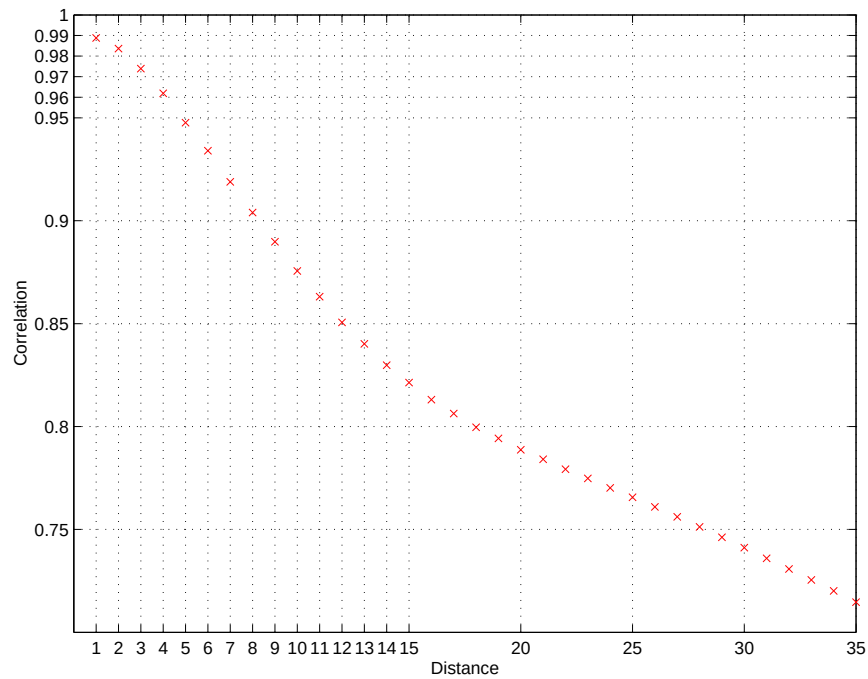


Figure 16: Correlogram for Ocean