JOURNAL OF SPATIAL INFORMATION SCIENCE
Number 6 (2013), pp. 1–42

doi:10.5311/JOSIS.2013.6.94

RESEARCH ARTICLE

Mining sensor datasets with spatiotemporal neighborhoods

Michael P. McGuire[1], Vandana P. Janeja[2], and Aryya Gangopadhyay[2]

[1]Department of Computer and Information Sciences, Towson University, USA
[2]Information Systems Department,University of Maryland, Baltimore County, USA

Received: June 28, 2012; returned: October 8, 2012; revised: December 7, 2012; accepted: December 11, 2012.

Abstract: Many spatiotemporal data mining methods are dependent on how relationships between a spatiotemporal unit and its neighbors are defined. These relationships are often termed the neighborhood of a spatiotemporal object. The focus of this paper is the discovery of spatiotemporal neighborhoods to find automatically spatiotemporal sub-regions in a sensor dataset. This research is motivated by the need to characterize large sensor datasets like those found in oceanographic and meteorological research. The approach presented in this paper finds spatiotemporal neighborhoods in sensor datasets by combining an agglomerative method to create temporal intervals and a graph-based method to find spatial neighborhoods within each temporal interval. These methods were tested on real-world datasets including (a) sea surface temperature data from the Tropical Atmospheric Ocean Project (TAO) array in the Equatorial Pacific Ocean and (b) NEXRAD precipitation data from the Hydro-NEXRAD system. The results were evaluated based on known patterns of the phenomenon being measured. Furthermore, the results were quantified by performing hypothesis testing to establish the statistical significance using Monte Carlo simulations. The approach was also compared with existing approaches using validation metrics namely spatial autocorrelation and temporal interval dissimilarity. The results of these experiments show that our approach indeed identifies highly refined spatiotemporal neighborhoods.

Keywords: spatiotemporal patterns, data mining, sensors, spatial neighborhoods, spatial clustering, discretization, change detection, spatial autocorrelation

1 Introduction

Spatiotemporal data from automatic sensing devices has become prevalent in many domains such as climatology, hydrology, transportation planning, and environmental science and we are currently experiencing a deluge of spatiotemporal data collected at increasingly numerous locations and fine temporal granularities. In the context of this paper, a sensor is defined as a device that automatically measures a physical quantity over time at a location. Finding spatiotemporal patterns in large sensor datasets helps identify physical processes governing the phenomenon being measured. As the data grows over time, spatiotemporal patterns become more difficult to analyze. Another challenge in spatiotemporal data mining is to identify the proper method for neighborhood generation. The spatiotemporal neighborhood is particularly important because it is used to define relationships between a spatiotemporal unit and its neighbors and many spatiotemporal data mining methods are dependent on how the neighborhood is defined [56]. This paper focuses on the discovery of spatiotemporal neighborhoods where, in space, the neighborhood is generally a set of locations that are proximal and have similar characteristics. In time, a neighborhood is a set of time periods that are similar for either a single location or set of locations. Our notion of a spatiotemporal neighborhood is distinct from the traditional notions since we consider both a spatial characterization as well as a temporal characterization of neighborhoods.

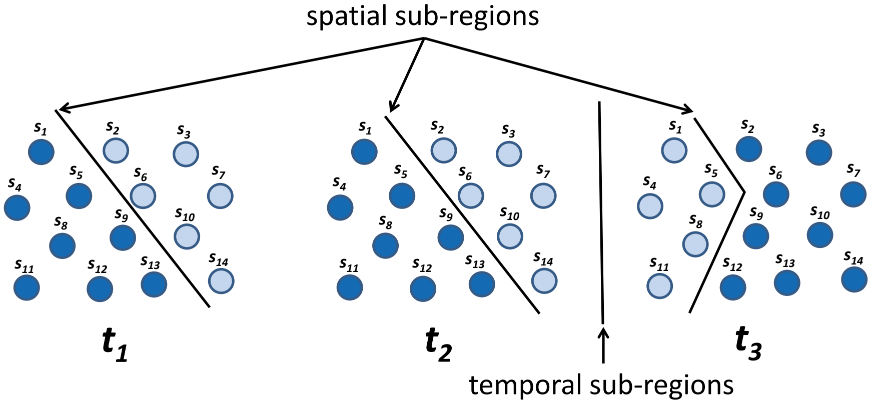


Figure 1: Spatiotemporal sub-regions

Figure 1 shows a simple example to illustrate spatiotemporal sub-regions in a set of sensors measuring temperature at three time periods t_1, t_2, and t_3 where the light sensors are cold and the dark sensors are hot. The spatial neighborhoods in this example are exhibited as the division in space between hot and cold measurements. The temporal neighborhoods are exhibited as the division between time periods where the spatial pattern of temperature changes. The primary objective of this paper is to characterize the pattern in spatiotemporal datasets by combining spatial and temporal sub-regions. The result can be thought of as a “thumb print” of a spatiotemporal dataset that identifies a spatial pattern for a period of time. The pattern of spatiotemporal sub-regions in this example suggests that an event has occurred between times t_2 and t_3 to change the spatial pattern of temperature. Now imagine a much larger dataset than the one depicted in Figure 1 where the pattern of spatiotemporal heterogeneity is much less evident. It becomes a challenge to find the pattern of changing spatiotemporal sub-regions and therefore approaches are needed to automatically uncover these patterns in large sensor datasets.

This paper proposes an approach to identify spatiotemporal neighborhoods to find the inherent pattern of spatiotemporal sub-regions in the data. The resulting characterization of spatiotemporal data can be seen as a key step to knowledge discovery in a number of domains including climatology, hydrology, transportation planning, and environmental science because it provides an automated way to find the homogeneous sub-regions in space and time in the dataset. The resulting regions can then be used in the identification and characterization of events. The following presents a motivating example in the domain of climatology:

1.1 Motivating Example

El Niño events are characterized by anomalously warm sea surface temperatures (SST) in the Equatorial Pacific Ocean and can have important implications for global weather conditions [47]. The TAO/TRITON array [46] consists of sensors installed on buoys positioned in the equatorial region of the Pacific Ocean. The sensors collect a wide range of meteorological and oceanographic measurements. SST measurements are reported every five minutes. Over time, this results in a massive dynamic spatiotemporal dataset. This data played an integral part in characterizing the 1997–98 El Niño [38] and are currently being used to initialize models for El Niño prediction. There have been a number of studies which assimilate meteorological and oceanographic data to offer a description of the phenomena associated with the events of the 1982–83 El Niño [5, 49] and the 1997–1998 El Niño [38]. These analyses show a particular importance in the spatiotemporal patterns of SST anomalies that characterize El Niño events. Understanding these patterns can lead to new knowledge about the global climate and in turn can assist in predicting local weather patterns such as drought and flooding.

As a use case, consider the perspective of a climatologist analyzing the SST data over time. The aim might be to find regions, boundaries, and outliers in the dataset; critical time periods where changes in these regions occur; and relations between these global patterns and local weather conditions. In the current mode of analysis, daily anomalies are typically calculated using a combination of in situ and satellite measurements where the degree of the anomaly is based on the difference between the current SST analysis value and SST monthly climatology. This method finds global outliers at coarse spatial and temporal resolutions, in the order of 1 day [51]. Instead, a climatologist might prefer to develop a more detailed spatiotemporal characterization of SST by using data from the TAO/TRITON network. To begin this analysis, the climatologist must first be able to characterize the evolution of El Niño events by finding distinct points in time where the pattern of SST changes.

For the purpose of this motivational example, consider Figure 2 which shows a time series of satellite measurements of SST for five consecutive days from February 4, 2006 to February 8, 2006. It is evident from visual inspection of the figure that the region between 150°E and 180° experiences a significant amount of change in the spatial pattern of SST. However, given longer time series, it becomes prohibitively difficult to determine where the critical time points exist and where the spatial pattern changes. It is also difficult to determine, using the daily data, the spatiotemporal pattern of the data at a finer temporal resolution.

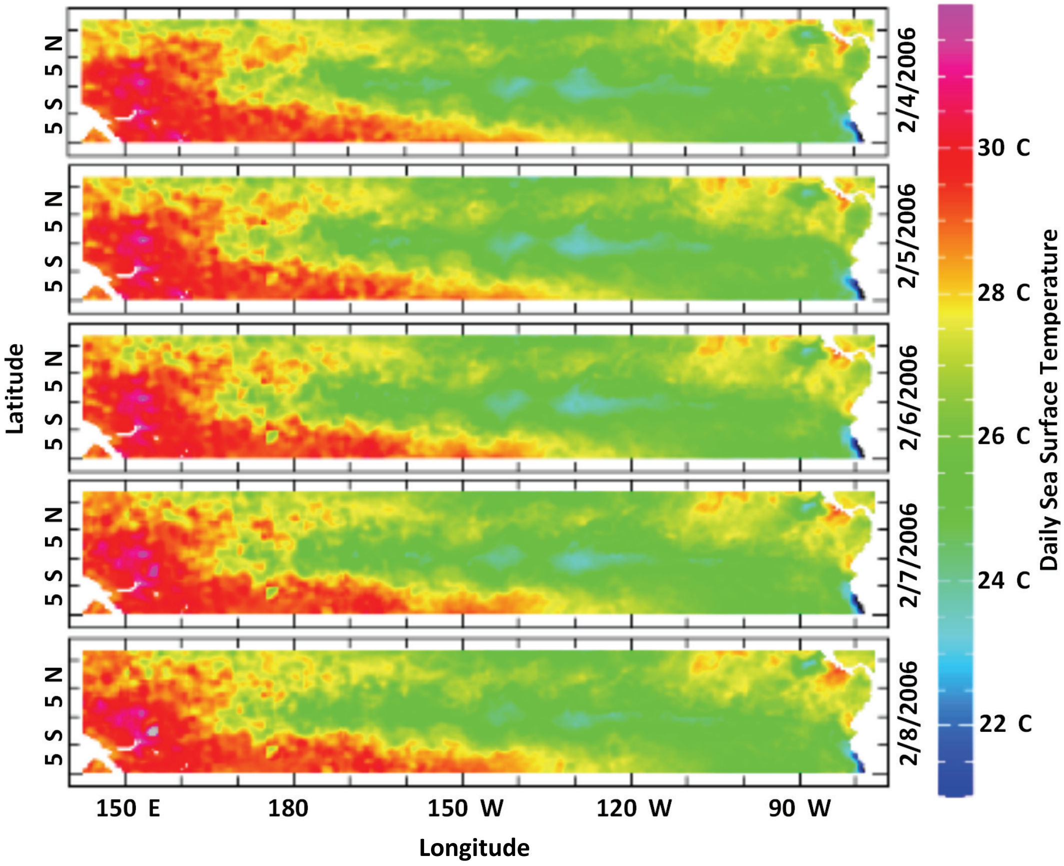


Figure 2: A time series of daily SST between the dates of 2/4/2006 and 2/8/2006 measured by the NOAA AVHRR satellite. The black dots in the figure represent the SST sensors of the TAO/Triton Array. The region between 150°E and 180° experiences a significant amount of change. Figure created using the IRI/LDEO Climate Data Library (http://iridl.ldeo.columbia.edu/).

Figure 3 shows the time series for a selected number of TAO/Triton Array sensors for the time period shown in Figure 2. From this figure, it is evident that the time series for SST exhibits a diurnal pattern where the temperature rises based on solar heating of the water. However, it is also evident that there are interesting areas in the time series that are not identifiable from Figure 2. For example, in Figure 2 there is a spike in SST for a sensor at a). A period of abnormal fluctuations occurs for a group of sensors around b). A single sensor exhibits a large positive shift in SST at c). Figure 3 provides evidence that the pattern of SST exhibits finer temporal granularity than depicted in Figure 2. However, the time series does not show relationships between proximal sensors in space. Furthermore, the example shown here only shows five days of SST measurements where the scale of analysis for El Niño events is over a much larger time period. Therefore it is important to understand the spatiotemporal pattern of SST at a fine temporal resolution over a long period of time.

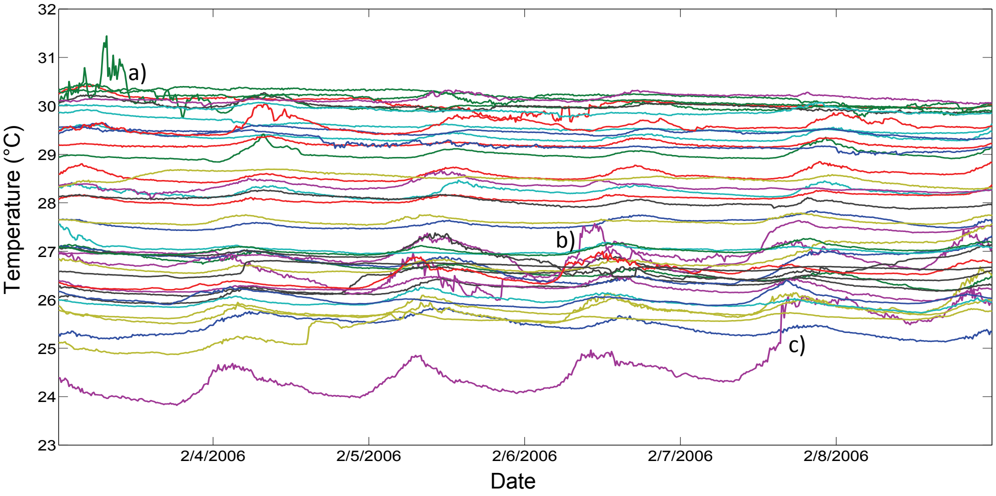


Figure 3: Time series of measurements for the SST sensors shown in Figure 2.

This use case presents a number of challenges. The first challenge is to find proximal sensors in the TAO/Triton network that have similar SST measurements in a particular time frame. To make the analysis more efficient, the climatologist would like to automatically find areas in the data where changes to the spatiotemporal patterns are most likely to occur and focus the analysis on finding anomalies in these areas. For example, by finding these areas, the climatologist could then pinpoint time periods where exploration of the satellite data might prove to be fruitful. Furthermore, finding these areas can lead to the discovery of events that shape the spatiotemporal pattern of SST. If these events can be identified in advance, they can be used in the prediction of global weather conditions such as localized drought and flooding. In short, the climatologist is in need of a mechanism that allows for the spatiotemporal characterization of the natural boundaries in space and time in the sensor network.

The approach to spatiotemporal neighborhoods presented in this paper first determines adjacent spatial nodes then applies an agglomerative method to create temporal intervals for multiple sensors based on spatial relationships between adjacent sensors. Then, a graph-based method is used to create spatial neighborhoods for each interval. The combination of the temporal intervals and spatial neighborhoods results in spatiotemporal neighborhoods.

The rest of the paper is organized as follows. Related research is discussed in Section 2. Section 3 provides the objectives and preliminaries for the discovery of spatiotemporal neighborhoods. Section 4 discusses the approach and algorithms as well as validation metrics. Detailed experimental results are discussed in Section 5. Finally, Section 7 offers some concluding remarks.

2 Related work and contributions

Related works relevant to this research are situated in the areas of spatial neighborhood discovery, time series segmentation, and spatiotemporal pattern discovery.

2.1 Spatial neighborhoods

The model used to determine the neighborhood of a spatial object is a critical step in spatial and spatiotemporal statistical analysis [8]. Spatial neighborhood formation has been identified as a critical challenge in future research in spatial data mining and is a key aspect to spatial data mining techniques [56]. This is never more true than in the case of spatial outlier detection. For example, the issue of graph-based spatial outlier detection using a single attribute has been addressed in [57]. Their definition of a neighborhood is similar to the neighborhood graph [10], which is primarily based on spatial relationships. However the process of selecting the spatial predicates and identifying the spatial relationship can be an intricate process in itself. Another approach generates neighborhoods using a combination of distance and semantic relationships [2]. In general the neighborhoods in these approaches have crisp boundaries and do not take the measurements from the spatial objects into account for the generation of the neighborhoods. The approach presented in this paper extends the crisp neighborhood by using a measure of connectivity strength to assign a degree of membership of spatial nodes to a particular neighborhood. Furthermore, this approach uses the measurements taken at the spatial nodes to initialize neighbor relationships and the number of neighborhoods are not known a priori.

Spatial neighborhood discovery can also be formulated as a spatial clustering problem. Most of the existing clustering methods for spatial data either cluster spatial data alone or treat attributes as another dimension along with spatial dimensions. The approach presented in this paper treats the spatial dimension separately from attribute dimensions in the dataset therefore generating neighborhoods that are constrained by the spatial dimension and defined by the parameter being measured. There have been many applications of various types of clustering algorithms on spatial data [20]. Clustering methods are typically categorized as partitioning-based, hierarchical, density-based, grid-based, or graph-based methods. Partitioning-based methods typically group objects in the data based on a distance from the closest cluster center. Partitioning methods include k-means [35], k-medoids [27], CLARANS [42], and affinity propagation [13].

Hierarchical methods introduced in [23] form a dendrogram of clustered objects by recursively splitting the dataset. Popular hierarchical clustering methods include BIRCH [62], AGNES [27], and CURE [18]. Density-based clustering methods such as DBSCAN [11] and OPTICS [3], instead of simply using distance between objects, form clusters in dense regions of points in the data. Grid-based clustering methods such as STING [60] and WaveCluster [55], allocate all data points into a grid structure and clustering is formed by agglomerations of grid cells. Finally, graph-based methods model the data using a graph structure and clusters are typically formed by using graph partitioning methods [12, 25, 26, 34, 61]. Most closely related to our work is the work on using a Delaunay triangulation for the clustering of spatial objects. In [25] the Delaunay triangulation is used to cluster spatial points based on the connectivity of the triangulation after applying an edge cut based on a spatial distance threshold. In [34] a Delaunay triangulation is used for clustering and boundary detection in spatial datasets. The work presented in this paper builds on this approach. However, instead of applying an edge cut based on a spatial distance threshold, we perform edge cuts based on the difference between the measurements at connected spatial nodes.

2.2 Time series segmentation

The concept of a temporal neighborhood is most closely related to the literature focused on time series segmentation. The methods presented in this paper focus on finding temporal intervals across a number of time series taken at a number of spatial locations. Moreover, this is the first approach to delineate temporal intervals that are based on relationships between adjacent spatial nodes. The existing literature primarily focuses on approximating a time series, and do not result in a set of discrete temporal intervals. Furthermore, the literature has largely focused on segmentation of a single time series. To the authors' knowledge, there is no existing approach that discretizes multiple time series in a spatiotemporal dataset. Numerous algorithms [1,4,21,29,32] have been written to segment time series. One of the most common solutions to this problem applies a piecewise linear approximation using dynamic programming [4]. Three common algorithms for time series segmentation are the bottom-up, top-down, and sliding window algorithms [29]. Another approach, global iterative replacement (GIR), uses a greedy algorithm to gradually move break points to more optimal positions [21]. Abonyi et al. (2003) [1] offer a method to segment time series based on fuzzy clustering. In this approach, principal component analysis (PCA) models are used to test the homogeneity of the resulting segments. Most recently [32] developed a method to segment time series using polynomial degrees with regressor-based costs.

2.3 Spatiotemporal pattern discovery

This paper also has a number of commonalities with literature in spatiotemporal data mining. Many of these approaches first perform a spatial characterization of the data then find the temporal pattern. The work presented in this paper sets itself apart from this literature by first finding temporal intervals in the dataset. Also, a novel aspect of this research is the idea of combining temporal intervals with spatial neighborhoods to find spatiotemporal neighborhoods. A number of works discover spatiotemporal patterns in sensor data [6, 15, 16, 30, 40, 57]. In [57] a simple definition of a spatiotemporal neighborhood is introduced as two or more nodes in a graph that are connected during a certain point in time. Graphs can be used to represent spatiotemporal features for the purposes of data mining. Time-expanded graphs were developed for the purpose of road traffic control to model traffic flows and solve flow problems on a network over time [30]. Building on this approach, George and Shekhar devised the time-aggregated graph, defined as a graph where at each node, a time series exists that represents the presence of the node at any period in time [16]. Spatiotemporal sensor graphs (STSG) [15] extend the concept of time-aggregated graphs to model spatiotemporal patterns in sensor networks. The STSG approach includes not only a time series for the representation of nodes but also for the representation of edges in the graph. This allows for the network which connects nodes to also be dynamic. Chan et al. [6] also uses a graph representation to mine spatiotemporal patterns. In this approach, clustering for spatiotemporal analysis of graphs (cSTAG) is used to mine spatiotemporal patterns in emerging graphs.

There have been a number of approaches to spatiotemporal clustering. In [50] a self organizing map (SOM) neural network is used to find spatiotemporal regions of precipitation data. Another approach improves spatiotemporal clustering by extending the distance measure traditionally used in most clustering algorithms to be a function of the position history of the spatiotemporal objects in the dataset [52]. In [54] a weighted kernel k-means algorithm is proposed to account for problems with nonlinear separability in spatiotemporal data. In [33] a tight clustering algorithm is presented where the clustering is based on a measure of process similarity. While this approach does account for spatiotemporal aspects of the data, it provides a global clustering for an entire time series and therefore, does not uncover changing patterns over time. A number of approaches are focused on finding dense areas or clusters in moving object databases. For example, in [59] spatiotemporal association rules are used to find stationary and high-traffic regions in object mobility databases. In [24] a combination of density-based clustering and time slices are used to find clusters of moving objects in trajectory databases. Finally, in [19] a grid-based technique is used to find dense groups of moving objects across time.

2.4 Contribution of this work

The primary contribution of this paper is the discovery of neighborhood for spatiotemporal data, which is a critical challenge in spatiotemporal statistical analysis [8] and spatiotemporal data mining [56]. The approach presented in this paper discretizes temporal intervals and discovers spatial neighborhoods within each temporal interval to form spatiotemporal neighborhoods. This notion of a spatiotemporal neighborhood is unique because the formation of these neighborhoods is based on both a spatial characterization as well as a temporal characterization. Also, there has yet to be an approach to spatiotemporal neighborhoods that is based on the ability to track relationships between spatial locations over time. Furthermore, experiments were performed on real world datasets on SST and precipitation data with promising results in finding distinct temporal interval and spatial neighborhoods in both datasets. The specific contributions of this research are as follows:

Temporal intervals Temporal intervals embody the concept of neighborhoods in time. One major contribution is the discovery of unequal width or unequal frequency intervals that are robust in the presence of outliers. Furthermore, this is the first approach to temporal intervals that is based on the relationships of measurements taken at adjacent spatial nodes. Lastly, the efficacy of the interval discovery method is demonstrated on very large real world sensor datasets from the TAO/TRITON Array [46] and Hydro-NEXRAD system [31].

Spatiotemporal neighborhoods The spatiotemporal neighborhood method finds groupings of locations in terms of the spatial distribution of measurements based on their relationships with neighboring locations. The spatiotemporal neighborhood approach presented in this paper is conceptually similar to clustering approaches which use a graph created by a Delaunay triangulation [25,34]. However, no existing approach combines temporal intervals with spatial neighborhoods to form spatiotemporal neighborhoods. Furthermore, the approach presented in this paper can accommodate for spatial nodes that are irregularly distributed as well as spatial nodes that are distributed in the form of a grid.

Validation The approaches presented in this paper are validated by using both established metrics as well as new measures for comparison with alternative approaches. The Moran's I statistic, a standard measure of spatial autocorrelation, is used to validate the quality of the spatial contiguity represented by the spatiotemporal neighborhoods. The between interval dissimilarity (*bid*) measures the quality of a set of temporal intervals by calculating the dissimilarity of adjacent intervals. This metric is used to compare our results with other established approaches. The significance of our results is then tested using Monte Carlo simulation.

3 Objectives and preliminaries

In most real-world sensor deployments, a heterogeneous pattern of spatial and temporal dependence exists based on the physical properties of the process being measured. With this in mind, finding the how this pattern is expressed by the formation of homogeneous spatiotemporal sub-regions in the data can lead to the discovery of distinct spatiotemporal sub-regions in the dataset. Considering the motivational example of climatology, finding these naturally occurring boundaries can lead to a better characterization of El Niño events which in turn can lead to the discovery of new impacts on global weather patterns. In Figure 4, the spatial pattern can be determined by grouping the locations into regions based on the measurements taken at each time period. Conversely, the temporal pattern can be determined by grouping the time periods based on the measurements taken at each locations. The goal of spatiotemporal neighborhoods (STN) is to find the spatial and temporal patterns in a dataset by first delineating temporal intervals in a spatiotemporal dataset across all locations, then, for each interval, to determine the spatial pattern in terms of groupings of similar spatial nodes. The specific objectives of STN are as follows:

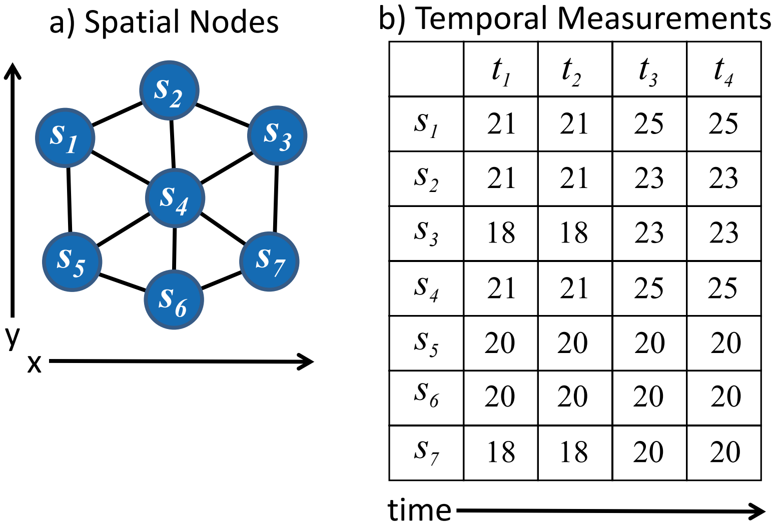
- Find the temporal pattern for a set of spatial nodes S and temporal measurements T by dividing the time series for a set of spatial nodes and temporal measurements into a set of unequal width temporal intervals.
- Find the spatial pattern for a set of spatial nodes S and temporal measurements T by finding the spatial neighborhoods for each temporal interval resulting in spatiotemporal intervals where the number of neighborhoods is not known a priori.

3.1 Sensor datasets: Spatial nodes and temporal measurements

Conceptually, sensor deployments consist of a set of spatial nodes distributed in Euclidean space where each spatial node is associated with a set of measurements taken over time. More formally we consider the following input:

- Let S represent a set of **spatial nodes** where $S = \{s_1, ..., s_n\}$ and each $s_i \in S$ has a set of coordinates in 2D Euclidean space (s_{ix}, s_{iy}).
- Each $s_i \in S$ also has a set of **spatial neighbors** $SN_i \subset S$ that are defined by a **spatial relationship** sr such that given two spatial nodes $(s_p, s_q) \in S$ a spatial relationship $sr(s_p, s_q)$ exists if there is either a distance, direction or topological relationship between them.
- Each $s_i \in S$ has a set of measurements that are taken for a set of time periods $T_i = \{t_{i1}, \ldots, t_{im}\}$ where $t_{i1} < t_{i2} < \cdots < t_{im}$. A **time period** is defined as any individual $t_{ij} \in T_i$.

The example depicted in Figure 4 shows a set of spatial nodes $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ and temperature measurements taken at each spatial node such that for spatial node s_1 the temporal measurements $T_1 = 21, 21, 25, 25$. Spatial relationships are illustrated by lines connecting the spatial nodes with their topological neighbors. For example, spatial node s_4 has a set of spatial neighbors $SN_4 = \{s_1, s_2, s_3, s_5, s_6\}$.



	t_1	t_2	t_3	t_4
s_1	21	21	25	25
s_2	21	21	23	23
s_3	18	18	23	23
s_4	21	21	25	25
s_5	20	20	20	20
s_6	20	20	20	20
s_7	18	18	20	20

Figure 4: A set of spatial nodes a) along with temporal measurements b) for each spatial node shown in table format.

Based on Tobler's first law of geography ("Everything is related to everything else, but nearby things are more related than distant things" [58]), it is assumed that for any spatial node $s_i \in S$ spatial dependence is defined by an sr with its spatial neighbors SN_i such that the temporal measurements T_i of s_i are similar in value to the temporal measurements TN_i of its spatial neighbors SN_i. Similarly, temporal dependence exists between each $t_{ij} \in T_i$ and its temporal neighbors (t_{ij-1}, t_{ij+1}), which represent the temporal measurements taken directly before and after t_{ij}. It can then be assumed that measurements taken at t_{ij-1}, t_{ij} and t_{ij+1} are generally similar.

4 Approach and algorithms

In this section, an approach to spatiotemporal neighborhoods is presented. The approach applies the principals discussed in the previous section. An overview of the STN approach is shown in Figure 5. The approach to spatiotemporal neighborhoods first determines adjacent spatial nodes then applies an agglomerative method to create temporal intervals for multiple sensors based on spatial relationships between adjacent sensors. Then, a graph-based method is used to create spatial neighborhoods for each interval. The combination of the temporal intervals and spatial neighborhoods results in spatiotemporal neighborhoods.

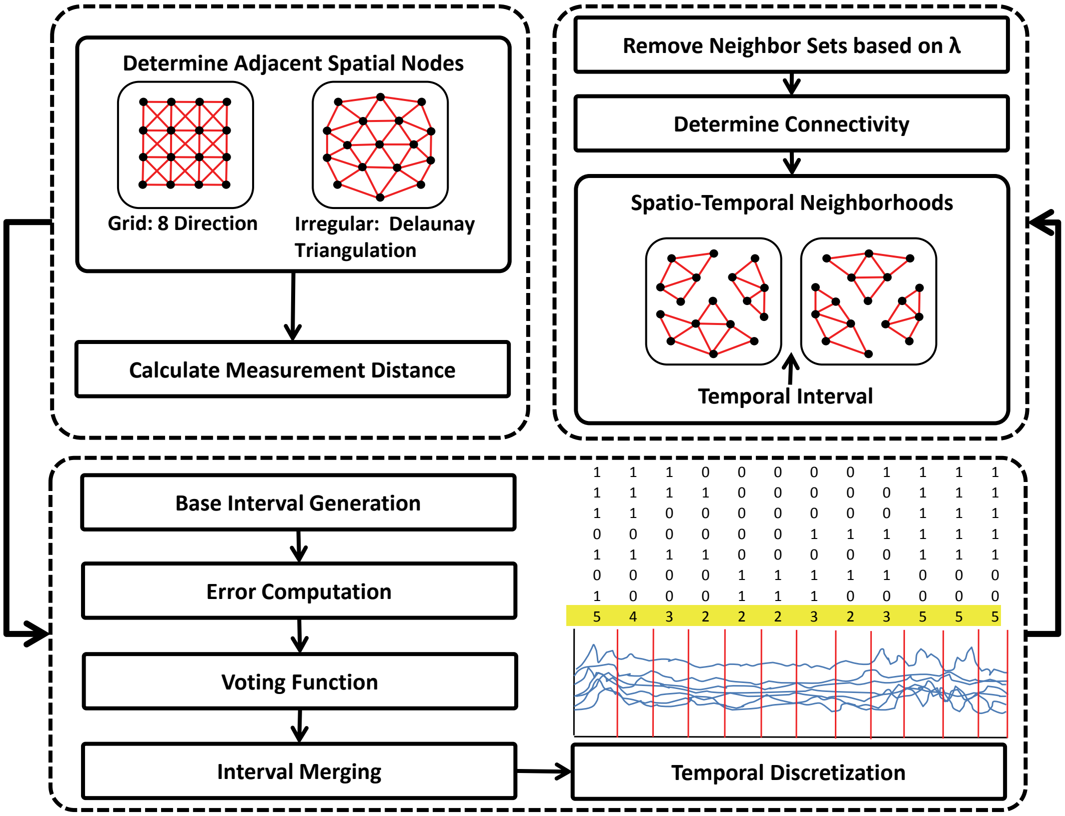


Figure 5: Overview of spatiotemporal neighborhoods (STN) approach.

4.1 Determine adjacent spatial nodes

Finding the spatial pattern first requires the identification of spatial neighbor sets for each spatial node based on a spatial relationship. The methods presented in this paper use a topological spatial relationship of adjacency. The first approach applies to irregularly distributed spatial nodes found in many in-situ sensor datasets where sensors are placed in the field to collect measurements. In this case a Delaunay triangulation (DT) [9] is used to create a network of adjacent spatial nodes. A DT creates a triangulation of the spatial nodes such that the adjacent nodes are connected by non-intersecting edges. Figure 6 shows an example triangulation where a) represents a set of irregularly spaced spatial nodes and b) shows the resulting triangulation and the spatial nodes that are adjacent to node s_4.

The DT is used to determine the local neighborhood of adjacent spatial nodes for irregularly spaced spatial nodes. The triangle-based adjacency relationship sr_{tri} is defined as follows:

Definition 1 (Triangulation-based adjacency)**.** *Given a set of irregularly distributed spatial nodes $S = \{s_1, ..., s_n\}$ and a Delaunay triangulation (DT) of S where for any $s_i \in S$ the triangulation-based adjacency sr_{tri} comprises of a set of neighboring spatial nodes $SN_i \subset S$ that are immediately adjacent to s_i by a single edge of the DT.*

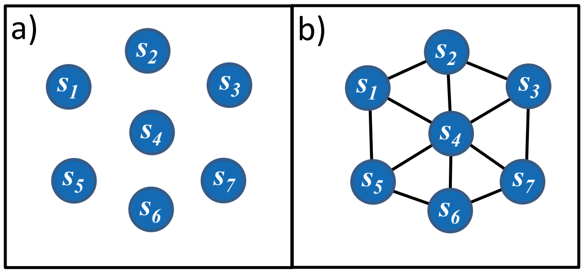


Figure 6: Triangulation of irregularly spaced spatial nodes where a) depicts a set of irregularly distributed spatial nodes and b) depicts the triangulation-based adjacency for the set of spatial nodes.

Certain assumptions are made in creating the DT [48]. First, spatial nodes are assumed to be non-collinear. Second, at least three points are required to create the triangulation. Third, any given four nodes are non-co-circular, or in other words, it is assumed that there are no four nodes on a circle. The DT captures the underlying spatial relationships between nodes using an adjacency matrix, a structure for accommodating spatial autocorrelation.

The second approach applies to spatial nodes that are distributed as a regular grid, such as those found in remotely sensed data. In this case, the centroids of the surrounding grid cells in eight directions are used such that the adjacent spatial nodes to any spatial node distributed on a grid are the nearest spatial nodes to the north, south, east, and west, northeast, northwest, southeast, and southwest. This is also known as the Moore neighborhood in the field of cellular automata and 8-connected pixels in computer graphics. For the purpose of this paper, the eight direction adjacency sr_{d8} is defined as follows:

Definition 2 (Eight direction adjacency)**.** *Given a set of spatial nodes $S = \{s_1, ..., s_n\}$ that are distributed on a regular grid such that each $s_i \in S$ is equidistant in the x and y direction. The eight direction adjacency sr_{d8} then consists of a set of spatial neighbors $SN_i \subset S$ that are immediately adjacent to s_i in the x, y, or diagonal directions.*

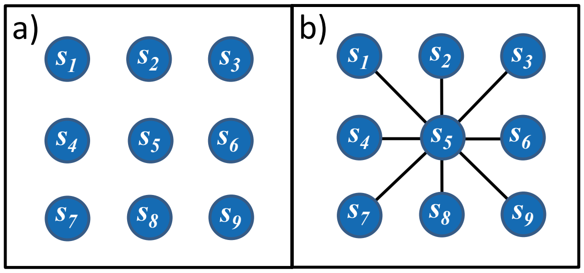


Figure 7: Eight direction adjacency for spatial nodes on a regular grid where a) depicts a set of spatial nodes distributed in a grid and b) shows the 8-direction adjacency for spatial node s_5.

An example of an eight direction neighborhood is shown in Figure 7 where a) shows a set of spatial nodes distributed on a regular grid and b) shows the eight dimension adjacency for node s_5. In the case of irregularly distributed spatial nodes, the triangulation-based spatial adjacency is used. Alternatively, in the case of spatial nodes distributed on a grid, the 8 direction spatial adjacency is used. We also use the concept of measurement distance *md* in this step because *md* is used in the next step to create temporal intervals.

4.1.1 Measurement distance

Finding the spatial pattern of a phenomenon measured by a sensor network also requires a method to evaluate differences between measurements taken at adjacent spatial nodes. In the motivational example, comparing measurements between adjacent SST sensors allows the climatologist to find where the spatial boundaries exist in the dataset. The measurement distance or *md* accomplishes this goal by calculating the Euclidean distance between measurements in attribute space taken at adjacent spatial nodes. More formally, *md* is defined as follows:

Definition 3 (Measurement distance)**.** *Given a spatial node s_i and a set of adjacent spatial nodes $SN_i \subset S$. The measurement distance md(s_i, SN_i) is the normalized Euclidean distance of a set of temporal measurements T_i and T_n between s_i and all SN_i such that:*

$$md = \frac{\sqrt{\sum_1^n (t_i - t_n)^2}}{n}$$

where $t_i \in T_i$ and $t_n \in T_n$ and n is the number of adjacent spatial nodes to s_i.

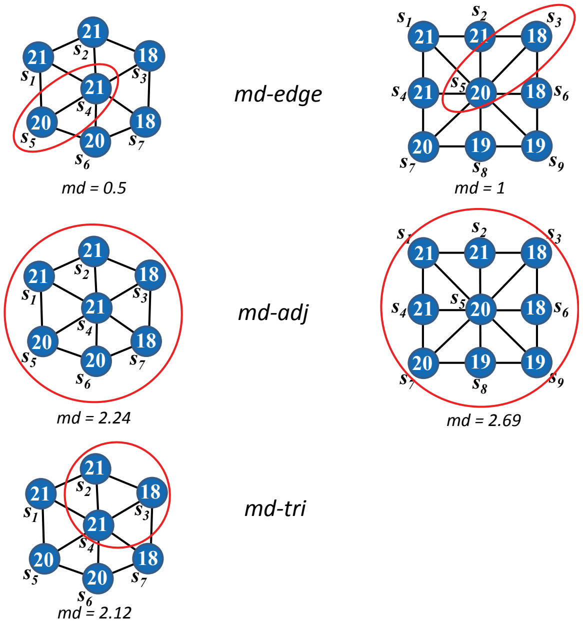


Figure 8: Illustration of possible methods used to calculate *md*.

The *md* is calculated for each spatial node and its adjacent neighbors. In this paper, a number of different methods are used to calculate *md*. These methods are generally based on different ways to combine a spatial node with its spatial neighbors. Figure 8 gives an illustration of the various ways in which *md* can be calculated. The first, and most obvious, calculates the distance between measurements individually for each adjacent spatial node. This is referred to as *md-edge* and can be calculated for both sr_{tri} and sr_{d8}. Other ways to calculate *md* are based on whether the triangulation-based adjacency or eight direction adjacency is used. In the case of sr_{tri}, *md* can be calculated between a spatial node and adjacent nodes for each triangle. This is referred to as *md-tri*. Finally for both the sr_{tri} and sr_{d8}, *md* can be calculated between a spatial node and all adjacent neighbors referred to as *md-adj*. The various approaches to calculating *md* are compared in Section 5. Table 1 shows the *md* values calculated at the *md-edge* level for the example set of spatial nodes and temporal measurements depicted in Figure 4.

	t_1	t_2	t_3	t_4
$md(s_1, s_2)$	0	0	0	0
$md(s_1, s_4)$	0	0	0.75	0.75
$md(s_1, s_5)$	0.5	0.5	0	0
$md(s_2, s_3)$	1.5	1.5	0.25	0.25
$md(s_2, s_4)$	0	0	0.75	0.75
$md(s_3, s_4)$	1.5	1.5	0.5	0.5
$md(s_3, s_7)$	0	0	0.25	0.25
$md(s_4, s_5)$	0.5	0.5	0.75	0.75
$md(s_4, s_6)$	0.5	0.5	0	0
$md(s_4, s_7)$	1.5	1.5	0.75	0.75
$md(s_5, s_6)$	0	0	0.75	0.75
$md(s_5, s_6)$	1	1	0.75	0.75

Table 1: Table of *md* values for the example in Figure 4. Values calculated using *md-edge*.

The algorithm for finding adjacent spatial nodes is shown in Algorithm 1. The algorithm takes as input a set of spatial nodes S and for each set spatial node $s_i \in S$ a set of temporal measurements T as well as an argument determining whether the spatial nodes are irregularly distributed or distributed on a grid. A conditional statement in line 2 determines if the points are irregularly distributed and if so, a DT is applied shown in line 3 and *md* is calculated in lines 4–8. If the points are distributed in a grid, the eight direction adjacency is applied in lines 6–16 and *md* is calculated in lines 17–21.

4.2 Delineate agglomerative temporal intervals

In this approach, an agglomerative method is used to create the temporal intervals. The agglomerative procedure first divides the time series into a set of small equal frequency temporal intervals then calculates the error for each base interval. Then an agglomerative method to combine adjacent high and low error base temporal intervals across the entire set of spatial nodes. A formal definition of a temporal interval is as follows:

Definition 4 (Temporal interval). *Given a set of spatial nodes* $S = \{s_1, ..., s_n\}$ *and a set of adjacent spatial neighbors* $SN_i \subset S$ *with a set of temporal measurements* $T_i = \{t_{i1}, \ldots, t_{im}\}$ *where* $t_{i1} < t_{i2} < \cdots < t_{im}$, *a set of temporal intervals* $INT = int_1, \ldots, int_r$ *where each temporal interval* $int_k = \{t_{i1}, \ldots, t_{ik}\}$ *is a division of* T *based on an* sr *between measurement*

Algorithm 1 Procedure: Determine adjacent spatial nodes

```
Require: A set of spatial nodes $S$, a set of temporal measurements $T$
Ensure: The set of adjacent spatial nodes for each $s_i \in S$ and
    calculate the measurement distance md for each $t_j \in T$
if irregular points then
    Delaunay Triangulation($S$)
    for all $S_n \in S$ do
        for all $t_j \in T$ do
            //Calculate measurement distance for each set of adjacent spatial nodes
            Calculate md
        end for
    end for
else if grid points then
    for all $s_i \in S$ do
        //find 8 adjacent neighbor cells
        $pnt_i$ = replicate $s_{ix}$ and $s_{iy}$ for the number of $s_i$ in $S$
        //calculate distance using replicated matrix of point $i$ from $S_{xy}$ of all $x$ and $y$ coords in $S$
        $dist = \sqrt{(pnt_{ix} - S_x)^2 + (pnt_{iy} - S_y)^2}$
        //get the distance value between adjacent cells in $x$ and $y$ directions
        $dist_{xy} = MIN(dist) > 0$
        //get the distance value between adjacent cells in diagonal directions
        $dist_{diag} = MIN(dist) > dist_{xy}$
        //assign adjacent cells based on difference value
        $adj_{d8} = dist == dist_{xy}$ or $dist == dist_{diag}$
    end for
    for all $S_n \in S$ do
        for all $t_j \in T$ do
            //Calculate measurement distance for each set of adjacent spatial nodes
            Calculate md
        end for
    end for
end if
```

values at all s_i and $SN_i \in S$ and $int_k \subset T$ where each int_k=$< int_k^{start}, int_k^{end} >$ and the size $int_k^{size} = (int_k^{end} - int_k^{start})$ where for any any two intervals int_a, int_b, $int_{size}^a \neq int_{size}^b$.

The agglomerative method begins with a base set of intervals $INT^{base} = \{int_1^{base}, \ldots, int_h^{base}\}$ where the size INT_{size}^{base} is a user defined parameter which largely depends on the domain and granularity of the analysis. A heuristic method can be used to set INT_{size}^{base} where initially a large INT_{size}^{base} is used. Then the resulting intervals are evaluated while iteratively decreasing the INT_{size}^{base} until a satisfactory set of intervals are found. After the base intervals are created, the next step is to calculate the amount of error within each base interval. For this purpose we use the sum of squared error (SSE) to identify high and low error base intervals to be merged. The SSE is calculated based on the measurement distance values for each set of adjacent spatial nodes. The SSE for the *md* values for each spatial node in a given int^{base} is calculated as follows:

$$\text{SSE} = \sum_{1}^{n} (md_i - \overline{md})^2$$

where n is the number of spatial nodes, $dist$ is the Euclidean distance, and $\overline{md}$ is the mean of all values within a base interval.

The agglomerative temporal intervals are based on the matrix of SSE values calculated for each int^{base} such that intervals with similar SSE values are merged. Figure 9 shows a conceptual set of intervals for the example spatial nodes and temporal measurements shown in Figure 4. In this figure the INT^{base}_{size} is 1 so that the measurements taken at each time period fall into their own interval. The SSE measurements are shown as a time series for each set of *md* values in each int^{base}. Temporal heterogeneity in this example is evident between times t_2 and t_3 and the agglomerative temporal intervals merge base intervals int^{base}_1 and int^{base}_2 to form temporal interval int_1 and base intervals int^{base}_3 and int^{base}_4 to form temporal interval int_2.

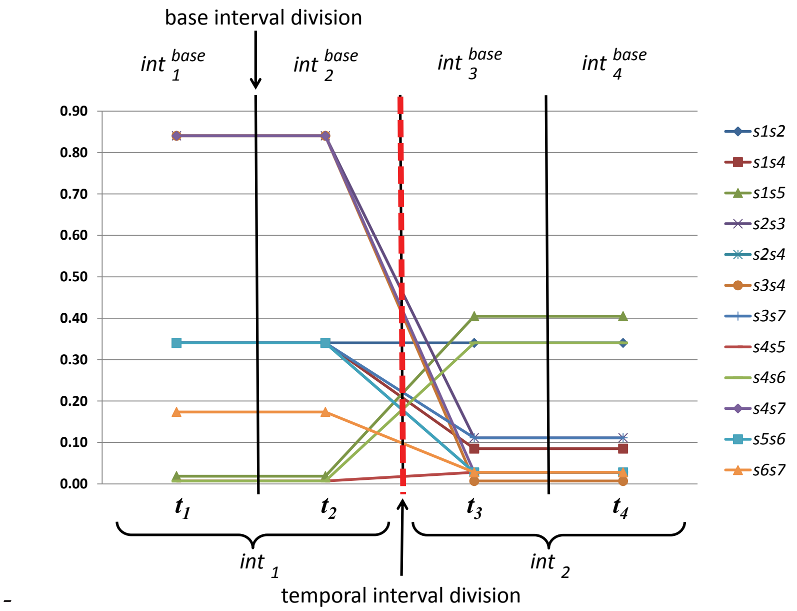


Figure 9: Illustration of conceptual temporal interval divisions for SSE values for the spatial nodes and temporal measurements shown in Figure 4 based on pairs of spatially related sensors. In this example, there are four base intervals int^{base}_1 through int^{base}_4. The graph shows the SSE values for each pair of spatial nodes. Notice that the SSE values change between int^{base}_2 and int^{base}_3 creating a spatiotemporal interval division with the resulting spatiotemporal intervals represented by int_1 and int_2.

After the base intervals are created, the next step is to classify each int^{base} as having a high or low error based on the SSE values. In this step a threshold λ is applied for each int^{base} for each spatial node. This results in a count ϵ of int^{base} with SSE $> \lambda$ for each spatial node. Then a threshold mv is applied to ϵ to determine intervals with a large number of spatial nodes with SSE $> \lambda$. Finally, consecutive high and low SSE intervals are merged resulting in a set of unequal width temporal intervals. A heuristic approach is used to set the λ and mv thresholds by initially using the mean SSE for λ and the mean ϵ for mv. The sensitivity of both thresholds is tested by progressively adding one standard deviation and evaluate the resulting intervals.

The algorithm for delineating temporal intervals is shown in Algorithm 2. The algorithm takes as input a set of spatial nodes S and a set of temporal measurements T for all $s_i \in S$, a base interval size INT_{size}^{base}, an error threshold λ, and a voting function threshold mv. In lines 1–8 the base temporal intervals are created and the SSE is calculated for each interval. The voting function is applied in lines 9–21 where ϵ is calculated for each int_{base} in lines 10–14 and the mv threshold and interval merging are applied in lines 15–21.

Algorithm 2 Procedure: Delineate temporal intervals

```
Require: a set of spatial nodes $S$, a set of temporal measurements $T$,
    a base interval size $INT_{size}^{base}$, a SSE threshold $\lambda$,
    and a voting threshold $mv$
Ensure: temporal intervals for $S$ and $T$ based on $int_{size}^{base}$,$\lambda$, and $mv$
 1: //Create base temporal intervals and calculate SSE
 2: Interval Start = 1
 3: Interval End = Interval Start + $int_{size}^{base}$
 4: while Interval Start $<$ count($t_j \in T$) do
 5:     CALCULATE SSE
 6:     Interval Start = Interval End + 1
 7:     Interval End = Interval Start + $int_{size}^{base}$
 8: end while
 9: //Apply Voting Function
10: for all $int_k \in INT_{base}$ do
11:     $\epsilon = 1$
12:     for all $s_i \in S$ do
13:         if SSE $> \lambda$ then
14:             $\epsilon = \epsilon + 1$
15:         end if
16:     end for
17:     //Apply $mv$ threshold and merge intervals
18:     if $\epsilon > mv$ and $\epsilon_{old} < mv$ then
19:         Output Interval Start, Interval End
20:     else if $\epsilon < mv$ and $\epsilon_{old} > mv$ then
21:         Output Interval Start, Interval End
22:         $\epsilon_{old} = \epsilon$
23:     end if
24: end for
```

4.3 Spatiotemporal neighborhood discovery

Once the temporal intervals are discovered the spatial pattern can be explored further by finding groupings of similar spatial nodes for a particular temporal interval. The spatial groupings combined with the temporal interval configuration discussed above is termed the spatiotemporal neighborhood and represents a set of divisions in time and space where boundaries occur in a spatiotemporal dataset. With this in mind, the spatiotemporal neighborhoods allow the climatologist to first identify an event, then analyze the spatial pattern of SST before and after the event. More formally, a spatiotemporal neighborhood is defined as follows:

Definition 5 (Spatiotemporal neighborhood)**.** *Given a set of spatial nodes $S = \{s_1, ..., s_n\}$, where each $s_i \in S$ has a set of adjacent spatial neighbors $SN_i \subset S$ as well as a set of temporal intervals $INT = \{int_1, \ldots, int_r\}$ where each temporal interval $int_k = \{t_{i1}, \ldots, t_{ik}\}$ the spatiotemporal neighborhood STN for any $int_k \in INT$ is a connected set of graph components represented by an adjacency matrix C of adjacent spatial nodes where for any edge in C, md $< \delta$ where δ is a threshold.*

In this definition, a threshold δ is introduced to remove edges that connect spatial nodes that do not have substantially similar temporal measurements. The δ threshold is generally a heuristic and depends largely on the spatial configuration of the sensor network as well as the nature of the phenomenon being measured.

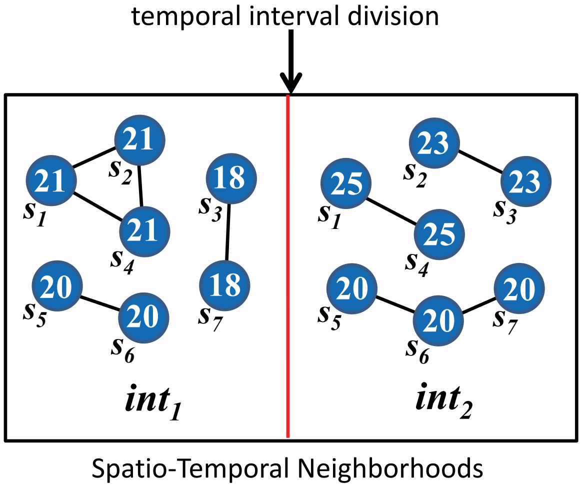


Figure 10: Spatial neighborhoods and temporal neighborhoods combined to form spatiotemporal neighborhoods for spatial nodes s_1 ,s_2, s_3 ,s_4, s_5, s_6, and s_7 and time periods t_1 , t_2 ,t_3, and t_4.

Figure 10 shows the spatiotemporal neighborhood configuration for the example spatial nodes and temporal measurements shown in Figure 4 where the spatiotemporal neighborhoods are depicted as the combination of the temporal intervals and spatial neighborhoods. As would be expected, the spatial configuration of the spatial neighborhoods changes for int_1 and int_2.

The spatiotemporal neighborhoods can be explored at a more local level by analyzing the connectivity strength based on the number of edges connecting each spatial node with its spatiotemporal neighborhood. The connectivity strength of a spatial node for any given temporal interval can be measured by counting the number of edges that connect each spatial node in the adjacency matrix C. The connectivity strength is defined as follows:

Definition 6 (Connectivity strength)**.** *Given a set of spatiotemporal neighborhoods $STN \subset S$ represented by a set of connected spatial nodes during a particular temporal interval int_k, the connectivity strength is defined as the number of edges connecting the spatial node to its spatiotemporal neighborhood. Any spatial node connected by at least 3 edges is considered to be strongly connected; any node connected by 1 or 2 edges is considered to be weakly connected; and any spatial node that is otherwise is considered to be disconnected.*

For the example shown in Figure 10 all the spatial nodes are weakly connected because no spatial node is connected by at least 3 edges. Also, there are no completely disconnected nodes. This suggests that a high level of spatial heterogeneity exists for both intervals in the spatial process represented in this figure.

The algorithm for discovering spatiotemporal neighborhoods is presented in Algorithm 3. The algorithm takes as input a set of spatial nodes S and for each $s_i \in S$ a set of temporal measurements T and a set of adjacent spatial nodes SN_i, a set of temporal intervals INT, and a measurement distance threshold δ. The *md* is calculated in lines 3–5. The λ threshold is applied in lines 6–8. The resulting set of spatial neighbors are added to the adjacency matrix C in lines 9–11.

Algorithm 3 Procedure: Discover spatiotemporal neighborhoods

```
Require: a set of spatial nodes$S$, a set of temporal
    measurements $T$, a set of adjacent spatial nodes $SN$, a set of
    temporal intervals $INT$, a measurement distance threshold $\delta$
Ensure: Spatiotemporal neighborhoods for $S$ and $T$ for each
    interval $INT$ based on threshold $\delta$
 1: //discover spatiotemporal neighborhoods
 2: for all $int_k \in INT$ do
 3:     for all $SN \in S$ do
 4:         //calculate measurement distance for each $SN$
 5:         Calculate md
 6:     end for
 7:     //incrementally remove neighbor sets based on $\delta$
 8:     while $MAX(adj_{md}) > \delta$ do
 9:         $SPN = S_n < max(adj_{md})$
10:     end while//create adjacency matrix $C$
11:     for all $s_i \in SN$ do
12:         Add to $C$
13:     end for
14: end for
```

4.3.1 Order Invariance of Approach

The STN algorithm is order invariant in that it will result in the same spatial neighborhoods regardless of the starting spatial node. The following offers a formal proof of this property:

Theorem 1. *For a set of spatial nodes S, Algorithm 3 will result in the same set of spatial neighborhoods regardless of the starting spatial node.*

Proof. The property of order invariance is proven by contradiction. Assume to the contrary that given a set of spatial nodes S and two spatial nodes s_p and $s_q \in S$ that produce two spatial graphs sg_p and sg_q each with a set of edges and nodes $\langle e_p, n_p \rangle$ and $\langle e_q, n_q \rangle$ respectively that result in two sets of spatial neighborhoods $SPN_p = \{spn_1^p, \ldots, spn_l^p\}$ and $SPN_q = \{spn_1^q, \ldots, spn_l^q\}$ derived from each graph. Because both the 8 direction and triangulation adjacencies produce a connected graph where every spatial node $s \in S$ is connected to every other spatial node $s \in S$ via a path $p \subset e$, the resulting set of edges and nodes $\langle e_p, n_p \rangle$ = $\langle e_q, n_q \rangle$ and therefore, sg_p = sg_q and SPN_p = SPN_q contradicting our assumption. □

4.4 Complexity analysis

When using *md-tri* or *md-adj* the complexity of algorithm 1 is $O(NM)$ where N is the number of spatial nodes and M is the number of temporal measurements. When using *md-edge* the complexity of Algorithm 1 is $O(EM)$ where E is the number of edges in the Delaunay triangulation or 8 dimension adjacency and M is the number of temporal measurements. The complexity of Algorithm 2 is $O(2N/I)$ where N represents the number of $t_j \in T$ and I represents the number of base intervals. The complexity of Algorithm 3 is $O(2NI)$ where I is the number of temporal intervals and N is the number of spatial nodes. Therefore, the overall complexity of the approach is $O(NM)+O(2N/I)+O(2NI) = O(5NM) = O(NM)$.

4.5 Validation metrics

A number of validation measures have been selected to ensure the efficacy of our approach and to compare it to other approaches. This includes the between interval dissimilarity for the temporal intervals, Moran's I for spatiotemporal neighbors, and significance testing of our results using Monte Carlo simulation.

4.5.1 Temporal intervals (between interval dissimilarity)

The objective of the temporal interval approach is to divide the time series into discrete intervals that are not similar to their neighboring intervals. Therefore, a method has been devised to measure the dissimilarity between adjacent intervals. First, the interval dissimilarity is measured across a set of consecutive temporal intervals by calculating the moving difference between each temporal interval. The moving difference is defined as follows:

Definition 7 (Moving difference)**.** *Given a set of temporal intervals* $INT = \{int_1, ..., int_n\}$ *where each interval has a set of temporal values* $T_i = \{t_{i1}, \ldots, t_{in}\}$ *the moving difference* mvd *is an* $n-1$ *vector of the absolute difference between the mean of the values for each pair of intervals such that:* $mvd = \left\{|\mu^{int_1} - \mu^{int_2}|, ..., |\mu^{int_{n-1}} - \mu^{int_n}|\right\}$.

Then, the between interval dissimilarity is calculated by taking the sum of mvd divided by n. The between interval dissimilarity bid is defined as follows:

Definition 8 (Between interval dissimilarity)**.** *Given a set of temporal intervals* $INT = \{int_1, ..., int_n\}$ *the between interval dissimilarity bid is the average* mvd *between the interval means such that bid* $= \frac{\sum_i^{n-1} |\mu^{int_i} - \mu^{int_{i+1}}|}{n}$ *where* n *is the number of intervals.*

The bid is then used to evaluate the global dissimilarity across a set of temporal intervals. These validation metrics are used to compare the performance of our algorithm to a number of other approaches.

4.5.2 Spatiotemporal neighborhoods (Moran's I)

The purpose of the spatiotemporal neighborhood approach is to find the spatial pattern of a phenomena measured at a set of spatial nodes. This approach is based on the assumption that there exists spatial dependence between some or all of the spatial nodes. Two spatial nodes s_p and s_q are considered to be spatially dependent when the variance of temporal measurements t_p and t_q is best explained by a spatial relationship sr. Most often there exist distinct regions of spatially dependent nodes. For example, given a set of spatial nodes S and two subsets of spatial nodes represented by spatial neighborhoods STN_1 and $STN_2 \subset S$ the pattern is heterogeneous if the spatial dependence of the nodes in STN_1 is distinct from the spatial dependence of the nodes in STN_2. This heterogeneous pattern can be caused by regions in the underlying geographical process. For example, in the SST data there are regions of warm and cool water in the Pacific Ocean that form a heterogeneous pattern.

Spatial autocorrelation can be used to measure the degree of spatial dependence of a set of spatial nodes. There are three types of spatial autocorrelation; positive where neighboring values are similar, negative where neighboring values are dissimilar, and zero where there is no spatial dependence whatsoever. The Moran's I statistic [41] and the Geary's C statistic [14] are the most commonly used measures of spatial autocorrelation. Both of these measures are based on a spatial contiguity matrix also known as a spatial weights matrix. It is a well known fact that measures of spatial autocorrelation are heavily dependent on the neighborhoods defined by the spatial contiguity matrix [8]. Therefore, we use a measure of spatial autocorrelation to test the quality of our spatiotemporal neighborhood approach as represented by the contiguity matrix C. The assumption that we make in choosing this method is that higher Moran's I values signify that a neighborhood approach has done a better job at determining the appropriate neighborhood relationships as represented by a contiguity matrix. Spatial autocorrelation is used because it takes into account the spatial structure of a region as well as the temporal measurements of the spatial nodes. Specifically, the I statistic was chosen because the variance of I is less affected by the distribution of the sample data [7]. The I statistic essentially measures the covariance between the temporal measurements at two spatial nodes [17] and is formally defined as follows:

Definition 9 (I statistic)**.** *Given a set of spatial nodes S and their attributes A and a contiguity matrix C defined by a set of spatial neighborhoods $SPN = \{spn_1, ..., spn_k\}$ the I statistic is defined as follows: $I = \frac{N}{\sum_i \sum_j C_{ij}} \frac{\sum_i \sum_j C_{ij}(t_i - \bar{t})(t_j - \bar{t})}{\sum_i (t_i - \bar{t})^2}$ where $\bar{t}$ is the mean of the attributes for all spatial nodes. If $I = -1$ then there exists negative spatial autocorrelation, if $I = 0$ then there exists no spatial autocorrelation, and if $I = 1$ then there exists a positive spatial autocorrelation.*

Therefore, if the I statistic is close to 1 the spatiotemporal neighborhood method has performed well in grouping sets of similar spatial nodes and has found the pattern of spatial dependence in the dataset. The value of the I statistic for a given interval will vary based on the amount of spatial correlation present in the dataset. The I statistic is a logical For validation purposes, the goal is to create a contiguity matrix that maximizes the value of I.

4.6 Significance testing

The significance of the STN approach can be tested using Monte Carlo simulations [39]. Monte Carlo simulation has many uses including risk analysis and the simulation of mathematical and physical systems to name a few. For the significance testing we adapt an approach to Monte Carlo simulation for significance testing used in [22] where the goal of the test is to determine that the spatial neighborhoods, temporal intervals, and the spatiotemporal neighborhoods identified in our approach are significant and not occurring randomly. This approach is used to find the significance of the results by finding the probability that the result could occur randomly. This probability value or "p-value" is calculated using

Monte Carlo simulation. In this case the data is randomized and the algorithm is run and validation metrics are calculated on the randomized dataset for a large number of simulations. For each component the null hypothesis H_0 states that the results are random and the alternative hypothesis H_A states that the resulting temporal intervals validated by the between interval dissimilarity bid_O and spatiotemporal neighborhoods validated by the I statistic I_0 are not random.

In every case described above the actual measures I_O and bid_O are calculated and Monte Carlo simulation measures I_r or bid_r are calculated for each iteration where the subscript r represents the iteration. The measures for all simulations are sorted in descending order. Then, where the measures I_O and bid_O fall in this ranking, determines the p-value by calculating the ranking divided by the number of simulations. In all cases a p-value of < 0.05 (5%) is significant in that it is within the 95% confidence interval and therefore, the null hypothesis H_0 is rejected.

5 Experimental results

In this section the results of experiments are presented where the spatiotemporal neighborhood approach is tested on real-world datasets including SST data for the equatorial Pacific Ocean and precipitation data for a watershed in Baltimore, Maryland, USA. The approaches were qualitatively validated empirically by providing ground-truth validations that show how finding the spatiotemporal neighborhoods in a dataset can lead to the discovery of interesting events. The approaches were also quantitatively compared with other approaches using the Moran's I and bid validation metrics along with the results of the significance testing using Monte Carlo simulation. The final part of this section discusses experiments to test the scalability of the approach.

5.1 Datasets

Experiments were performed on two datasets. The following provides a detailed description of the datasets.

SST data SST data was retrieved from the TAO Project data delivery website [46]. High resolution data (10 minute average) was downloaded for the entire year of 2006. This consisted of data from 55 sensors, 13 of which were missing an extensive number of time periods, and 42 had a full record for the year and had no-data values where measurements were missing. Therefore, 42 sensors that had a full record were used in the experiment. The dataset consisted of 52,563 temporal measurements for each spatial node resulting in a total of 2,207,646 data points.

Precipitation data Precipitation data was retrieved from the Hydro-NEXRAD system [31]. The data was in grid format where each grid cell maps directly to a NEXRAD cell. For the purpose of these experiments, the center of each grid cell is treated as an individual sensor.

The data was extracted for the Gwynns Falls Watershed which lies to the west of Baltimore, Maryland, USA. The dataset consisted of 198 grid cells and 33,492 temporal measurements for each spatial node resulting in a total of 6,631,416 data points.

5.2 Setting thresholds

The spatiotemporal neighborhoods require a number of thresholds to be set at initialization. For this experiment we used a number of heuristic based approaches to set the thresholds. The temporal intervals require a base interval size INT_{size}^{base}. For this threshold a large INT_{size}^{base} is first chosen and iteratively decreased until a satisfactory discretization is found. The λ and mv thresholds are initialized with the mean SSE and mean ϵ respectively. The sensitivity of both thresholds is tested by progressively adding one standard deviation. The resulting intervals are visually inspected at each step until a satisfactory set of intervals is found. A similar approach is taken for the δ threshold applied to *md* in the spatiotemporal neighborhood step where the sensitivity of the threshold is tested by adding one standard deviation to the mean *md* for all spatial nodes. The results are then visually inspected at each step until satisfactory neighborhoods are found.

5.3 Temporal intervals

In this section the results for the discovery of temporal intervals are presented. This section begins with a presentation of the empirical results for the SST and precipitation data. Then the temporal interval approach with a other approaches including piecewise linear representation and equal width temporal intervals followed by the results of the significance testing using Monte Carlo simulations.

5.3.1 Knowledge discovery in temporal intervals

Temporal intervals were found for both the SST data and the precipitation data.

Validation The algorithm was able to delineate unequal width intervals of stable and unstable periods of SST across the sensor array. The intervals become much more frequent in March of 2006. This would signify an area of interest in the spatiotemporal dataset to the climatologist and indicates a shift in the data. This shift is verified in Figure 11.

By calculating the SSE for each interval the location where the most change occurs can be identified. The interval with the second highest SSE value is represented in Figure 11 a). This interval, which occurred in March of 2006, coincides with a shift in the Oceanic Niño Index (ONI), an index used to classify El Niño and La Niña periods, from positive to negative according to the NOAA National Weather Service El Niño Cold and Warm Episodes by Season website [45]. This result shows that the algorithm was successful in identifying the 2006 El Niño event [43].

Validation Once again, the algorithm was effective in delineating unequal width intervals in the precipitation data. The interval divisions occur where there are large increases or decreases in precipitation across the area. The intervals become more segmented during the summer months. This is due to increased precipitation from summer thunderstorms.

Meaningful intervals that identify interesting events in the dataset were also found. The interval with the largest SSE is shown in Figure 12 a). In this case, the temporal interval approach identified the onset of a significant rain storm that lasted for five days and dropped almost 6 inches (15 cm) of rain on the Baltimore region [53].

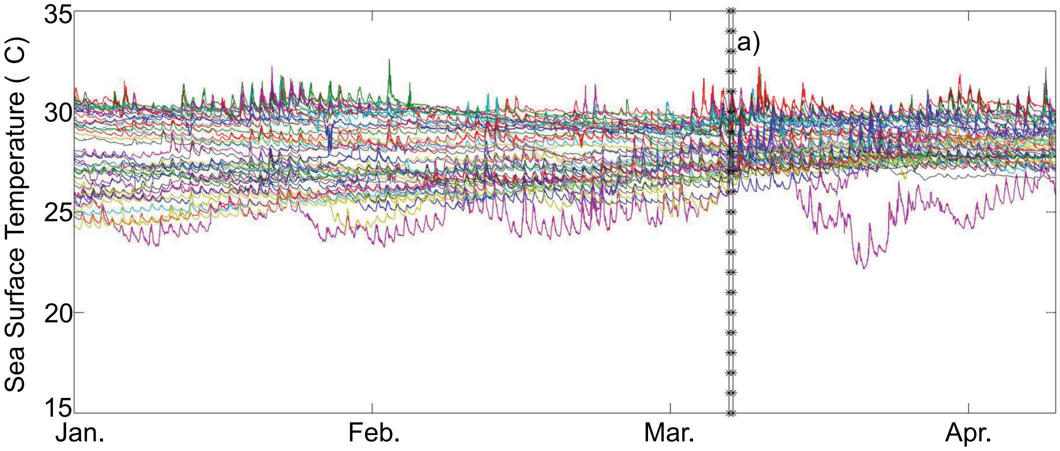


Figure 11: Temporal interval with greatest SSE for SST data where a) shows the location of the interval division.

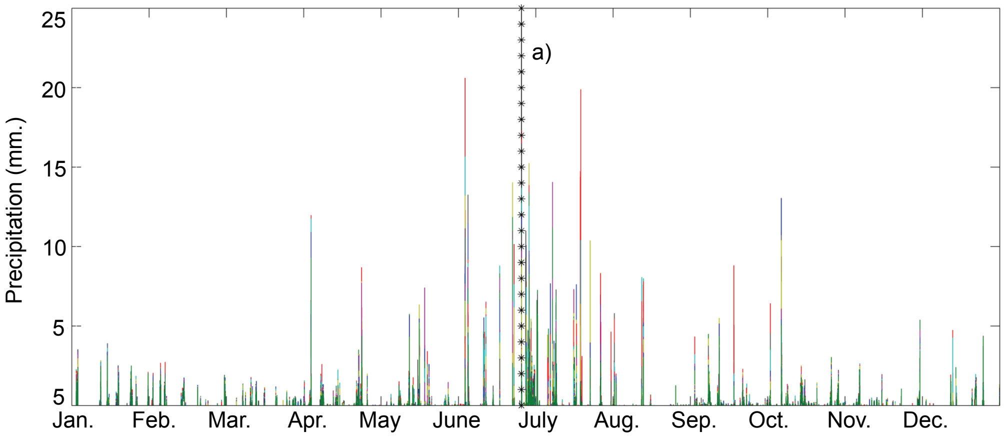


Figure 12: Temporal Interval with greatest SSE for precipitation data where a) shows the location of the interval division.

5.3.2 Comparison of temporal intervals to other approaches

The quality of the temporal intervals was compared using the between interval dissimilarity measure with equal width temporal intervals and piecewise linear representation. Time series data is often segmented using equally sized bins. We compare our method with an equal width binning of the time series to prove the need for unequal-width intervals to discretize spatiotemporal data. Equal width temporal intervals were created by dividing the time series into equally sized bins. We also compare our approach with a time series segmentation method that results in unequally sized bins. Piecewise linear approximation is a commonly used method for representing a complex time series in a given number of segments where the time series is approximated using a given number of linear segments. The resulting segments are then used to represent a high-level discretization of a time series that can be used for the purposes of data mining. In this experiment the bottom up segmentation algorithm found in [28] was used. The piecewise linear approximation algorithm was applied to the mean of the time series for both datasets. Since the temporal intervals are based on *md* between two spatial nodes, the different methods to calculate *md*, shown in Figure 8, were also compared. One fundamental difference between the agglomerative method and the equal width and piecewise linear representation methods is that the agglomerative method requires a base interval size while the others require the number of intervals. Because of this, the equal width and piecewise linear approaches were supplied with the number of intervals found by the temporal interval approach.

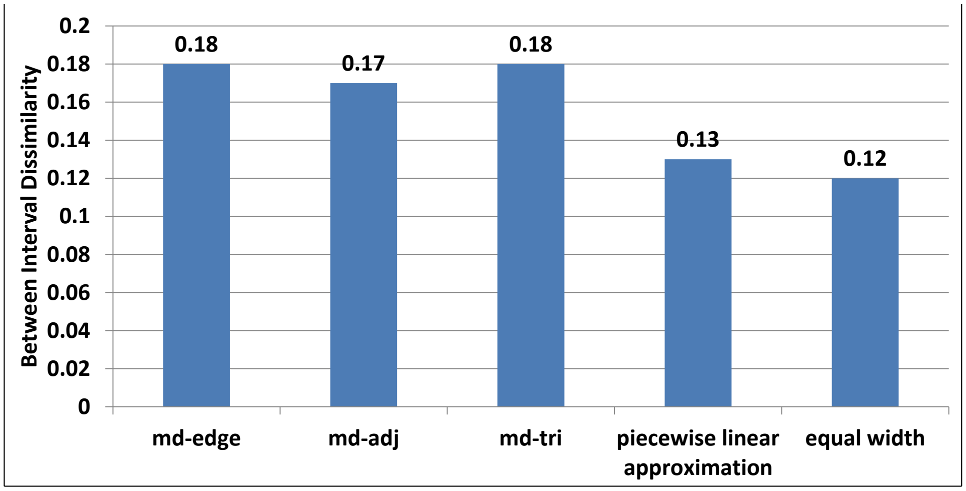


Figure 13: Comparison of *bid* for SST data.

The comparison results for the SST data are shown in Figure 13. The intervals created *md-tri* distance calculation slightly outperformed the *md-edge* and *md-adj* methods. Regardless of the way that the *md* is calculated, the intervals created by STN outperformed the intervals generated by piecewise linear approximation and equal-width methods. However, it must be noted that the goal of piecewise linear approximation is to approximate a time series with a given number of segments and that because of the variance present in the SST data, it was not an ideal approach to use to create temporal intervals.

The comparison results for the precipitation data are shown in Figure 14. The intervals created using the *md-adj* distance calculation slightly outperformed the *md-edge* method. The piecewise linear approximation outperformed STN for the precipitation data. This was because presence of precipitation either exists or does not exist and therefore, the mean across all sensors was adequate for generating temporal intervals using piecewise linear approximation. Finally, the equal width intervals had the lowest *bid* value.

5.3.3 Significance testing of temporal intervals

The significance of the temporal intervals was tested using the Monte Carlo simulation method shown in Section 4.6. Monte Carlo simulations were run with 10,000 iterations where the temporal measurements were kept the same for each spatial node and the in-

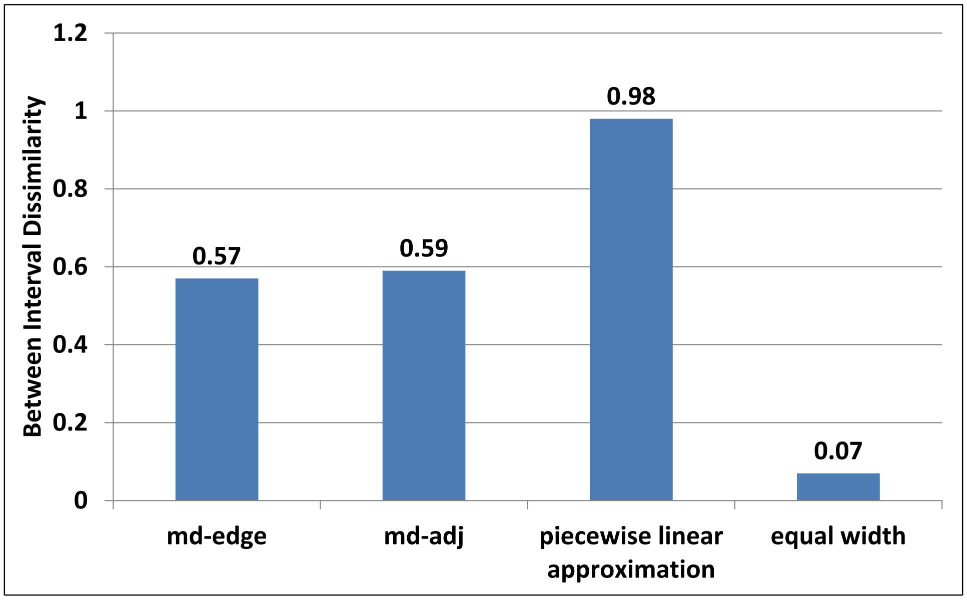


Figure 14: Comparison of bid for precipitation data.

terval divisions were placed randomly and the bid was calculated for each iteration. The Monte Carlo simulations were run for both the SST data and precipitation data. The significance was tested for the *md-edge*, *md-adj*, *md-tri*, piecewise linear representation, and equal width approaches. The results for the significance testing are shown in Table 2.

	SST Data	Precipitation Data
md-edge	0.001	0.001
md-tri	0.001	N/A
md-adj	0.001	0.001
Piecewise Linear Approximation	0.06	0.001
Equal Width	0.8	0.02

Table 2: Significance testing for temporal intervals.

The temporal intervals for all *md* types were significant beyond the 99% confidence interval with p-values of 0.001. The piecewise linear approximation approach was not significant for the SST data with a p-value of 0.06 and was significant for the precipitation data with a p-value of 0.001. The equal-width intervals were not significant for the SST data with a p-value of 0.8 and were significant for the precipitation data beyond the 95% confidence interval with a p-value of 0.02. Because of this, our approach outperformed the alternative methods in that we were able to find significant intervals for both datasets.

5.4 Spatiotemporal neighborhoods

In this section the results for spatiotemporal neighborhood discovery are presented for the SST and precipitation datasets. The empirical results for the experiments are presented. Then the STN approach is compared to a method that uses a fully connected graph [36]. Finally the significance of the STN approach is tested across the set of temporal intervals.

5.4.1 Knowledge discovery in spatiotemporal neighborhoods

For each dataset an example of the spatiotemporal neighborhoods were mapped for two adjacent temporal intervals. Then the connectivity of the spatiotemporal neighborhoods was assessed for a given time period. For this purpose the number of connections was counted for each spatial node across an entire year of SST data. Then k-means clustering was used to identify three clusters. In this classification stable nodes were highly connected across all intervals; boundary nodes had a significant number of intervals where they are weakly connected; and unstable nodes were weakly connected or completely disconnected across a large number of intervals.

Spatiotemporal neighborhoods for the SST data are shown in Figure 15. In this figure groups of connected spatial nodes are labeled with the same number. Weakly connected nodes are shown with squares and disconnected nodes are shown with circles. The spatial nodes are displayed with a satellite of SST as the background. The time series for each spatial node along with the interval division are shown below the depiction of the spatial nodes.

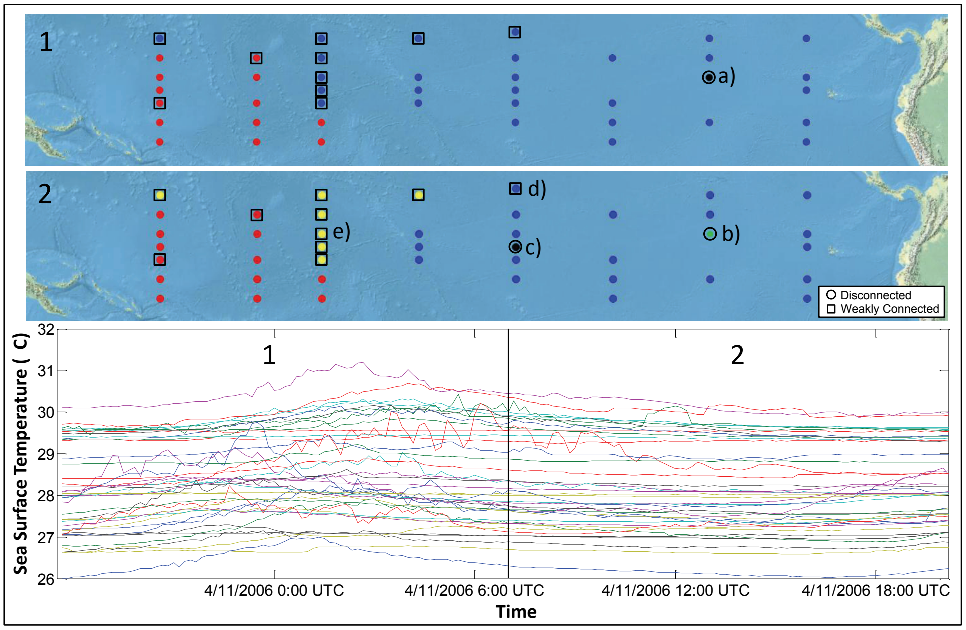


Figure 15: Spatiotemporal neighborhoods for two consecutive temporal intervals of SST data. The node shown in a) and b) is disconnected for both intervals. The node shown in c) becomes completely disconnected in interval 2. A moving trend is depicted in d) and e).

Validation The algorithm was able to find the pattern of SST as validated by the satellite imagery. Figure 15 a) depicts a completely disconnected spatial node. This is due to cold water that typically travels west along the equator. This spatial node is disconnected in 15 b). The cold water continues to travel westward in 15 c), where a node that was previously weakly connected in interval 1 is disconnected in interval 2. Figure 15 d) indicates another location where a moving trend is detected. In this scenario, a new neighborhood is created in Figure 15 e) and spreads north and eastward.

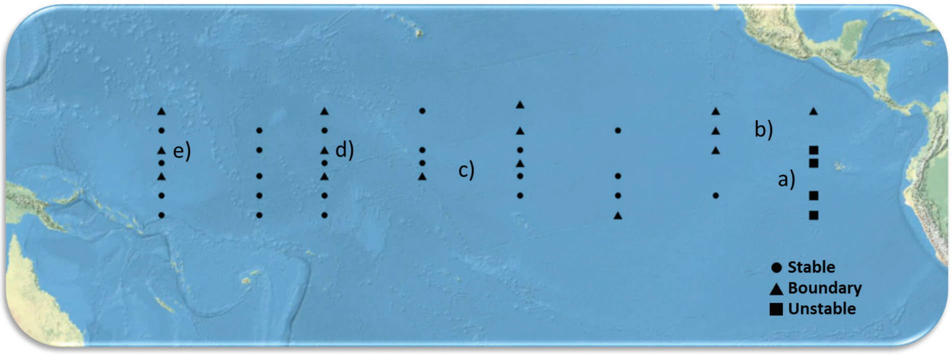
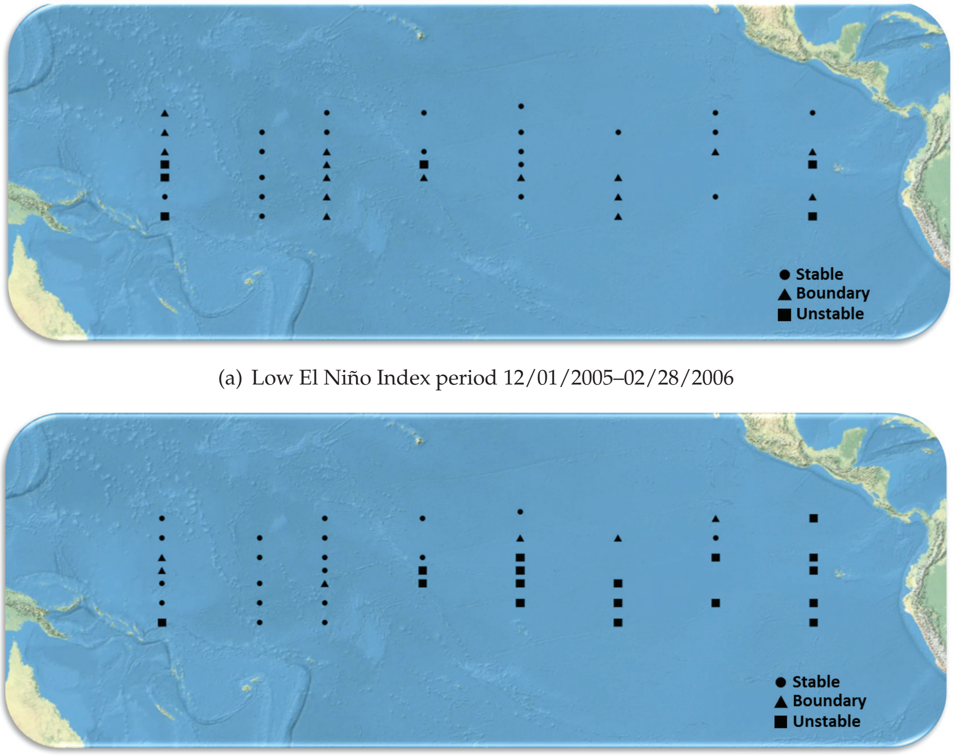


Figure 16: Pattern of stability for one year of SST data. a) and e) disconnected nodes; b) boundary nodes; c) and d) instability caused by cold water traveling west along the equator; e) instability caused by warm water coming from the southwest.

The results of the connectivity analysis are shown in 16. It is evident that in 16 a) there are four nodes that are consistently disconnected and in 16 b) there are four nodes that are consistently boundary nodes. The two boundary nodes shown in 16 c) and d) are caused by the cold water traveling west along the equator. Finally, the boundary nodes shown in 16 e) reflect warm water coming from the south west.

The connectivity strength pattern shown in Figure 16 naturally leads to the question of whether there is a different pattern of connectivity strength in high and low El Niño Index periods. To explore this further, two time periods were selected where the El Niño Index was low (12/01/2005–02/28/2006) and where the El Niño Index was high (10/01/2002–12/31/2002). Figure 5.4.1 shows the connectivity strength for each period. The low El Niño period is shown in 5.4.1 a) and the high El Niño period is shown in 5.4.1 b). Here we see a striking difference between the connectivity strength for each time period where for the low El Niño period the sensors are generally more connected in that there are less unstable nodes and in the high El Niño period there are a large number of unstable nodes that are clustered in the southeast quadrant of the sensor network where there exists a more variable spatial pattern. The sensors in the western Pacific Ocean tend to be much more stable. This is largely because during an El Niño event, warm water comes from the southwest Pacific Ocean.

Spatiotemporal neighborhoods were also discovered for the precipitation data. Figure 18 shows the spatiotemporal neighborhoods for two consecutive temporal intervals. The time series for each spatial node along with the interval division is shown in the bottom of the figure.



(a) Low El Niño Index period 12/01/2005–02/28/2006

(b) High El Niño Index period 10/01/2002–12/31/2002

Figure 17: Connectivity strength for high and low El Niño periods.

Validation The algorithm was effective in finding spatiotemporal neighborhoods in the precipitation data. The interval shown in Figure 18 characterizes a heavy precipitation event while in Figure 18-2 there is light precipitation. Figure 18 a) shows a completely disconnected node. Figure 18 b) shows a large neighborhood of strongly connected node. Figure 18 c) shows three areas where there are weakly connected nodes. The pattern shown here can be used to characterize precipitation for this particular temporal interval. Figures 18 a) and c) represent areas of highly variable precipitation where there exists a large gradient suggesting a locally heavy downpour. Figure 18 b) represents an area with low variability which indicates a homogeneous region of precipitation. In interval 2 in Figure 18, d) and f) show areas with strong connectivity while Figure 18 e) and g) show areas with weak connectivity. For this particular interval, these disconnected areas represent locally heavy precipitation cells caused by local thunderstorms. This result can be used to characterize this precipitation event represented by interval 1: the upper part of the watershed has a highly variable pattern of precipitation and the lower part of the watershed is largely homogeneous.

Figure 19 shows the results of the connectivity analysis for the entire year of precipitation data. It is evident that there is some spatial instability caused by the spatial configuration of the dataset in terms of unstable nodes occurring along the outside boundary. A

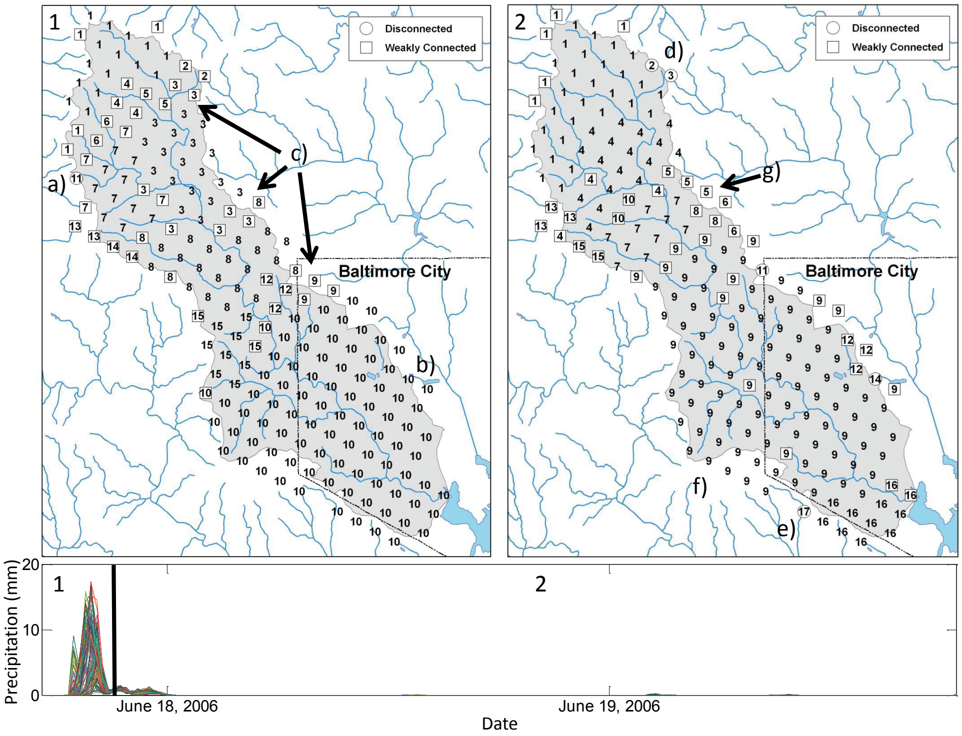


Figure 18: Spatiotemporal neighborhoods for two consecutive temporal intervals of precipitation data. a) a completely disconnected node. b) and f) strongly connected nodes; c), d), and g) areas with weakly connected nodes. The thunder storm characterized by these intervals produced highly variable precipitation in the upper portion of the watershed and largely homogeneous precipitation in the lower part of the watershed.

pattern of stability can be seen within the interior nodes of the dataset. Figure 19 a) and c) show two unstable interior nodes suggesting that the precipitation is highly variable over this time period for these areas in the watershed. Figure 19 b) and d) show lines of boundary nodes. The pattern of the boundary nodes seems to follow the streams shown on the map. This shows a possible physiographic influence on precipitation. From a knowledge discovery standpoint, this analysis could generate a hypothesis for the meteorologist to test. Furthermore, given multiple years of NEXRAD data, the connectivity strength could be used to compare the spatiotemporal pattern of precipitation between years.

5.4.2 Comparison of spatiotemporal neighborhoods to other approaches

This section presents a comparison of the methods to calculate *md*, a graph-based spatial neighborhood approach which uses a fully connected graph [36] and DBSCAN [11]. For the DBSCAN algorithm, we ran DBSCAN clustering as the neighborhood generation approach for each interval generated by the *md-edge* approach. The approaches were tested on both the SST data and precipitation data and the Moran's I statistic was calculated based on the resulting contiguity matrix for each approach across all intervals. The idea here is that the

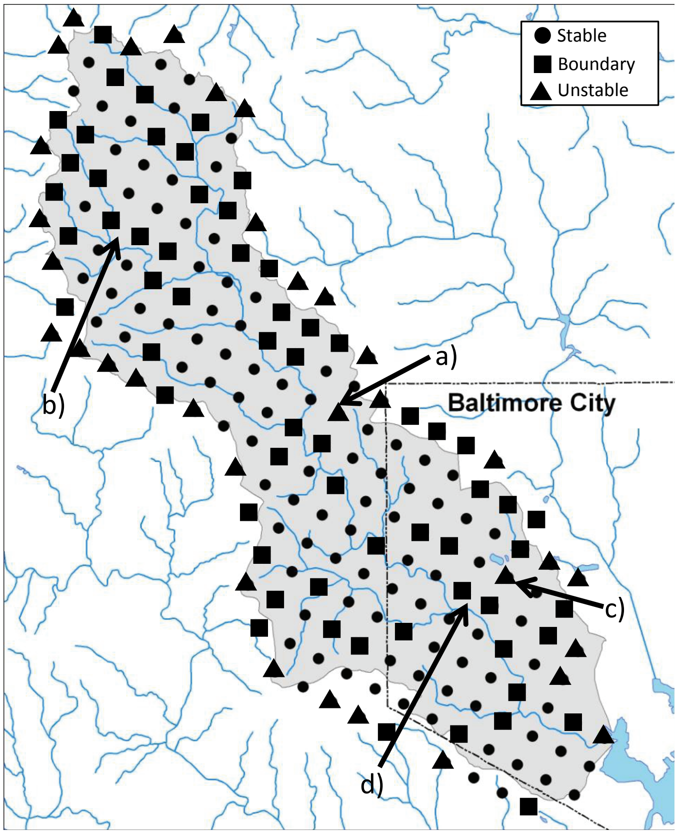


Figure 19: Pattern of stability for one year of precipitation data: a) and c) unstable nodes; b) and d) boundary nodes.

neighborhood configuration with the highest Moran's *I* value is the best representation of the spatial pattern for each temporal interval.

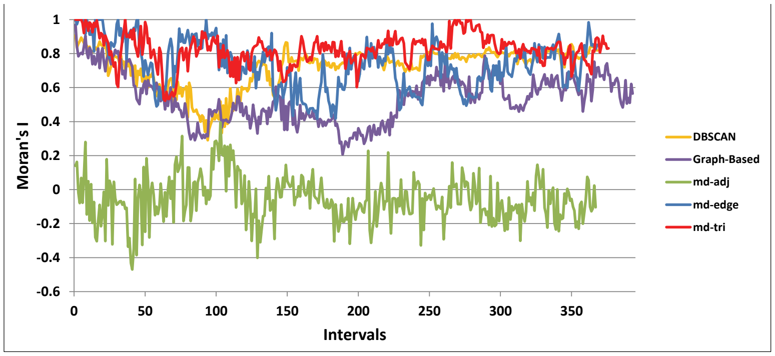


Figure 20: Comparison of Moran's *I* (SST data).

The results for the SST data are shown in Figure 20. Here the pattern of the Moran's I statistic is variable and generally decreases across the temporal intervals. Based on this result, the *md-tri* outperforms *md-edge*, *md-adj*, DBSCAN, and the graph-based neighborhoods, indicated by its ability to maximize the Moran's I value for a large number of intervals. Also, it must be noted that the contiguity matrix generated by *md-adj* was ineffective in estimating the pattern of spatial dependence indicated by low Moran's I values. Furthermore, DBSCAN, performs well in some intervals but generally does not outperform the *md-edge* and *md-tri* approaches.

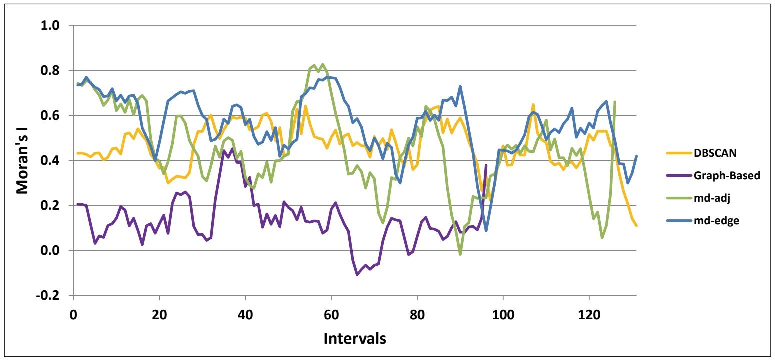


Figure 21: Comparison of Moran's I (precipitation data).

The results for the precipitation data are shown in Figure 21. Once again, the Moran's I statistic is highly variable across all approaches. The *md-edge* outperformed the *md-adj*, graph-based, and DBSCAN neighborhoods based on the fact that the correlation matrix generated by *md-edge* was able to achieve the highest Moran's I values. However, it must be noted that for one particular interval, the *md-edge* had a significantly lower Moran's I value. The contiguity matrix generated by the graph-based neighborhoods were much less effective in finding the pattern of spatial dependence based on the Moran's I statistic. The contiguity matrix generated by the DBSCAN algorithm, while not able to find high Moran's I values, generally had less variability in its ability to capture the pattern of spatial dependence.

5.5 Significance testing of spatiotemporal neighborhoods

The significance of the spatiotemporal neighborhood approach was tested by performing Monte Carlo simulations as described in Section 4.6. For this analysis the locations of the spatial nodes were kept constant and a random shuffle was performed on the temporal measurements for each time step. The *md-edge* approach was used to test the significance since it outperformed the other approaches. The results of the significance testing are shown in Figure 22.

In this example, for both the SST and precipitation data, the spatiotemporal neighborhoods were largely significant with p-values below the 95% confidence level. More specifically, for the SST data, 95% of the intervals were significant below the 99% confidence level and 99% of the intervals were significant below the 95% confidence level. For the precipitation data, 55% of the intervals were below the 99% confidence level and 73% were

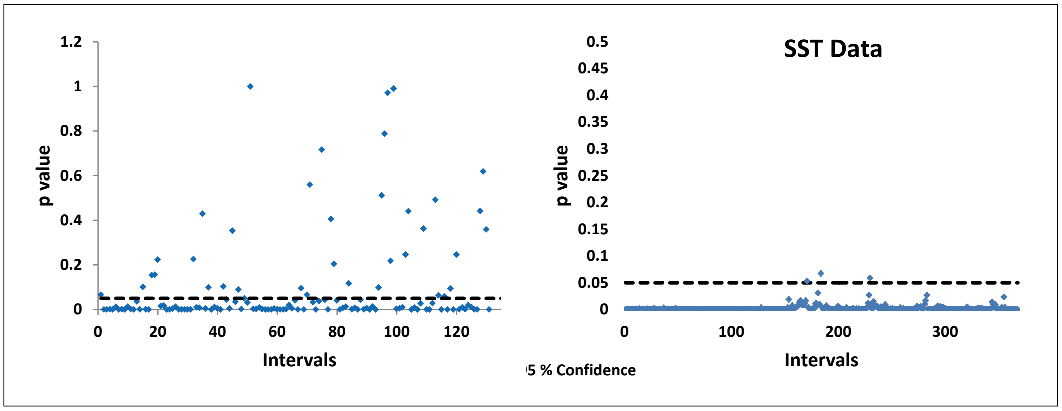


Figure 22: Significance of spatiotemporal neighborhoods.

below the 95% confidence level. Intervals were found to be insignificant where the resulting spatiotemporal neighborhoods have low spatial autocorrelation in general. The results of the significance testing are interesting in that not only are the spatiotemporal neighborhoods significant, the insignificant intervals suggest places in the data where further analysis should be focused.

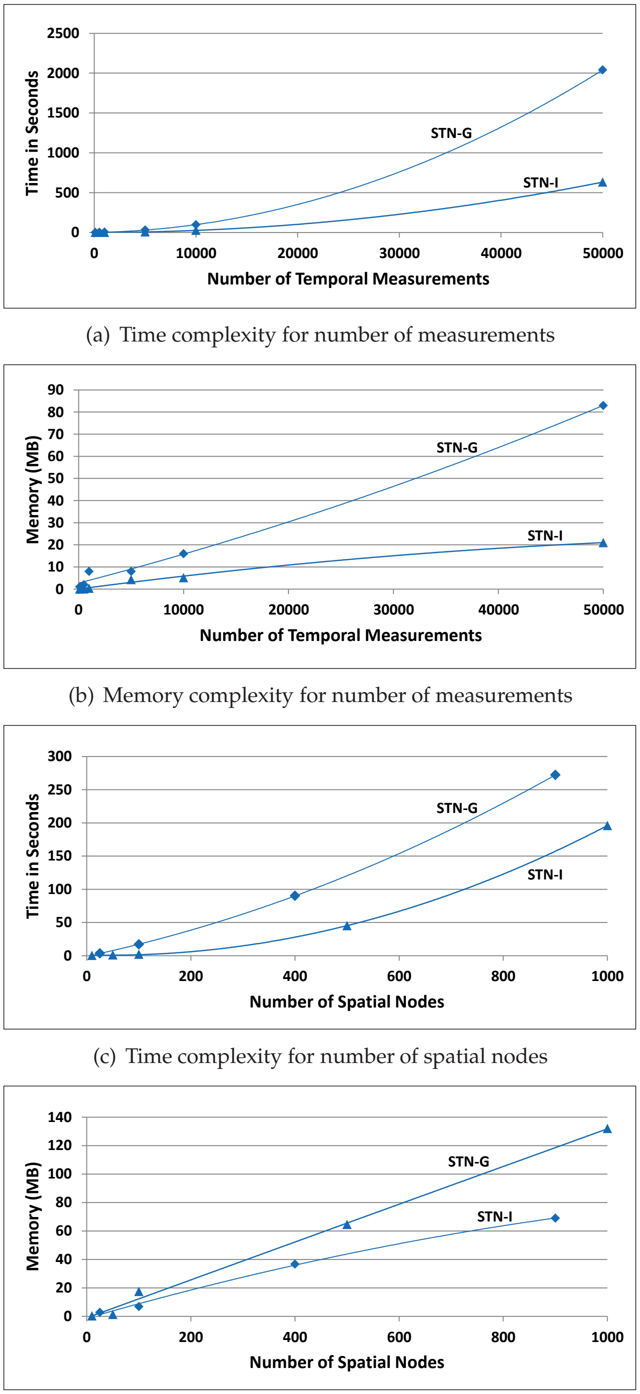
5.6 Time and memory complexity

Experiments were also performed to test the scalability of the approach in terms of time and memory efficiency. The experiments were performed on a workstation running the Microsoft Windows XP 64-bit operating system with two 2.8 GHz Intel Quad Core Processors and 8 GB of RAM. The algorithms were run in Matlab R2009a. Measurements were taken to record the execution time for the algorithm and the amount of memory required. Because the operating system has the ability to reuse memory, the average of 5 runs of the program was used. A simulated random dataset was used and the algorithm was run on a range of the number of spatial nodes as well as the number of measurements in the time series. The experiments included both STN-G for points distributed on a grid and STN-I for irregularly spaced points.

The time complexity results for the number of temporal measurements is shown in Figure 23 a). The experiment was run using 25 spatial nodes and 100, 500, 1000, 10,000, and 50,000 temporal measurements. The algorithm ran in polynomial time in both cases with STN-I slightly outperforming STN-G. This difference is caused by the triangulation adjacency being more efficient than 8 neighbor adjacency.

The effect of increasing the number of temporal measurements on memory usage was also tested. The results to this analysis are shown in Figure 23 b). In this case both algorithms are linear with the amount of memory increasing as the number of temporal measurements increases. However, STN-I uses less memory than STN-G as the number of temporal measurements becomes significantly larger.

The time complexity results for the number of spatial nodes is shown in Figure 23 c). For this experiment the number of temporal measurements were held constant at 1000. For STN-G 25, 100, 400, and 900 spatial nodes were used. For STN-I 10, 50, 100, 500, and 1000



(a) Time complexity for number of measurements

(b) Memory complexity for number of measurements

(c) Time complexity for number of spatial nodes

(d) Memory complexity for number of spatial nodes

Figure 23: Time and memory complexity of approach.

spatial nodes were used. The algorithm ran in polynomial time for this experiment with STN-I performing slightly better than STN-G for small numbers of spatial nodes. However, the time complexity converges at approximately 900 spatial nodes.

The memory complexity results are shown in Figure 23 d). For both algorithms the amount of memory used increases linearly with the number of spatial nodes. In particular, STN-G uses less memory as compared to STN-I. This is due to the amount of memory used to store the Delaunay triangulation and adjacency matrix for STN-I versus only the adjacency matrix for STN-G.

6 Discussion of results

The approach presented in this paper was able to find significant and meaningful unequal width temporal intervals and spatiotemporal neighborhoods. Our results lead to interesting observations about both the SST and precipitation data. For example the approach uncovered an interval with significant anomalies in the SST data, already identified by [44]. Furthermore, we were able to pinpoint an interval with significant precipitation. We were also able to find interesting spatiotemporal patterns using the connectivity strength, uncovering striking differences in the pattern of El Niño and La Niña conditions in the SST data. Connectivity strength was also effective in characterizing highly dynamic regions in the precipitation data. The method presented in this paper also performed well as compared to a number of other approaches. For the temporal intervals, our approach seems to be more suited to the SST data: the piecewise linear approximation of the precipitation data outperformed our method. This suggests that STN is more appropriate for measurements that have gradual fluctuations as opposed to measurements that fluctuate rapidly, such as the precipitation data. Significance testing further supports this suggestion. The piecewise linear approximation and equal width intervals were not significant, with p values above the 95% confidence threshold. For the spatiotemporal intervals the STN approach outperformed the other approaches in its ability to approximate the contiguity matrix with the highest Moran's I values. Again, the STN method was better suited for the SST data in that it identified more significant spatiotemporal intervals. It must be noted that the Moran's I values for the precipitation data were more variable than the SST data and there were more instances where the neighborhoods for the precipitation data were not significant. However, such variation exists across all approaches. In turn, this suggests that forming neighborhoods in the precipitation data for the analysis of spatial autocorrelation is a more challenging than in the SST data.

There exist a number of limitations to the approaches presented in this paper. First, the method that is used to create the temporal intervals is based on the amount of error present in a given set of base intervals. The base intervals are then merged according to their level of error. High error intervals are merged with other high error intervals and the same is the case for low error intervals. The approach is therefore well-suited to datasets that remain relatively constant within the spatiotemporal domain, such as environmental sensors and remotely sensed data. It is therefore unknown how the approach would perform in situations where there are large differences across the spatiotemporal domain. Furthermore, the approach does not capture periodicity such as daily, monthly, and seasonal fluctuations in a time series or large scale trends in the data.

Another limitation exists in that the algorithms require parameterization using a number of thresholds. The most important threshold is δ. In the spatiotemporal neighborhoods, lower values of δ result in a larger number of edges being cut from the graph and therefore a more disconnected neighborhood configuration. This is illustrated in Figure 24 a), which shows increasing values of δ based on the quantile of all *md* values for all intervals. Figure 24 a), a quantile of 0.5 represents the median of the *md* values; a higher quantile value represents a higher δ threshold; and a lower quantile value represents a lower δ threshold. Figure 24 a) shows that as the δ threshold increases, the number of neighborhoods decrease. At a quantile of 0.6, only one neighborhood is identified, an unacceptable result. Alternatively, when the δ threshold is at its lowest point, 31 neighborhoods are identified. This is also unacceptable because there will be a large number of sensors that form their own neighborhoods. An iterative approach is needed to choose the δ threshold, with the objective of ensuring that each neighborhood has a small number of sensors at the same time as adequately representing the spatial configuration of the data. The ideal δ threshold is usually dependent on the domain and the granularity required for the analysis.

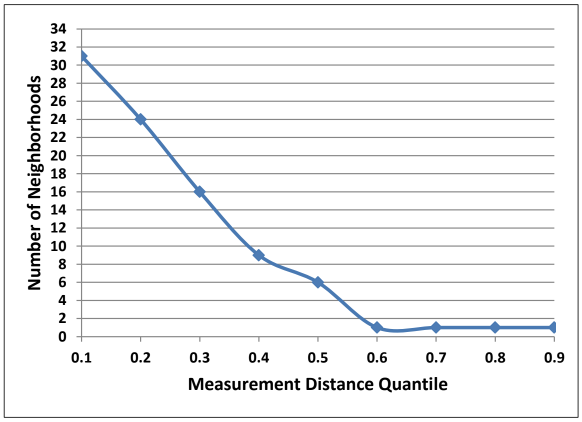


Figure 24: Effect of δ threshold on the average number of neighborhoods and average Moran's I value for the SST data.

7 Conclusion

This paper presents a novel method to identify spatiotemporal intervals and neighborhoods. The approach first discovers temporal intervals based on spatial relationships; and second discovers spatial neighborhoods within the temporal intervals. The approach is novel in that it is the first approach to combine spatial neighborhoods with temporal intervals to create temporal intervals and neighborhoods. The methods were validated using empirical ground truth evidence; measures such as Moran's I and between interval dissimilarity; as well as significance testing using Monte Carlo simulation. The methods were also compared with alternative methods. The results indicate that our spatial neighborhood approach outperforms a grid-based approach. The temporal intervals also outperformed all other methods where there is a high level of noise present in the data.

There are a number of areas where this work can be extended. The temporal intervals could be extended to include more properties of the time series, such as noise and periodicity and long term trends. A logical extension to this approach would be to include functionality to detect return frequencies and trends in the base intervals and to use this information to detect periodic and long-term trends in the data. For such an extension, a multi-resolution approach might be more suitable [37]. Another approach that has been recently proposed uses a combination of temporal and spatial autocorrelation to find spatiotemporal outliers [63]. This approach shows promising results in outlier and event detection. A similar approach using spatiotemporal autocorrelation could be used to find periodic and long-term trends in the data. Another direction for extension of this work would be to include a method to differentiate between interval divisions caused by malfunctioning sensors versus those caused by naturally occurring events. Lastly, the method presented in this paper is based on the analysis of a single attribute over space and time. This approach could be extended to deal with multiple attributes in a spatiotemporal dataset simply by adapting the *md* calculation, based on the Euclidean distance, to multiple attributes.

Acknowledgments

This work has been funded in part by the United States National Oceanic and Atmospheric Administration Grants NA06OAR4310243, NA07OAR4170518, and NA10OAR310220. The statements, findings, conclusions, and recommendations are those of the authors and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration or the Department of Commerce. This work was also partially supported by the FDRC grant of Towson University.

References

[1] ABONYI, J., FEIL, B., NÉMETH, S., AND ARVA, P. Fuzzy clustering based segmentation of time-series. In *Proc. 5th International Symposium on Intelligent Data Analysis (IDA)* (2003), M. R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt, Eds., Lecture Notes in Computer Science, Springer, pp. 275–285. doi:10.1007/978-3-540-45231-7_26.

[2] ADAM, N. R., JANEJA, V. P., AND ATLURI, V. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *Proc. ACM Symposium on Applied Computing (SAC)* (2004), pp. 576–583. doi:10.1145/967900.968020.

[3] ANKERST, M., BREUNIG, M., KRIEGEL, H., AND SANDER, J. OPTICS: Ordering points to identify the clustering structure. In *Proc. ACM SIGMOD International Conference on Management of Data* (1999), ACM Press, pp. 49–60. doi:10.1145/304182.304187.

[4] BELLMAN, R., AND ROTH, R. Curve fitting by segmented straight lines. *Journal of the American Statistical Association 64*, 327 (1969), 1079–1084. doi:10.1080/01621459.1969.10501038.

[5] CANE, M. Oceanographic events during El Niño. *Science 222*, 4629 (1983), 1189–1195. doi:10.1126/science.222.4629.1189.

[6] CHAN, J., BAILEY, J., AND LECKIE, C. Discovering and summarising regions of correlated spatio-temporal change in evolving graphs. In *Proc. 6th IEEE International Conference on Data Mining* (2006), IEEE Computer Society, pp. 361–365. doi:10.1109/ICDMW.2006.61.

[7] CLIFF, A. D., AND ORD, J. K. *Spatial Autocorrelation*. Pion, London, 1973. doi:10.2307/143144.

[8] CRESSIE, N., AND WIKLE, C. *Statistics for spatio-temporal data*. Wiley, Hoboken, NJ, 2011.

[9] DELAUNAY, B. Sur la sphere vide. *Bulletin of Academy of Sciences of the USSR*, 6 (1934), 793–800.

[10] ESTER, M., KRIEGEL, H., AND SANDER, J. Spatial data mining: A database approach. In *5th International Symposium on Advances in Spatial Databases* (1997), Springer, pp. 47–66. doi:10.1007/3-540-63238-7_24.

[11] ESTER, M., KRIEGEL, H., SANDER, J., AND XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining* (1996), pp. 226–231. doi:10.1.1.121.9220.

[12] FLAKE, G., TARJAN, R., AND TSIOUTSIOULIKLIS, K. Graph clustering and minimum cut trees. *Internet Mathematics 1*, 4 (2004), 385–408.

[13] FREY, B., AND DUECK, D. Clustering by passing messages between data points. *Science 315*, 5814 (2007), 972–976. doi:10.1126/science.1136800.

[14] GEARY, R. The contiguity ratio and statistical mapping. *The Incorporated Statistician* (1954), 115–145. doi:10.2307/2986645.

[15] GEORGE, B., KANG, J., AND SHEKHAR, S. Spatio-temporal sensor graphs (STSG): A sensor model for the discovery of spatio-temporal patterns. In *Proc. First International Workshop on Knowledge Discovery from Sensor Data* (2007). doi:10.3233/IDA-2009-0376.

[16] GEORGE, B., AND SHEKHAR, S. Time-aggregated graphs for modeling spatio-temporal networks. *Lecture Notes in Computer Science 4231* (2006), 85. doi:10.1007/11908883_12.

[17] GOODCHILD, M. F. *Spatial Autocorrelation*. Geo Books, Norwich, UK, 1986.

[18] GUHA, S., RASTOGI, R., AND SHIM, K. Cure: An efficient clustering algorithm for large databases. *Information Systems 26*, 1 (2001), 35–58. doi:10.1145/276304.276312.

[19] HADJIELEFTHERIOU, M., KOLLIOS, G., GUNOPULOS, D., AND TSOTRAS, V. On-line discovery of dense areas in spatio-temporal databases. *Lecture Notes in Computer Science* (2003), 306–324. doi:10.1.1.7.2606.

[20] HAN, J., M. K., AND TUNG, A. K. H. Spatial clustering methods in data mining: A survey. In *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, London, UK, 2001, ch. 1, pp. 1–29.

[21] HIMBERG, J., KORPIAHO, K., MANNILA, H., TIKANMAKI, J., AND TOIVONEN, H. Time series segmentation for context recognition in mobile devices. In *Proc. International Conference on Data Mining (ICDM)* (2001), pp. 203–210. doi:10.1109/ICDM.2001.989520.

[22] JANEJA, V., AND ATLURI, V. Random walks to identify anomalous free-form spatial scan windows. *IEEE Transactions on Knowledge and Data Engineering 20*, 10 (2008), 1378–1392.

[23] JOHNSON, S. Hierarchical clustering schemes. *Psychometrika 32*, 3 (1967), 241–254. doi:10.1007/BF02289588.

[24] KALNIS, P., MAMOULIS, N., AND BAKIRAS, S. On discovering moving clusters in spatio-temporal data. In *Proc. Advances in Spatial and Temporal Databases* (2005), C. Medeiros, M. Egenhofer, and E. Bertino, Eds., vol. 3633 of *Lecture Notes in Computer Science*, pp. 364–381.

[25] KANG, I.-S., KIM, T.-W., AND LI, K.-J. A spatial data mining method by Delaunay triangulation. In *Proc. 5th ACM International Workshop on Advances in Geographic Information Systems* (1997), ACM, pp. 35–39. doi:10.1145/267825.267836.

[26] KARYPIS, G., HAN, E.-H., AND KUMAR, V. Chameleon: Hierarchical clustering using dynamic modeling. *Computer 32*, 8 (1999), 68–75. doi:10.1109/2.781637.

[27] KAUFMAN, L., AND ROUSSEEUW, P. *Finding Groups in Data*. Wiley, New York, 1990.

[28] KEOGH, E., CHU, S., HART, D., AND PAZZANI, M. An online algorithm for segmenting time series. In *Proc. IEEE International Conference on Data Mining* (2001), IEEE Computer Society, pp. 289–296. doi:10.1109/ICDM.2001.989531.

[29] KEOGH, E., AND SMYTH, P. A probabilistic approach to fast pattern matching in time series databases. In *Proc. 3rd International Conference on Knowledge Discovery and Data Mining (KDD)* (1997), pp. 24–30.

[30] KOHLER, E., LANGKAU, K., AND SKUTELLA, M. Time-expanded graphs for flow-dependent transit times. In *Proc. 10th Annual European Symposium on Algorithms (ESA)* (2002), vol. 2, Springer, pp. 599–611.

[31] KRAJEWSKI, W. F., KRUGER, A., SMITH, J. A., BAECK, M. L., DOMASZCZYNSKI, P., GOSKA, R., SEO, B., CUNHA, L., GUNYON, C., VILLARINI, G., AND NTELEKOS, A. Hydro-NEXRAD: A community resource for future research on improving rainfall-rainfall estimation and hydrologic applications. In *AGU Fall Meeting Abstracts* (Dec. 2007).

[32] LEMIRE, D. A better alternative to piecewise linear time series segmentation. In *Proc. 7th SIAM International Conference on Data Mining* (2007).

[33] LIN, F., XIE, K., SONG, G., AND WU, T. A novel spatio-temporal clustering approach by process similarity. In *Proc. 6th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (2009), vol. 5, pp. 150–154. doi:10.1109/FSKD.2009.584.

[34] LIU, D., NOSOVSKIY, G. V., AND SOURINA, O. Effective clustering and boundary detection algorithm based on delaunay triangulation. *Pattern Recognition Letters 29*, 9 (2008), 1261–1273. doi:10.1016/j.patrec.2008.01.028.

[35] MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability* (1967), vol. 1, pp. 281–297.

[36] MCGUIRE, M., JANEJA, V., AND GANGOPADHYAY, A. Spatiotemporal neighborhood discovery for sensor data. *Lecture Notes in Computer Science 5840* (2010), 203–225. doi:10.1007/978-3-642-12519-5_12.

[37] MCGUIRE, M., JANEJA, V., AND GANGOPADHYAY, A. Characterizing sensor datasets with multi-granular spatio-temporal intervals. In *Proc. 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2011), ACM. doi:10.1145/2F2093973.2094045.

[38] MCPHADEN, M. Genesis and evolution of the 1997–98 el niño. *Science 283* (1999), 950–954. doi:10.1126/science.283.5404.950.

[39] METROPOLIS, N., AND ULAM, S. The Monte Carlo method. *Journal of the American Statistical Association* (1949), 335–341. doi:10.1080/01621459.1949.10483310.

[40] MOHAMMADI, S., JANEJA, V., AND GANGOPADHYAY, A. Discretized spatio-temporal scan window. In *Proc. 9th SIAM International Conference on Data Mining* (2009).

[41] MORAN, P. The interpretation of statistical maps. *Journal of the Royal Statistical Society B 10*, 2 (1948), 245–251.

[42] NG, R., AND HAN, J. CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering 14*, 5 (2002), 1003–1016. doi:10.1109/TKDE.2002.1033770.

[43] NOAA, NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. State of the Climate: El Niño/Southern Oscillation Analysis for Annual 2006. http://www.ncdc.noaa.gov/sotc/enso/2006/13, December 2006. Accessed 11 February 2013.

[44] NOAA, NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. State of the Climate El Niño/Southern Oscillation Analysis November 2006. http://www.ncdc.noaa.gov/sotc/?report=enso&year=2006&month=11&submitted=Get+Report, 2006. Accessed 11 February 2013.

[45] NOAA, NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. Cold and warm episodes by season. http://www.cpc.noaa.gov/products/analysis_monitoring/ensostuff/ensoyears.shtml, 2009. Accessed 11 February 2013.

[46] NOAA, NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. Tropical atmosphere ocean project. http://www.pmel.noaa.gov/tao/jsdisplay/, 2009. Accessed 11 February 2013.

[47] NOAA, NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION. El Niño Theme Page. http://www.pmel.noaa.gov/tao/elnino/el-nino-story.html, 2010. Accessed 11 February 2013.

[48] OKABE, A., BOOTS, B., SUGIHARA, K., AND CHIU., S. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000.

[49] RASMUSSON, E., AND WALLACE, J. Meteorological aspects of the El Niño/Southern Oscillation. *Science 222*, 4629 (1983), 1195–1202. doi:10.1126/science.222.4629.1195.

[50] RELJIN, I., RELJIN D, B., AND JOVANOVIĆ, G. Clustering and mapping spatial-temporal datasets using som neural networks. *Journal of Automatic Control 13*, 1 (2003), 55–60.

[51] REYNOLDS, R., SMITH, T., LIU, C., CHELTON, D., CASEY, K., AND SCHLAX, M. Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate 20*, 22 (2007), 5473–5496. doi:10.1175/2007JCLI1824.1.

[52] ROSSWOG, J. GHOSE, K. Detecting and tracking spatio-temporal clusters with adaptive history filtering. In *Proc. IEEE International Conference on Data Mining Workshops (ICDMW)* (2008), pp. 448–457. doi:10.1109/ICDMW.2008.93.

[53] ROYLANCE, F. Maryland weather blog. http://weblogs.marylandweather.com, 2006. Accessed 11 February 2013.

[54] SAP, M.N.M. AWAN, A. Finding spatio-temporal patterns in climate data using clustering. In *Proc. International Conference on Cyberworlds* (2005). doi:10.1109/CW.2005.45.

[55] SHEIKHOLESLAMI, G., CHATTERJEE, S., AND ZHANG, A. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. 24rd International Conference on Very Large Data Bases* (1998), pp. 428–439.

[56] SHEKHAR, S., EVANS, M., KANG, J., AND MOHAN, P. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1*, 3 (2011), 193–214. doi:10.1002/widm.25.

[57] SHEKHAR, S., LU, C., AND ZHANG, P. Detecting graph-based spatial outliers: Algorithms and applications (a summary of results). In *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2001), pp. 371–376. doi:10.1145/502512.502567.

[58] TOBLER, W. A computer movie simulating urban growth in the Detroit region. *Economic Geography 46* (1970), 234–240.

[59] VERHEIN, F., AND CHAWLA, S. Mining spatio-temporal patterns in object mobility databases. *Data Mining and Knowledge Discovery 16*, 1 (2008), 5–38. doi:10.1007/s10618-007-0079-5.

[60] WANG, W., YANG, J., AND MUNTZ, R. *STING: A Statistical Information Grid Approach to Spatial Data Mining*. Morgan Kaufmann, San Francisco, CA, 1997.

[61] ZABIH, R., AND KOLMOGOROV, V. Spatially coherent clustering using graph cuts. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Los Alamitos, CA, 2004), vol. 2, IEEE Computer Society, pp. 437–444.

[62] ZHANG, T., RAMAKRISHNAN, R., AND LIVNY, M. BIRCH: An efficient data clustering method for very large databases. In *Proc. ACM SIGMOD International Conference on Management of Data* (1996), ACM Press New York, NY, USA, pp. 103–114. doi:10.1.1.152.7115.

[63] ZHANG, Y., HAMM, N., MERATNIA, N., STEIN, A., VAN DE VOORT, M., AND HAVINGA, P. Statistics-based outlier detection for wireless sensor networks. *International Journal of Geographical Information Science 26*, 8 (2012), 1373–1392. doi:10.1080/13658816.2012.654493.