Abstract

Increasingly, the size, variety, and update rate of spatial datasets exceed the capacity of commonly used spatial computing and spatial database technologies to learn, manage, and process data with reasonable effort. We believe that this data, which we call Spatial Big Data (SBD), represents the next frontier in spatial computing. Examples of emerging SBD include temporally detailed roadmaps that provide traffic speed values every minute for every road in a city, GPS trajectory data from cell-phones, and engine measurements of fuel consumption, greenhouse gas emissions, etc. A 2011 McKinsey Global Institute report defines traditional big data as data featuring one or more of the 3 "V's": Volume, Velocity, and Variety. This chapter discusses Spatial Big Data through case-studies on real-world datasets that feature one or more of the 3 "V's": a study on change detection in climate data illustrates volume, a study on finding anomalies in real-time highway traffic sensors shows Velocity, and two studies on Variety demonstrate both variety in input and output. Spatial data has traditionally challenged traditional data querying and mining algorithms, requiring new and interesting algorithms to be developed. Spatial Big Data highlights these challenges and provides for a rich area of research.

Spatial Big Data: Case Studies on Volume, Velocity, and Variety

Michael R. Evans
Dev Oliver
Xun Zhou
Shashi Shekhar

Department of Computer Science
University of Minnesota
Minneapolis, MN

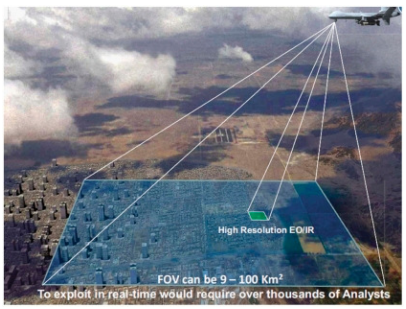
1 Introduction

Spatial computing encompasses the ideas, solutions, tools, technologies, and systems that transform our lives and society by creating a new understanding of spaces, their locations, places, as well as properties; how we know, communicate, and visualize our relation to places in a space of interest; and how we navigate through those places. From virtual globes to consumer global navigation satellite system devices, spatial computing is transforming society. With the rise of new Spatial Big Data, spatial computing researchers will be working to develop a compelling array of new geo-related capabilities. We believe that this data, which we call Spatial Big Data (SBD), represents the next frontier in spatial computing. Examples of emerging SBD include temporally detailed roadmaps that provide traffic speed values every minute for every road in a city, GPS trajectory data from cell-phones, and engine measurements of fuel consumption, greenhouse gas emissions, etc. A 2011 McKinsey Global Institute report defines traditional big data as data featuring one or more of the 3 "V's": Volume, Velocity, and Variety [1]. Spatial data frequently demonstrates at least one of these core features, given the variety of datatypes in spatial computing such as points, lines, and polygons. In addition, spatial analytics have shown to be more computationally expensive than their non-spatial brethren [2] as they need to account for spatial autocorrelation and non-stationarity, etc.

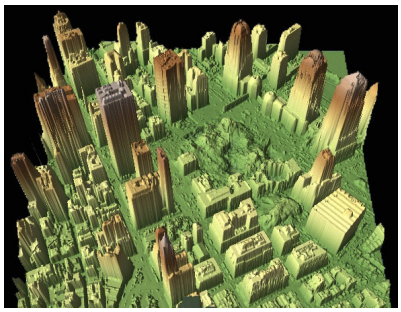
In this chapter we begin in Section 2 by defining spatial big data, enumerating three traditional categories of spatial data, and discussing their spatial big data equivalents. We then use case-studies to demonstrate the 3 "V's" of spatial big data: Volume, Velocity and Variety [1]. A case-study on climate data in Section 3 illustrates the challenges of utilizing large volumes of spatial big data. Velocity is demonstrated in Section 4 through a case-study on loop detector (traffic speed) data on the Twin Cities, MN highway network. Lastly, variety in spatial big data can refer to both the type of data input used and the variety in the type of output representations. We illustrate variety in data types through a case-study on GPS trajectory data to find cyclist commuter corridors in Minneapolis, MN, and variety in data output is demonstrated through network activity summarization of pedestrian fatality data from Orlando, FL.

2 What is Spatial Big Data?

Spatial data are discrete representations of continuous phenomena. Discretization of continuous space is necessitated by the nature of digital representation. There are three basic models to represent spatial data: raster (grid), vector and network. Satellite images are good examples of raster data. On the other hand, vector data consists of points, lines, polygons and their aggregate (or multi-) counterparts. Graphs consisting of spatial networks are another important data type used to represent road networks. We define Spatial Big Data as simply instances of these data types that exhibit at least one of the 3 "V's", Volume, Velocity and Variety. Below, we provide examples of Spatial Big Data in each of these



(a) Wide-area persistent surveillance. FOV: Field of view. (Photo courtesy of the Defense Advanced Research Projects Agency.) EO: Electro-optical. [4]



(b) LIDAR images of ground zero rendered Sept. 27, 2001 by the U.S. Army Joint Precision Strike Demonstration from data collected by NOAA flights. Thanks to NOAA/U.S. Army JPSD.

Figure 1: Spatial Big Data brings new challenges through data volume, velocity and variety.

core spatial data types: Raster, Vector, Network.

Raster data, such as geo-images (Google Earth), are frequently used for remote sensing and land classification. New Spatial Big Raster Datasets are emerging from a number of sources:

UAV Data: Wide area motion imagery sensors are increasingly being used for persistent surveillance of large areas, including densely populated urban areas. The wide-area video coverage and 24/7 persistent surveillance of these sensor systems allow for new and interesting patterns to be found via temporal aggregation of information. However, there are several challenges associated with using UAVs in gathering and managing raster datasets. First, UAV has a small footprint due to the relatively low flying height, therefore, it captures a large amount of images in a very short period of time to achieve the spatial coverage for many applications. This poses a significant challenge to store increasing large digital images. Image processing is another challenge because traditional approaches have shown to be too time consuming and costly to rectify and mosaic the UAV photography for large areas. The large quantity of data far exceeds the capacity of the available pool of human analysts [3]. It is essential to develop automated, efficient, and accurate technique to handle these spatial big data.

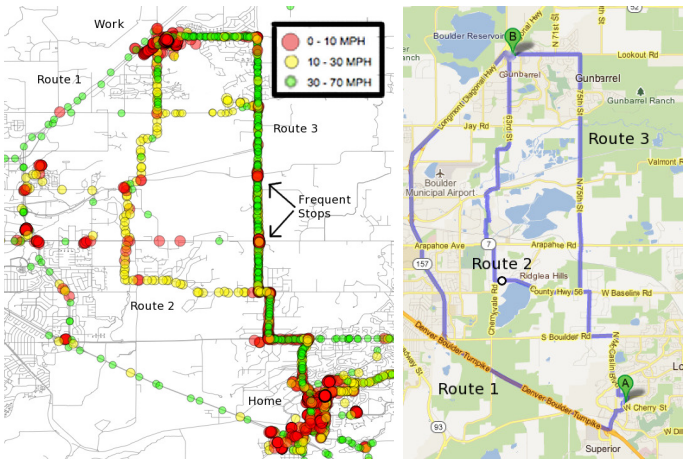
LiDAR: Lidar (Light Detection and Ranging or Laser Imaging Detection and Ranging) data is generated by timing laser pulses from an aerial position (plane or satellite) over a selected area to produce a surface mapping [5]. Lidar data are very rich for use cases related to surface analysis or feature extraction. However, these datasets are noisy and may contain irrelevant data for spatial analysis and sometimes miss critical information. These large volumes of data from multiple sources pose a big challenge on management, analysis, and timely accessibility. Particularly, Lidar points and their attributes have tremendous sizes making them difficult to categorize these datasets for end-users. Data integration from multiple spatial sources is another challenge due to the massive amounts of Lidar datasets. Therefore, Spatial Big Data is an essential issue for Lidar remote sensing.

Vector data models over space is a framework to formalize specific relationships among a set of objects. Vector data consists of points, lines and polygons; and with the rise of Spatial Big Data, corresponding datasets have arisen from a variety of sources:

VGI Data: Volunteered geographic information (VGI) brings a new notion of infrastructure to collect, synthesize, verify, and redistribute geographic data through geo-location technology, mobile devices, and geo-databases. These geographic data are provided, modified, and shared based on user interactive online services (e.g., OpenStreetMap, Wikimapia, GoogleMap, GoogleEarth, Microsofts Virtual Earth, Flickr, etc). In recent years, VGI leads an explosive growth in the availability of user-generated geographic information and requires scalable storage models to handle large-scale spatial datasets. The challenge for VGI is to enhance data service quality with regard to accuracy, credibility, reliability, and overall value [6].

GPS Trace Data: GPS trajectories are quickly becoming available for a larger collection of vehicles due to rapid proliferation of cell-phones, in-vehicle navigation devices, and other GPS data-logging devices [7] such as those distributed by insurance companies [8]. Such GPS traces allow indirect estimation of fuel

efficiency and greenhouse gas (GHG) emissions via estimation of vehicle-speed, idling and congestion. They also make it possible to provide personalized route suggestions to users to reduce fuel consumption and GHG emissions. For example, Figure 2 shows 3 months of GPS trace data from a commuter with each point representing a GPS record taken at 1 minute intervals, 24 hours a day, 7 days a week. As can be seen, 3 alternative commute routes were identified between home and work from this dataset. These routes may be compared for engine idling which are represented by darker (red) circles. Assuming the availability of a model to estimate fuel consumption from speed profiles, one may even rank alternative routes for fuel efficiency. In recent years, consumer GPS products [7, 9] are evaluating the potential of this approach. Again, a key hurdle is the dataset size, which can reach 10^{13} items per year given constant minute-resolution measurements for all 100 million US vehicles.



(a) GPS trace data. Color indicates speed.

(b) Routes 1, 2, & 3 [10].

Figure 2: A commuter's GPS tracks over three months reveal preferred routes. (Best viewed in color)

Network data is commonly used to represent road maps for routing queries. While the network structure of the graph may not change, the amount of information about the network is rising drastically. New temporally-detailed road maps give minute by minute speed information, along with elevation and engine measurements to allow for more sophisticated querying of road networks.

Spatio-Temporal Engine Measurement Data: Many modern fleet vehicles include rich instrumentation such as GPS receivers, sensors to periodically measure sub-system properties [11–16], and auxiliary computing, storage and communication devices to log and transfer accumulated datasets. Engine measurement datasets may be used to study the impacts of the environment (e.g., elevation changes, weather), vehicles (e.g., weight, engine size, energy-source), traffic management systems (e.g., traffic light timing policies), and driver behaviors (e.g., gentle acceleration or braking) on fuel savings and GHG emissions. These datasets may include a time-series of attributes such as vehicle location, fuel levels, vehicle speed, odometer values, engine speed in revolutions per minute (RPM), engine load, emissions of greenhouse gases (e.g., CO2 and NOX), etc. Fuel efficiency can be estimated from fuel levels and distance traveled as well as engine idling from engine RPM. These attributes may be compared with geographic contexts such as elevation changes and traffic signal patterns to improve understanding of fuel efficiency and GHG emission. For example, Figure 3 shows heavy truck fuel consumption as a function of elevation from a recent study at Oak Ridge National Laboratory [17]. Notice how fuel consumption changes drastically with elevation slope changes. Fleet owners have studied such datasets to fine-tune routes to reduce unnecessary idling [18, 19]. It is tantalizing to explore the potential of such datasets to help consumers gain similar fuel savings and GHG emission reduction. However, these datasets can grow big. For example, measurements of 10 engine variables, once a minute, over the 100 million US vehicles in existence [20, 21], may have 10^{14} data-items per year.

Historical Speed Profiles: Typically, digital road maps consist of center lines and topologies of road networks [22, 23]. These maps are used by navigation devices and web applications such as Google Maps [10] to suggest routes to users. New datasets from companies such as NAVTEQ [24], use probe vehicles and highway sensors (e.g., loop detectors) to compile travel time information across road segments throughout the day and week at fine temporal resolutions (seconds or minutes). This data is applied to a profile model, and patterns in the road speeds are identified throughout the day. The profiles have

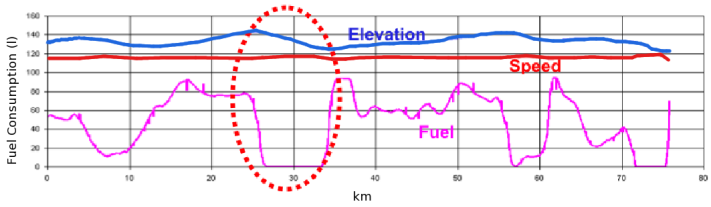
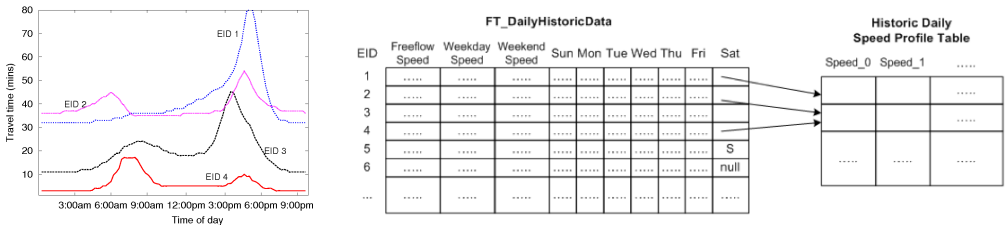


Figure 3: Engine measurement data improve understanding of fuel consumption [17]. (Best in color)

data for every five minutes, which can then be applied to the road segment, building up an accurate picture of speeds based on historical data. Such TD roadmaps contain much more speed information than traditional roadmaps. While traditional roadmaps have only one scalar value of speed for a given road segment (e.g., EID 1), TD roadmaps may potentially list speed/travel time for a road segment (e.g., EID 1) for thousands of time points, see Figure 4(a), in a typical week. This allows a commuter to compare alternate start-times in addition to alternative routes. It may even allow comparison of (start-time, route) combinations to select distinct preferred routes and distinct start-times. For example, route ranking may differ across rush hour and non-rush hour and in general across different start times. However, TD roadmaps are large and their size may exceed 10^{13} items per year for the 100 million road-segments in the US when associated with per-minute values for speed or travel-time. Thus, industry is using speed-profiles, a lossy compression based on the idea of a typical day of a week, as illustrated in Figure 4(b), where each (road-segment, day of the week) pair is associated with a time-series of speed values for each hour of the day.



(a) Travel time along four road segments over a day.

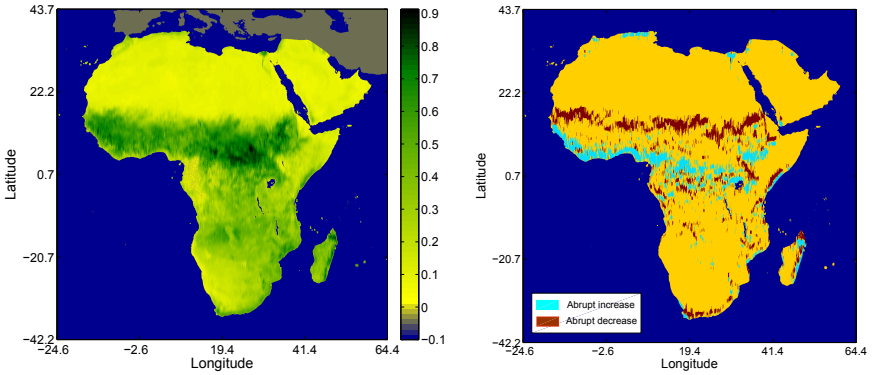
(b) Schema for Daily Historic Speed Data.

Figure 4: Spatial Big Data on Historical Speed Profiles. (Best viewed in color)

3 Volume: Discovering Sub-paths in Climate Data

Sub-paths (i.e., intervals) in spatio-temporal (ST) datasets can be defined as contiguous subsets of locations. Given a ST dataset and a path in its embedding ST framework, the goal of the interesting spatio-temporal sub-path discovery problem is to identify all the dominant (i.e., not a subset of any other) interesting sub-paths along the path defined by a given interest measure. The ability to discover interesting sub-paths is important to many societal applications. For example, coastal area authorities may be interested in intervals of coastal lines which are prone to rapid environmental change due to rising ocean levels and melting polar icecaps. Water quality monitors may be interested in river segments where water quality changes abruptly.

An extended example from eco-climate science illustrates the interesting sub-path discovery problem in details. This example comes from our collaboration with scientists studying the response of ecosystems to climate change by observing changes in vegetation cover across ecological zones. Sub-paths of abrupt vegetation cover change may serve to outline the spatial footprint of ecotones, the transitional areas between these zones [25]. Due to their vulnerability to climate changes, finding and tracking ecotones gives us important information about how the ecosystem responds to climate changes. Figure 5 illustrates the application of interesting sub-path discovery on the Africa vegetation cover in normalized difference vegetation index (NDVI) data, August, 1981. Figure 5(a) shows a map of vegetation cover in Africa [26]. Each longitudinal path is taken as an input of the problem. The output, as shown in Figure 5 (b), is a map of longitudinal sub-paths with abrupt vegetation cover changes, where red and blue represent sub-paths of abrupt vegetation cover decrease and increase northward respectively. As indicated by the



(a) Smoothed Africa vegetation dataset (measured in NDVI) in August, 1981.

(b) Longitudinal intervals of abrupt vegetation change in August, 1981

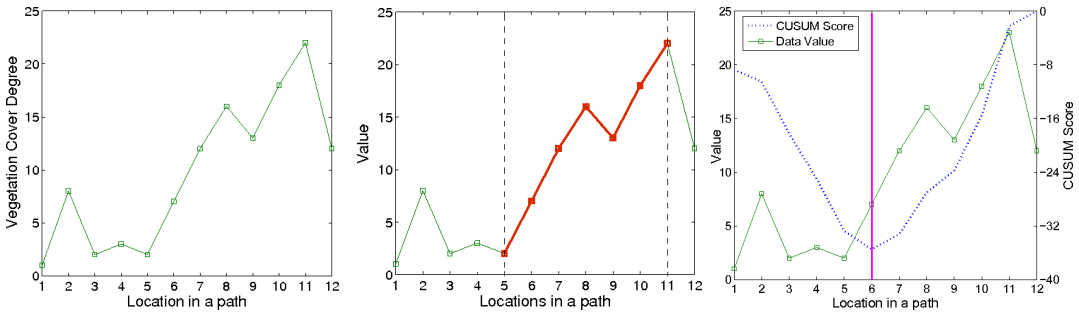
Figure 5: An application example of the interesting interval discovery problem. (Best viewed in color)

two colors as in Figure 5, footprints of several ecotones in Africa are discovered. One of them is the Sahel region (in the middle in red), where vegetation cover exhibits an abrupt decreasing trend from south to north.

Discovering interesting sub-paths is challenging due to the following reasons. First, the length of the sub-paths of interest may vary, without a pre-defined maximum length. For example, the length of flood-prone interval in long rivers (e.g., the Gange, Mississippi, etc.) may extend hundreds or thousands of miles. Second, the interestingness in a sub-path may not exhibit monotonicity, i.e., uninteresting intervals may be included in an interesting sub-path. Third, the data volume is potentially large. For example, consider the problem of finding all the interesting longitude sub-paths exhibiting abrupt change in an eco-climate dataset with attributes such as vegetation, temperature, precipitation, etc., over hundreds of years from different global climate models and sensor networks. The volume of such spatial big data ranges from terabytes to petabytes.

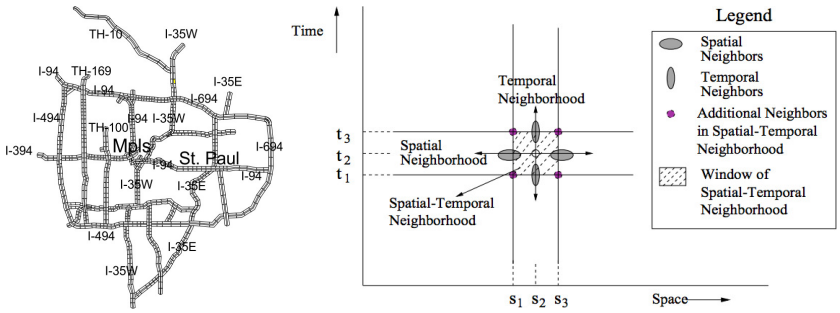
Previous work on interesting spatiotemporal sub-path/interval discovery focused on change point detection using one dimensional or two dimensional approaches. One dimensional approaches aim to find points in a time series where there is a shift in the data distribution [27–29]. Figure 6(a) shows a sample dataset in vegetation cover along a particular longitude. Figure 6(b) shows an sub-path in this dataset from location 5 to 11 whose data exhibits an abruptly increasing trend. In contrast, Figure 6(c) shows the output of a specific implementation of the popular statistical measure CUSUM [27, 30] on the same data where only location 6 is identified as a point of interest (with abrupt change from below the mean to above the mean). Two dimensional approaches such as edge detection [31] aim at finding boundaries between different areas in an image. However, the footprints of an identified edge over each one-dimensional path (e.g., row or column) are still points. The above related works are limited to detecting points of interest in a ST path, rather than finding long interesting sub-paths/intervals. In contrast, our novel computational frameworks discover sub-paths of arbitrary length based on certain interest measures.

In our preliminary work [32], a sub-path enumeration and pruning (SEP) approach was proposed. In the approach, the enumeration space of all the sub-paths is modeled as a grid-based directed acyclic graph (G-DAG), where nodes are sub-paths and edges are subset relationships between sub-paths. The approach enumerates all the sub-path by performing a breadth-first traversal on the G-DAG, starting from the root (longest sub-path). Each sub-path is evaluated by commuting its algebraic interest measure. Should an interesting sub-path be identified, all its subsets are pruned. By doing this, we significantly reduce the number of sub-path evaluations. We apply this approach on eco-climate datasets to find abrupt change sub-paths. An algebraic interest measure named "sameness degree" was designed to evaluate the change abruptness and persistence of a sub-path. As noted earlier, case study results on Normalized Difference Vegetation Index (NDVI) vegetation cover dataset showed that the approach can discover important patterns such as ecotones (e.g., the Sahel region). We also apply this approach on temporal paths (e.g., precipitation time series) and discovered patterns such as abrupt precipitation shifts in Africa. Experimental results on large synthetic datasets confirmed that the proposed approach is efficient and scalable.



(a) Vegetation cover along a longitudinal path (b) An interesting sub-path in the path (c) Result of CUSUM

Figure 6: A comparison of interesting sub-paths in the data and change point found by related work. (Best viewed in color)



(a) Traffic speed detector map. (b) Spatio-temporal outlier detection framework

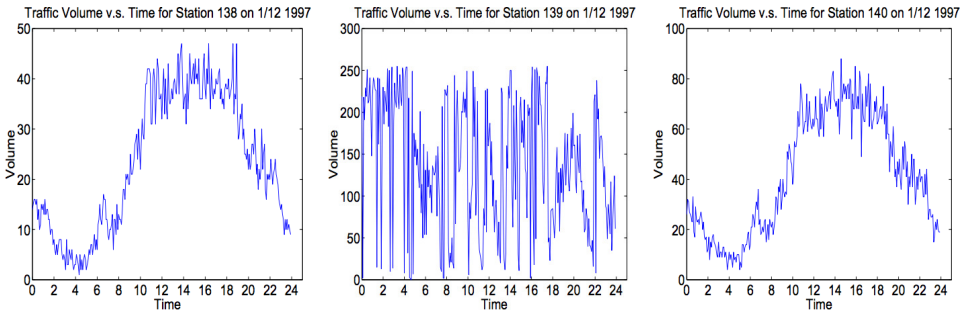
Figure 7: Detecting outliers in real-time traffic data.

4 Velocity: Spatial Graph Outlier Detection in Traffic Data

In this section, we demonstrate spatial big data featuring velocity via a case-study on real-time traffic monitoring datasets for detecting outliers in spatial graph datasets. We formalize the problem of spatio-temporal outlier detection and propose an efficient graph-based outlier detection algorithm. We use our algorithm to detect spatial and temporal outliers in a real-world Minneapolis-St. Paul dataset, and show effectiveness of our approach.

In 1997, the University of Minnesota and the Track Management Center Freeway Operations group started a joint project to archive sensor network measurements from the freeway system in the Twin Cities [33]. The sensor network includes about nine hundred stations, each of which contains one to four loop detectors, depending on the number of lanes. Sensors embedded in the freeways and interstate monitor the occupancy and volume of track on the road. At regular intervals, this information is sent to the track Management Center for operational purposes, e.g., ramp meter control, as well as research on track modeling and experiments. Figure 7(a) shows a map of the stations on the highways within the Twin-Cities metropolitan area, where each polygon represents one station. The interstate freeways include I-35W, I35E, I-94, I-394, I-494, and I-694. The state trunk highways include TH-100, TH-169, TH-212, TH-252, TH-5, TH-55, TH-62, TH-65, and TH-77. I-494 and I-694 together form a ring around the Twin-Cities. I-94 passes from East to North-West, while I-35W and I-35E run in a South-North direction. Downtown Minneapolis is located at the intersection of I-94, I-394, and I-35W, and downtown Saint Paul is located at the intersection of I-35E and I-94. For each station, there is one detector installed in each lane. The track flow information measured by each detector can then be aggregated to the station level. The system records all the volume and occupancy information within each 5-minute time slot at each particular station.

In this application, we are interested in discovering 1) the location of stations whose measurements are inconsistent with those of their graph-based spatial neighbors and 2) time periods when those



(a) Station 138 (b) Station 139 (c) Station 140

Figure 8: Outlier station 139 and its neighbor stations over one day.

abnormalities arise. We use three neighborhood definitions in this application as shown in Figure 7(b). First, we define a neighborhood based on the spatial graph connectivity as a spatial graph neighborhood. In Figure 7(b), (s1; t2) and (s3; t2) are the spatial neighbors of (s2; t2) if s1 and s3 are connected to s2 in a spatial graph. Second, we define a neighborhood based on a time series as a temporal neighborhood. In Figure 7(b), (s2; t1) and (s2; t3) are the temporal neighbors of (s2; t2) if t1, t2, and t3 are consecutive time slots. In addition, we define a neighborhood based on both space and time series as a spatial-temporal neighborhood. In Figure 7(b), (s1; t1), (s1; t2), (s1; t3), (s2; t1), (s2; t3), (s3; t1), (s3; t2), and (s3; t3) are the spatial-temporal neighbors of (s2; t2) if s1 and s3 are connected to s2 in a spatial graph, and t1, t2, and t3 are consecutive time slots.

The test for detecting an outlier can be described as follows: $|\frac{S(x)-\mu_s}{\sigma_s}| > \theta$ For each data object x with an attribute value $f(x)$, the $S(x)$ is the difference of the attribute value of data object x and the average attribute value of its neighbors. μ_s is the mean value of all $S(x)$, and σ_s is the standard deviation of all $S(x)$. Choice of θ depends on specified confidence interval. For example, a confidence interval of 95 percent will lead to $\theta \approx 2$.

In prior work [33], we proposed an I/O efficient algorithm to calculate the test parameters, e.g., mean and standard deviation for the statistics. The computed mean and standard deviation can then be used to validate the outlier of the incoming dataset. Given an attribute dataset V and the connectivity graph G, the TPC algorithm first retrieves the neighbor nodes from G for each data object x, then it computes the difference of the attribute value of x to the average of the attribute values of x's neighbor nodes. These different values are then stored as a set. Finally, that set is computed to get the distribution value μ_s and σ_s. Note that the data objects are processed on a page basis to reduce redundant I/O. In other words, all the nodes within the same disk page are processed before retrieving the nodes of the next disk page.

The neighborhood aggregate statistics value, e.g., mean and standard deviation, can be used to verify the outlier of an incoming dataset. The two verification procedures are Route Outlier Detection and Random Node Verification. The Route Outlier Detection procedure detects the spatial outliers from a user specified route. The Random Node Verification procedure check the outlierness from a set of randomly generated nodes. The step to detect outliers in both algorithms are similar, except that the Random Node Verification has no shared data access needs across test for different nodes. The storage of dataset should support I/O efficient computation of this operation.

Given a route RN in the dataset D with graph structure G, the Route Outlier Detection algorithm first retrieves the neighboring nodes form G for each data object x in the route RN, then it computes the difference $S(x)$ between the attribute value of x and the average of attribute values of x's neighboring nodes. Each $S(x)$ can then be tested using the spatial outlier detection test $|\frac{S(x)-\mu_s}{\sigma_s}| > \theta$. The θ is predetermined by the given confidence interval.

We tested the effectiveness of our algorithm on the Twin-Cities track dataset and detected numerous outliers, as described in the following examples. In Figure 8(b), the abnormal station (Station 139) was detected whose volume values are significantly inconsistent with the volume values of its neighboring stations 138 and 140. Note that our basic algorithm detects outlier stations in each time slot; the detected outlier stations in each time slot are then aggregated to a daily basis.

Figure 8 shows an example of loop detector outliers. Figures 8(a) and 8(c) are the track volume maps for I-35W North Bound and South Bound, respectively, on 1/21/1997. The X-axis is the 5-minute time slot for the whole day and the Y-axis is the label of the stations installed on the highway, starting from 1

(a) Recorded GPS points from bicyclists in Minneapolis, MN. Intensity indicates number of points.

(b) Set of *8*-Primary Corridors identified from Bike GPS traces in Minneapolis, MN. (Best in color)

Figure 9: Example input and output of the k-Primary Corridor problem.

in the north end to 61 in the south end. The abnormal dark line at time slot 177 and the dark rectangle during time slot 100 to 120 on X-axis and between station 29 to 34 on Y-axis can be easily observed from both 8(a) and 8(c). This dark line at time slot 177 is an instance of temporal outliers, where the dark rectangle is a spatial-temporal outlier. Moreover, station 9 in Figure 8(a) exhibits inconsistent track data compared with its neighboring stations, and was detected as a spatial outlier.

5 Variety in Data Types: Identifying Bike Corridors

Given a set of trajectories on a road network, the goal of the k-Primary Corridors problem is to summarize trajectories into k groups, each represented by its most central trajectory. Figure 10(a) shows a real-world GPS dataset of a number of trips taken by bicyclists in Minneapolis, MN. The darkness indicates the usage levels of each road segment. The computational problem is summarizing this set of trajectories into a set of k-Primary Corridors. One potential solution is shown in Figure 9(b) for k=8. Each identified primary corridor represents a subset of bike tracks from the original dataset. Note that the output of the k-PC problem is distinct from that of hot or frequent routes, as it is a summary of all the given trajectories partitioned into k primary corridors.

The k-Primary Corridor (kPC) problem is important due to a number of societal applications, such as city-wide bus route modification or bicycle corridor selection, among other urban development applications. Let us consider the problem of determining primary bicycle corridors through a city to facilitate safe and efficient bicycle travel. By selecting representative trajectories for a given group of commuters, the overall alteration to commuters routes is minimized, encouraging use. Facilitating commuter bicycle traffic has shown in the past to have numerous societal benefits, such as reduced greenhouse gas emissions and healthcare costs [34].

Clustering trajectories on road networks is challenging due to the computational cost of computing pairwise graph-based minimum-node-distance similarity metrics between trajectories in large GPS datasets as shown in our previous work [35]. We proposed a baseline algorithm using a graph-based approach to compute a single element of the trajectory similarity matrix, requiring multiple invocations of common shortest-path algorithms (e.g., Dijkstra [36]). For example, given two trajectories consisting of 100 nodes each, a baseline approach to calculate NHD would need to compute the shortest distance between all pairs of nodes (10^4), which over a large trajectory dataset (e.g., 10,000 trajectories) would require 10^{12} shortest path distance computations. This quickly becomes computationally prohibitive without faster algorithms.

Trajectory pattern mining is a popular field with a number of interesting problems both in geometric (Euclidean) spaces [37] and networks (graphs) [38]. A key component to traditional data mining in these domains is the notion of a similarity metric, the measure of sameness or closeness between a pair of objects. A variety of trajectory similarity metrics, both geometric and network, have been proposed in the literature [39]. One popular metric is Hausdorff distance, a commonly used measure to compare similarity between two geometric objects (e.g., polygons, lines, sets of points) [40]. A number of methods have focused on applying Hausdorff distance to trajectories in geometric space [41–44].

We formalize the Network Hausdorff Distance and propose a novel approach that is orders of magnitude faster than the baseline approach and our previous work, allowing for Network Hausdorff Distance to

be computed efficiently on large trajectory datasets. A baseline approach for solving the kPC problem would involve comparing all nodes in each pairwise combination of trajectories. That is, when comparing two trajectories, each node in the first trajectory needs to find the shortest distance to any node in the opposite trajectory. The maximum value found is the Hausdorff distance [40]. While this approach does find the correct Hausdorff distance between the two trajectories, it is computationally expensive due to the node-to-node pairwise distance computations between each pair of trajectories. We will demonstrate using experimental and analytical analysis how prohibitive that cost is on large datasets. Due to this, related work solving the kPC problem has resulted in various heuristics [45–49].

We propose a novel approach that ensures correctness while remaining computationally efficient. While the baseline approach depends on computing node-to-node distances, the Network Hausdorff Distance (NHD) requires node-to-trajectory minimum distance. We take advantage of this insight to compute the NHD between nodes and trajectories directly by modifying the underlying graph and computing from a trajectory to an entire set of trajectories with a single shortest-paths distance computation. This approach retains correctness while proving significantly faster than the baseline approach and our previous work [35].

Our previous work [35] in the kPC problem requires $O(|T|^2)$ invocations of a shortest-path algorithm to compute the necessary trajectory similarity matrix (TSM), becoming prohibitively expensive when dealing with datasets with a large number of trajectories. Therefore, we developed a novel row-wise algorithm to compute the kPC problem on spatial big data.

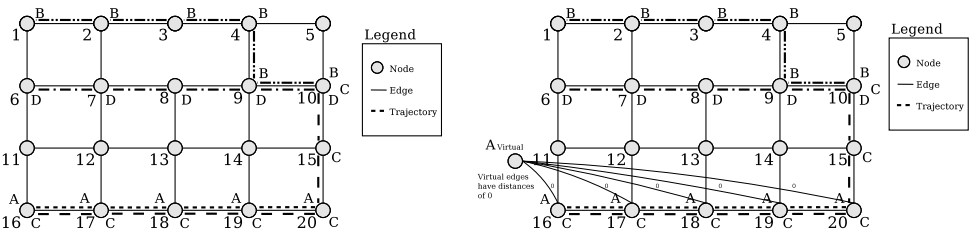
Computing the Network Hausdorff Distance $NHD(t_x, t_y)$ between two trajectories does not require the shortest distance between all-pairs of nodes in t_x and t_y. We require the shortest network distance from each node in t_x to the *closest* node in t_y. In [35], we proposed a novel approach to find this distance, as compared to enumerating the all-pair shortest-path distances as GNTS does. This significantly reduced the number of distance calculations and node iterations needed to compute the TSM. In Figure 10(b), to calculate $NHD(t_B, t_A)$, we began by inserting a virtual node ($A_{virtual}$) representing Trajectory t_A into the graph. This node had edges with weights of 0 connecting it to each other node in Trajectory t_A. We then ran a shortest-path distance computation from the virtual node as a source, with the destination being every node in Trajectory t_B. The result was the shortest distance from each node in Trajectory t_B to the virtual node $A_{virtual}$. Since the virtual node is only connected to nodes in Trajectory t_A, and all the edge weights are 0, we had the shortest-path from each node in Trajectory t_B to the closest node in Trajectory t_A, exactly what $NHD(t_B, t_A)$ requires for computation.

However, our previous work [35] focused on computing a single cell in the trajectory similarity matrix per invocation of a single-source shortest-paths algorithm. That meant at least $O(|T|^2)$ shortest-path invocations to compute the TSM for the kPC problem, still quite expensive. We propose ROW-TS to compute an entire *row* of the TSM with one invocation of a single-source shortest-paths algorithm. Using a row-based approach, we can essentially calculate $NHD(t \in T, t_A)$ with one single-source shortest-paths invocation from $A_{virtual}$. This approach reduces the overall number of shortest-paths invocations to $O(|T|)$ at the cost of additional bookkeeping, as we will show below. However, due to the expensive cost of shortest-path algorithms, this results in significant performance savings.

In Summer 2006, University of Minnesota researchers collected a variety of data to help gain a better understanding of commuter bicyclist behavior using GPS equipment and personal surveys to record bicyclist movements and behaviors [50]. The broad study examined a number of interesting issues with commuter cycling, for example, results showed that as perceived safety decreases (possibly due to nearby vehicles), riders appear to be more cautious and move more slowly. One possible issue they looked at was identifying popular transportation corridors for the Minnesota Department of Transportation to focus funds and repairs. At the time, they hand-crafted the primary corridors. Shortly after this study, the U.S. Department of Transportation began a four-year, $100 million pilot project in four communities (including Minneapolis) aimed to determine whether investing in bike and pedestrian infrastructure encouraged significant increases in public use. As a direct result of this project, the US DoT found that biking increased 50%, 7,700 fewer tons of carbon dioxide were emitted, 1.2 fewer gallons of gas was burned, and there was a $6.9 million/year reduction in health care costs [34].

6 Variety in Output: Spatial Network Activity Summarization

Spatial network activity summarization (SNAS) is important in several application domains including crime analysis and disaster response [51]. For example, crime analysts look for concentrations of individual events that might indicate a series of related crimes [52]. Crime analysts need to summarize such incidents on a map, so that law enforcement is better equipped to make resource allocation decisions [52]. In disaster



(a) Road network represented as an undirected graph with four trajectories illustrated with bold dashed lines.

(b) Inserting a virtual node ($A_{virtual}$) to represent Track A for efficient Network Hausdorff Distance computation.

Figure 10: By altering the underlying graph, we can quickly compute the Network Hausdorff Distance.

response-related applications, action is taken immediately after a disastrous event with the aim of saving life, protecting property, and dealing with immediate disruption, damage or other effects caused by the disaster [53]. Disaster response played an important role in the 2010 earthquake in Haiti, where there were many requests for assistance such as food, water and medical supplies [54]. Emergency managers need the means to summarize these requests efficiently so that they can better understand how to allocate relief supplies.

The SNAS problem is defined informally as follows: Given a spatial network, a collection of activities and their locations (e.g., a node or an edge), a set of paths P, and a desired number of paths k, find a subset of k paths in P that maximizes the number of activities on each path. An activity is an object of interest in the network. In crime analysis, an activity could be the location of a crime (e.g., theft). In disaster response-related applications, an activity might be the location of a request for relief supplies. Figures 10(a) and 9(b) illustrate an input and output example of SNAS, respectively. The input consists of six nodes, eight edges (with edge weights of 1 for simplicity), fourteen activities, the set of shortest paths for this network, and $k = 2$, indicating that two routes are desired. The output contains two routes from the given set of shortest paths that maximize the activity coverage; route $\langle C, A, B\rangle$ covers activities $1, 2, 3, 4, 5$ and route $\langle C, D, E\rangle$ covers activities $7, 8, 12, 13, 14$.

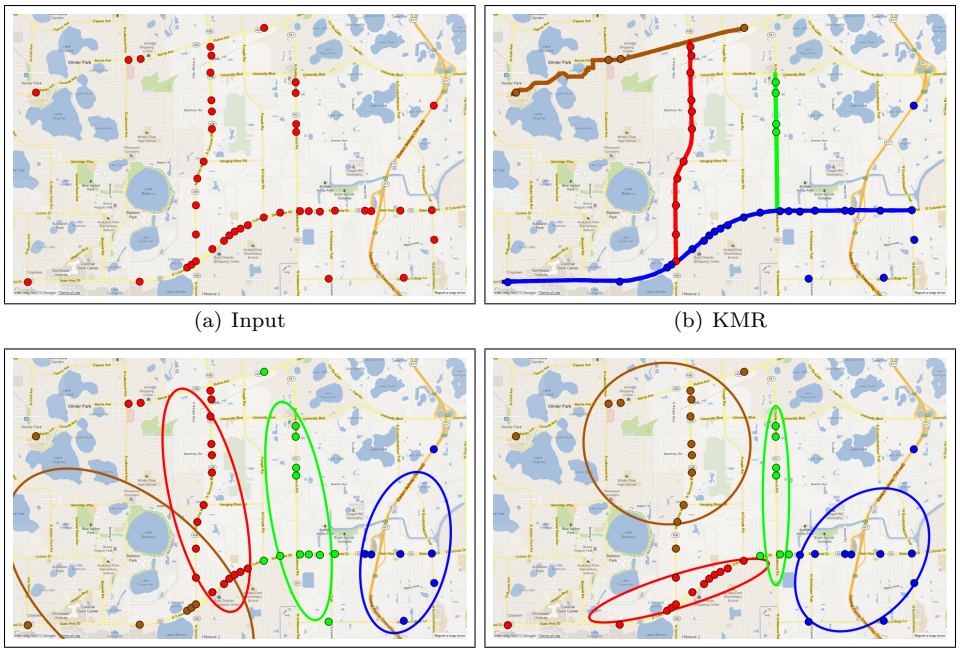
In network-based summarization, spatial objects are grouped using network (e.g., road) distance. Existing methods of network-based summarization such as Mean Streets [55], Maximal Subgraph Finding (MSGF) [56], and Clumping [57–64] group activities over multiple paths, a single path/subgraph or no paths at all. Mean Streets [55] finds anomalous streets or routes with unusually high activity levels. It is not designed to summarize activities over k paths because the number of high crime streets returned is always relatively small. MSGF [56] identifies the maximal subgraph (e.g., a single path, $k = 1$) under the constraint of a user specified length and cannot summarize activities when $k > 1$. The Network-Based Variable-Distance Clumping Method (NT-VCM) [64] is an example of the clumping technique [57–64]. NT-VCM groups activities that are within a certain shortest path distance of each other on the network; in order to run NT-VCM, a distance threshold is needed.

In this chapter, we propose a K-Main Routes (KMR) approach that finds a set of k routes to summarize activities. KMR aims to maximize the number of activities covered on each k route. KMR employs an inactive node pruning algorithm where instead of calculating the shortest paths between all pair of nodes, only shortest paths between active nodes and all other nodes in the spatial network are calculated. This results in computational savings (without affecting the resulting summary paths) that are reported in the experimental evaluation. The inputs of KMR include the following: 1) an undirected spatial network $G = (N, E)$, 2) a set of activities A and 3) a number of routes, k, where $k \geq 1$. The output of KMR is a set of k routes where the objective is to maximize the activity coverage of each k route. Each k route is a shortest path between its end-nodes and each activity $a_i \in A$ is associated with only one edge $e_i \in E$.

KMR first calculates shortest paths between active nodes and all other nodes and then selects k shortest paths as initial summary paths. The main loop of KMR assigns and updates steps of phases 1 and 2 are repeated until the summary paths do not change.

Phase 1: Assign activities to summary paths. The first step of this phase initializes the set of next clusters, i.e., $nextClusters$, to the empty set. In general, this phase is concerned with forming k clusters by assigning each activity to its nearest summary path. To accomplish this, KMR considers each activity and each cluster in determining the nearest summary path to an activity.

KMR uses the following proximity measure to quantify nearness: $prox(s_i, a_i)$. The distance between an activity and a summary path is the minimum network distance between each node of the edge that



(a) Input (b) KMR

(c) Crimestat K-means with Euclidean Distance (d) Crimestat K-means with Network Distance

Figure 11: Comparing KMR and Crimestat K-means output for $k = 4$ on pedestrian fatality data from Orlando, FL [65].

the activity is on and each node of the summary path. Once the distance from each activity to each summary path is calculated, the activity is assigned to the cluster with the nearest summary path.

Phase 2: Recompute summary paths. This phase is concerned with recomputing the summary path of each cluster so as to further maximize the activity coverage. This entails iterating over each cluster and initializing the summary path for each cluster. The summary path of each cluster is updated based on the activities assigned to the cluster. The summary path with the maximum activity coverage is chosen as the new summary path for each cluster c_i, i.e., $sp_{max} \leftarrow max(AC(sp_k) \in c_i \mid k = 1...|sp|)$. Phases 1 and 2 are repeated until the summary paths of each cluster do not change. At the end of each iteration the current clusters are initialized to the next clusters. Once the summary paths of each cluster do not change, a set of k routes is returned.

We conducted a qualitative evaluation of KMR comparing its output with the output of Crimestat [66] K-means [67] (a popular summarization technique) on a real pedestrian fatality dataset [65], shown in Figure 11(a). The input consists of 43 pedestrian Fatalities (represented as dots) in Orlando, Florida occurring between 2000 and 2009. As we have explained, KMR uses paths and network distance to group activities on a spatial network. By contrast, in geometry-based summarization, the partitioning of spatial data is based on grouping similar points distributed in planar space where the distance is calculated using Euclidean distance. Such techniques focus on the discovery of the geometry (e.g., circle, ellipse) of high density regions [52] and include K-means [**?**,67–71], K-medoid [72,73], P-median [74] and Nearest Neighbor Hierarchical Clustering [75] algorithms.

Figures 11(b), 11(c), and 11(d) show the results of KMR, K-means using Euclidean distance, and K-Means using network distance, respectively. In all cases, K was set to 4. The output of each technique shows (1) the partitioning of activities represented by different colors and (2) the representative of each partition (e.g., paths or ellipses). For example, Figure 11(c) shows (1) activities that are colored brown, red, green, and blue, representing the four different partitions to which each activity belongs and (2) ellipses representing each partition of activities (e.g., the red ellipse is the representative for the red partition of activities).

This work explored the problem of spatial network activity summarization in relation to important application domains such as crime analysis and disaster response. We proposed a K-Main Routes (KMR) algorithm that discovers a set of k paths to group activities. KMR uses inactive node pruning, Network Voronoi activity Assignment and Divide and Conquer Summary Path Recomputation to enhance its performance and scalability. Experimental evaluation using both synthetic and real-world datasets

indicated that the performance-tuning decisions utilized by KMR yielded substantial computational savings without reducing the coverage of the resulting summary paths. For qualitative evaluation, a case study comparing the output of KMR with the output of a current geometry-based summarization technique highlighted the potential usefulness of KMR to summarize activities on spatial networks.

7 Summary

Increasingly, location-aware datasets are of a size, variety, and update rate that exceed the capability of spatial computing technologies. This chapter discussed some of the emerging challenges posed by such datasets, referred to as Spatial Big Data (SBD). SBD examples include trajectories of cell-phones and GPS devices, temporally detailed (TD) road maps, vehicle engine measurements, etc. SBD has the potential to transform society. A recent McKinsey Global Institute report estimates that spatial big data, such as personal location data, could save consumers hundreds of billions of dollars annually by 2020 by helping vehicles avoid congestion via next-generation routing services such as eco-routing.

Spatial Big Data has immense potential to benefit a number of societal applications. By harnessing this increasingly large, varied and changing data, new opportunities to solve worldwide problems are presented. To capitalize on these new datasets, inherent challenges that come with spatial big data need to be addressed. For example, many spatial operations are iterative by nature, something that parallelization has not yet been able to handle completely. By expanding cyber-infrastructure, we can harness the power of these massive spatial datasets. New forms of analytics using simpler models and richer neighborhoods will enable solutions in a variety of disciplines.

References

[1] J. Manyika *et al.*, "Big data: The next frontier for innovation, competition and productivity," *McKinsey Global Institute, May*, 2011.

[2] S. Shekhar, M. Evans, J. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 193–214, 2011. Wiley Online Library.

[3] New York Times, "Military Is Awash in Data From Drones." `http://www.nytimes.com/2010/01/11/business/11drone.html?pagewanted=all`, 2010.

[4] G. Levchuk, A. Bobick, and E. Jones, "Activity and function recognition for moving and static objects in urban environments from wide-area persistent surveillance inputs," in *Proceedings of SPIE*, vol. 7704, p. 77040P, 2010.

[5] New York Times, "Mapping Ancient Civilization, in a Matter of Days." `http://www.nytimes.com/2010/05/11/science/11maya.html`, 2010.

[6] InformationWeek, "Red Cross Unveils Social Media Monitoring Operation." `http://www.informationweek.com/government/information-management/red-cross-unveils-social-media-monitorin/232602219`, 2012.

[7] Garmin. `http://www.garmin.com/us/`.

[8] Wikipedia, "Usage-based insurance — wikipedia, the free encyclopedia." `http://goo.gl/NqJE5`, 2011. [Online; accessed 15-December-2011].

[9] TomTom, "TomTom GPS Navigation." `http://www.tomtom.com/`, 2011.

[10] Google Maps. `http://maps.google.com`.

[11] H. Kargupta, J. Gama, and W. Fan, "The next generation of transportation systems, greenhouse emissions, and data mining," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1209–1212, ACM, 2010.

[12] H. Kargupta, V. Puttagunta, M. Klein, and K. Sarkar, "On-board vehicle data stream monitoring using minefleet and fast resource constrained monitoring of correlation matrices," *New Generation Computing*, vol. 25, no. 1, pp. 5–32, 2006. Springer.

[13] Lynx GIS. `http://www.lynxgis.com/`.

[14] MasterNaut, "Green Solutions." `http://www.masternaut.co.uk/carbon-calculator/`.

[15] TeleNav. `http://www.telenav.com/`.

[16] TeloGIS. http://www.telogis.com/.

[17] G. Capps, O. Franzese, B. Knee, M. Lascurain, and P. Otaduy, "Class-8 heavy truck duty cycle project final report," *ORNL/TM-2008/122*, 2008.

[18] A. T. R. I. (ATRI), "Fpm congestion monitoring at 250 freight significant highway location: Final results of the 2010 performance assessment." http://goo.gl/3cAjr, 2010.

[19] A. T. R. I. (ATRI), "Atri and fhwa release bottleneck analysis of 100 freight significant highway locations." http://goo.gl/CONuD, 2010.

[20] D. Sperling and D. Gordon, *Two billion cars*. Oxford University Press, 2009.

[21] Federal Highway Administration, "Highway Statistics," *HM-63, HM-64*, 2008.

[22] B. George and S. Shekhar, "Road maps, digital," in *Encyclopedia of GIS*, pp. 967–972, Springer, 2008.

[23] S. Shekhar and H. Xiong, *Encyclopedia of GIS*. Springer Publishing Company, Incorporated, 2007.

[24] NAVTEQ. www.navteq.com.

[25] I. Noble, "A model of the responses of ecotones to climate change," *Ecological Applications*, vol. 3, no. 3, pp. 396–403, 1993. JSTOR.

[26] Tucker, C.J., J.E. Pinzon, M.E. Brown, "Global inventory modeling and mapping studies." Global Land Cover Facility, University of Maryland, College Park, Maryland, 1981-2006.

[27] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. JSTOR.

[28] D. Nikovski and A. Jain, "Fast adaptive algorithms for abrupt change detection," *Machine learning*, vol. 79, no. 3, pp. 283–306, 2010. Springer.

[29] M. Sharifzadeh, F. Azmoodeh, and C. Shahabi, "Change detection in time series data using wavelet footprints," *Advances in Spatial and Temporal Databases*, pp. 127–144, 2005. Springer.

[30] J. Kucera, P. Barbosa, and P. Strobl, "Cumulative sum charts-a novel technique for processing daily time series of modis data for burnt area mapping in portugal," in *Analysis of Multi-temporal Remote Sensing Images, 2007. MultiTemp 2007. International Workshop on the*, pp. 1–6, IEEE.

[31] J. Canny, "A computational approach to edge detection," *Readings in computer vision: issues, problems, principles, and paradigms*, vol. 184, no. 87-116, p. 86, 1987. Morgan Kaufmann.

[32] X. Zhou, S. Shekhar, P. Mohan, S. Liess, and P. Snyder, "Discovering interesting sub-paths in spatiotemporal datasets: A summary of results," *Proceedings of the 19th International Conference on Advances in Geographical Information Systems (ACMGIS 2011)*, vol. November 1-4, 2011, Chicago, IL, USA. ACM.

[33] S. Shekhar, C.-T. Lu, and P. Zhang, "Detecting graph-based spatial outliers: algorithms and applications (a summary of results)," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 371–376, ACM, 2001.

[34] J. Marcotty, "Federal Funding for Bike Routes Pays Off in Twin Cities." http://www.startribune.com/local/minneapolis/150105625.html.

[35] M. R. Evans, D. Oliver, S. Shekhar, and F. Harvey, "Summarizing trajectories into k-primary corridors: a summary of results," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12, (New York, NY, USA), pp. 454–457, ACM, 2012.

[36] T. Cormen, *Introduction to algorithms*. The MIT press, 2001.

[37] Y. Zheng and X. Zhou, *Computing with Spatial Trajectories*. Springer Publishing Company, Incorporated, 1st ed., 2011.

[38] R. H. Güting, V. T. De Almeida, and Z. Ding, "Modeling and querying moving objects in networks," *The VLDB Journal*, vol. 15, no. 2, pp. 165–190, 2006.

[39] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 312–319, IEEE, 2009.

[40] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg, "On dynamic voronoi diagrams and the minimum hausdorff distance for point sets under euclidean motion in the plane," in *Proceedings of the eighth annual symposium on Computational geometry*, pp. 110–119, ACM, 1992.

[41] J. Henrikson, "Completeness and total boundedness of the hausdorff metric," *MIT Undergraduate Journal of Mathematics*, vol. 1, pp. 69–79, 1999.

[42] S. Nutanong, E. H. Jacox, and H. Samet, "An incremental hausdorff distance calculation algorithm," *Proceedings of the VLDB Endowment*, vol. 4, no. 8, pp. 506–517, 2011.

[43] J. Chen, R. Wang, L. Liu, and J. Song, "Clustering of trajectories based on hausdorff distance," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pp. 1940–1944, IEEE, 2011.

[44] H. Cao and O. Wolfson, "Nonmaterialized motion information in transport networks," *Database Theory-ICDT 2005*, pp. 173–188, 2005.

[45] G. Roh and S. Hwang, "Nncluster: An efficient clustering algorithm for road network trajectories," in *Database Systems for Advanced Applications*, pp. 47–61, Springer, 2010.

[46] J.-R. Hwang, H.-Y. Kang, and K.-J. Li, "Spatio-temporal similarity analysis between trajectories on road networks," in *Proceedings of the 24th international conference on Perspectives in Conceptual Modeling*, ER'05, (Berlin, Heidelberg), pp. 280–289, Springer-Verlag, 2005.

[47] J.-R. Hwang, H.-Y. Kang, and K.-J. Li, "Searching for similar trajectories on road networks using spatio-temporal similarity," in *Advances in Databases and Information Systems*, pp. 282–295, Springer, 2006.

[48] E. Tiakas, A. N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan, "Searching for similar trajectories in spatial networks," *J. Syst. Softw.*, vol. 82, pp. 772–788, May 2009.

[49] E. Tiakas, A. N. Papadopoulos, A. Nanopoulos, Y. Manolopoulos, D. Stojanovic, and S. Djordjevic-Kajan, "Trajectory similarity search in spatial networks," in *Database Engineering and Applications Symposium, 2006. IDEAS'06. 10th International*, pp. 185–192, IEEE, 2006.

[50] F. Harvey and K. Krizek, "Commuter Bicyclist Behavior and Facility Disruption," Tech. Rep. Report no. MnDOT 2007-15, University of Minnesota, 2007.

[51] D. Oliver, A. Bannur, J. M. Kang, S. Shekhar, and R. Bousselaire, "A k-main routes approach to spatial network activity summarization: A summary of results," in *ICDM Workshops*, pp. 265–272, 2010.

[52] J. Eck, S. Chainey, J. Cameron, M. Leitner, and R. Wilson, "Mapping crime: Understanding hot spots," 2005.

[53] W. Carter, *Disaster management: A disaster manager's handbook*. Asian Development Bank, 1991.

[54] Crisis Map of Haiti. http://haiti.ushahidi.com/.

[55] M. Celik, S. Shekhar, B. George, J. Rogers, and J. Shine, "Discovering and quantifying mean streets: A summary of results," tech. rep., Technical Report 07-025, University of Minnesota, Computer Science and Engineering, 2007.

[56] K. Buchin, S. Cabello, J. Gudmundsson, M. Löffler, J. Luo, G. Rote, R. I. Silveira, B. Speckmann, and T. Wolle, "Detecting Hotspots in Geographic Networks," *Advances in GIScience*, pp. 217–231.

[57] S. Roach, *The theory of random clumping*. Methuen, 1968.

[58] A. Okabe, K. Okunuki, and S. Shiode, "The SANET toolbox: New methods for network spatial analysis," *Transactions in GIS*, vol. 10, no. 4, pp. 535–550, 2006.

[59] S. Shiode and A. Okabe, "Network variable clumping method for analyzing point patterns on a network," in *Unpublished paper presented at the Annual Meeting of the Associations of American Geographers, Philadelphia, Pennsylvania*, 2004.

[60] K. Aerts, C. Lathuy, T. Steenberghen, and I. Thomas, "Spatial clustering of traffic accidents using distances along the network," in *Proc. 19th Workshop of the International Cooperation on Theories and Concepts in Traffic Safety*, 2006.

[61] P. Spooner, I. Lunt, A. Okabe, and S. Shiode, "Spatial analysis of roadside Acacia populations on a road network using the network K-function," *Landscape ecology*, vol. 19, no. 5, pp. 491–499, 2004.

[62] T. Steenberghen, T. Dufays, I. Thomas, and B. Flahaut, "Intra-urban location and clustering of road accidents using GIS: a Belgian example," *International Journal of Geographical Information Science*, vol. 18, no. 2, pp. 169–181, 2004.

[63] I. Yamada and J. Thill, "Local indicators of network-constrained clusters in spatial point patterns," *Geographical Analysis*, vol. 39, no. 3, pp. 268–292, 2007.

[64] S. Shiode and N. Shiode, "Detection of multi-scale clusters in network space," *International Journal of Geographical Information Science*, vol. 23, no. 1, pp. 75–92, 2009.

[65] Fatality Analysis Reporting System (FARS), National Highway Traffic Safety Administration (NHTSA), http://www.nhtsa.gov/FARS.

[66] N. Levine, "CrimeStat: A spatial statistics program for the analysis of crime incident locations (v 2.0)," *Ned Levine & Associates, Houston, TX, and the National Institute of Justice, Washington, DC*, 2002.

[67] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, p. 14, California, USA, 1967.

[68] S. Borah and M. Ghose, "Performance analysis of AIM-K-means & K-means in quality cluster generation," *Arxiv preprint arXiv:0912.3983*, 2009.

[69] A. Barakbah and Y. Kiyoki, "A pillar algorithm for K-Means optimization by distance maximization for initial centroid designation," *IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Nashville-Tennessee*, 2009.

[70] S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1293–1302, 2004.

[71] D. Pelleg and A. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, San Francisco, 2000.

[72] L. Kaufman and P. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. Wiley Online Library, 1990.

[73] R. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 144–144, Citeseer, 1994.

[74] M. Resende and R. Werneck, "A hybrid heuristic for the p-median problem," *Journal of Heuristics*, vol. 10, no. 1, pp. 59–88, 2004.

[75] R. D'Andrade, "U-statistic hierarchical clustering," *Psychometrika*, vol. 43, no. 1, pp. 59–67, 1978.