Riding from Urban Data to Insight Using New York City Taxis

Juliana Freire[1,2] Cláudio Silva[1,2] Huy Vo[1,2]
Harish Doraiswamy[1] Nivan Ferreira[1] Jorge Poco[1]
{juliana.freire,csilva,hvo,harishd,nivan.ferreira,jpocom}@nyu.edu
[1]Department of Computer Science and Engineering
[2] Center for Urban Science and Progress
New York University

Abstract

About half of humanity lives in urban environments today and that number will grow to 80% by the middle of this century. Cities are thus the loci of resource consumption, of economic activity, and of innovation. Given our increasing ability to collect, transmit, store, and analyze data, there is a great opportunity to better understand cities, and enable them to deliver services efficiently and sustainably while keeping their citizens safe, healthy, prosperous, and well-informed. But making sense of all the data available is hard. Currently, urban data exploration is often limited to confirmatory analyses consisting of batch-oriented queries and the exploration of well-defined questions over specific regions. The lack of interactivity makes this process both time-consuming and cumbersome. This problem is compounded in the presence of big, multivariate spatio-temporal data, which is ubiquitous in urban environments. Another challenge comes from the need to empower social scientists, policy makes and urban residents who lack computer science expertise to leverage these data. In this paper, we give an overview of our recent work on techniques that combine data management and visualization to enable a broad set of users to interactively explore large, spatio-temporal data. We describe a visual query interface that simplifies the process of specifying spatio-temporal queries as well as new indexing technique that enables these queries to be evaluated at interactive rates. We also present a scalable framework that applies computational topology to automatically find interesting *data slices so as to help guide users in the exploratory process.*

1 Introduction

Today, 50% of the world's population lives in cities and the number will be 70% by 2050; North America is already 80% in cities, rising to 90% by 2050 [46]. Cities are thus the loci of economic activity and innovation. At the same time, most cities face huge challenges around transportation, resource consumption, housing affordability, and inadequate or aging infrastructure. Data, along with visualization and analytics capabilities, can help significantly with these challenges.

Our increasing ability to collect, transmit, and store data, coupled with the growing trend towards openness [4, 18, 20, 23, 36, 37, 43], creates a unique opportunity that can benefit government, science, citizens and industry. By integrating and analyzing multiple data sets, city governments can go beyond today's imperfect and

Copyright 0000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

often anecdotal understanding of cities to enable better operations and informed planning [16, 20]. Scientists can engage in data-driven science and explore longitudinal processes to understand people's behavior [21]; identify causal relationships across data sets, which can in turn, influence policy decisions [11, 42]; or create models and derive predictions that benefit citizens [15]. Putting urban data in the hands of citizens has the potential to improve governance and participation, and in the hands of entrepreneurs and corporations it will lead to new products and services for governments, firms, and consumers. The challenge now lies in making sense of all the data so that they can be used effectively to answer the right questions.

Urban data is unique in that it captures the behavior of the different components of a city, namely its citizens, existing infrastructure (physical and policies), the environment (e.g., weather), and interactions between these elements [28]. To understand a city and how its components interact, intricate analyses are necessary. These include flexible exploration and visualizations that span different geographical regions and multiple time slices. However, urban data analysis has often been limited to well-defined questions, or confirmatory data analysis [45]. The common practice is for domain experts to formulate hypotheses on the basis of theory and anecdotal experience, then for data scientists with expertise in geographical information systems (GIS) and statistical analysis tools to select relevant data, carry out analyses, and finally, the domain experts can inspect the results to verify whether they disprove or support the hypotheses. In order to answer even simple questions, a large number of plots and tables may have to be generated, each of which is individually programmed and manually composed for analysis. Because data selection is often decoupled from the analysis and visualization, the context switch between the different software components used for these tasks makes it hard to keep spatio-temporal context. Together with the glut of plots derived, this creates a heavy cognitive load for the users, while the batch-oriented analysis pipeline hampers exploration across data sets, which is essential for understanding trends and potential causal mechanisms. Furthermore, the dependency on data specialists distances the domain experts from the data, limiting their opportunity to explore new directions. The trend towards broad-scale data collection, rather than limited collection targeted at specific questions, makes it clear that this process cannot scale, and that tools are needed that foster hypothesis-generating analyses.

One of the shifts we're seeing with observational data is broad-scale collection, rather than limited collection targeted at one hypothesis. So the thought is that the kind of tools that worked for analyzing a "single hypothesis" dataset might not be a good match for the kind of exploratory,

The lack of interactivity, along with the recent explosion in data volume and complexity, make it clear that this process cannot scale.

An important goal of our research is to *enable domain experts to freely explore a large number of urban data sets and interactively analyze the many different facets of these data*. This involves fundamental challenges. First and foremost, we need usable tools, designed for users who do not have computer science training. Second, not only can urban data be large, but often they contain both temporal and spatial components, in addition to multiple variables. In a recent study of open data published by cities in North America, we found that over 50% of the tabular data contained spatial attributes and roughly 48% included time information [4]. Attaining interactivity while exploring spatio-temporal data is difficult. Even though there has been substantial work on spatio-temporal indexing, most techniques aim to speed up batch queries, and are not able to support the query rates that interactive visual analytics applications demand. Another challenge comes from the fact that there are too many data slices to explore, that cover different regions and time ranges, making it hard to identify interesting patterns or events.

In this paper, we give an overview of recent work that combines data management and visualization to support the interactive exploration of large urban data [9, 14]. We use New York City taxi data as a case study to both illustrate general challenges that arise in urban data exploration, and to demonstrate the effectiveness and usefulness of the techniques we have developed. We describe the taxi data in Section 2. In Section 3, we present TaxiVis, a visual analytics tool that allows users to specify complex spatio-temporal queries through a visual interface. We describe the visual language as well as a new indexing strategy that allows queries to be evaluated at interactive rates. TaxiVis also implements a number of visualization and interaction techniques to

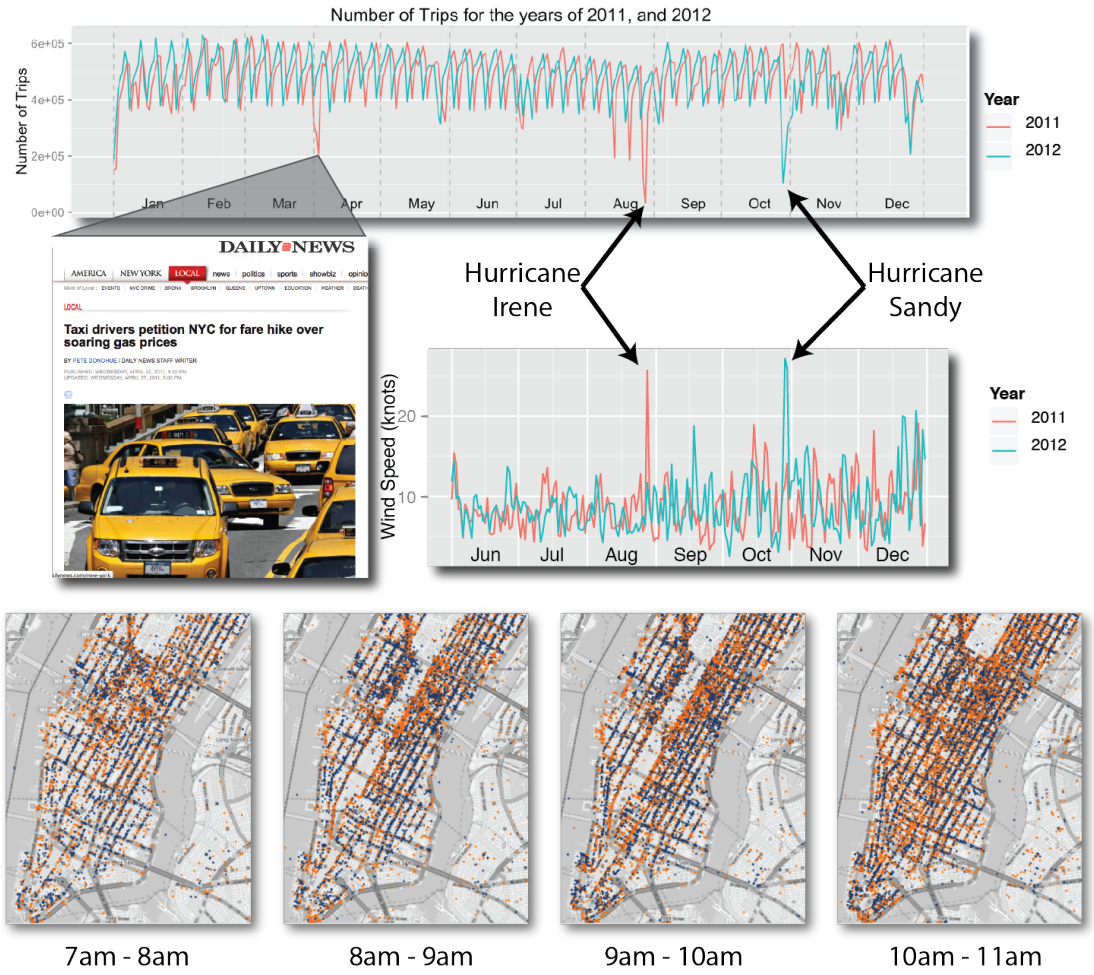

Figure 1: Taxis as sensors of city life. The plot on the top shows how the number of trips varies over 2011 and 2012. While some patterns are regular and appear on both years, some anomalies are clear, e.g., the drops in August 2011 (Hurricane Irene) and in October 2012 (Hurricane Sandy). Another large discrepancy happened in April 2011, when gas prices increased. In the bottom, we show pickups (blue) and dropoffs (orange) in Manhattan on May 1st from 7am to 11am. Notice that from 8-10am, there are virtually no trips along 6th Avenue, indicating the traffic was blocked.

streamline exploration, including, coordinated views and parameter sweeps. The system has been successfully deployed at two New York City agencies: the Taxi & Limousine Commission (TLC) and at the Department of Transportation (DoT), where users have lauded its usability and speed. But even with such a system, looking for interesting patterns across a very large number of spatio-temporal slices can be like looking for a needle in a haystack. To help guide users towards interesting times slices, we designed a new topology-based event detection technique that is both scalable and, unlike existing approaches, can detect events with arbitrary spatial geometry. We describe the technique in Section 4 and show how it can support event queries. We conclude in Section 5, where we discuss open problems and directions for future work.

2 Exploring New York City Taxi Data: Opportunities and Challenges

Taxis are central component of the New York City transportation system. Every day, there are over 500,000 taxi trips transporting about 600,000 people [44]. Through the meters installed in each vehicle, the Taxi & Limousine Commission (TLC) captures detailed information about these trips, which consists of two spatial

attributes (pickup and dropoff locations), two temporal attributes (pickup and dropoff times), and additional attributes including taxi id, distance traveled, fare and tip amount. The large number of vehicles, trips they make, and people they carry, make taxis valuable sensors that can provide unprecedented insight into many different aspects of city life, from economic activity and human behavior to mobility patterns. For example, the transactional attributes present in the taxi data enables the study of the economics of fare structure and optimal fleet size [40, 41].

Consider the plot in Figure 1 (top), which shows how the number of trips per day varies over 2011 and 2012. Note the regularity in the trip distribution over the two years. For example, on Thanksgiving, Christmas and New Year's eve, there is a substantial drop in the number of trips. But the plot also shows some anomalies. There are big drops in August 2011 and October 2012, which correspond to hurricanes Irene and Sandy, respectively. Looking at the data at a finer scale, other interesting patterns emerge. The maps in Figure 1 (bottom) show the density of taxis across Manhattan from 7am to 11am, on May 1st, 2011. From 8am to 10am, taxis disappear along 6th avenue, from Midtown to Downtown; and then, at 10am they reappear. As it turns out, during this period, the streets were closed to traffic for the NYC Five Boro Bike Tour.[1] Other useful information can also be discovered by analyzing the taxi data set, from popular night spots and economically disadvantaged neighborhoods that are underserved by taxis, to mobility patterns across regions at different times and days.

Not surprisingly, exploring these data is challenging due to its size and complexity. There are over 170 million taxi trips in NYC every year. The current approach used by domain experts is to store the data in a general-purpose relational DMBS and perform queries to answer questions posed out of intuition or field observation. The results of these queries are then fed into different software tools such as R, Excel and ArcGIS for further analysis and visualization. This workflow makes the analysis process inefficient. The disconnect between the data selection process and visualization hampers exploration. The context switch between the different software components creates a heavy cognitive load for the users and makes it hard to keep spatio-temporal context. There is also a steep learning curve for users to master these tools. Furthermore, off-the-shelf DMBS are not built for interactivity. For example, common queries over on the taxi data (even in the presence of spatial indexes) range from tens of seconds to minutes (see Section 3). These response times are not acceptable for interactive visual analytics, since users perceive questioning and answering as separate tasks [17].

3 Visually Exploring Spatio-Temporal Data

Visualization and visual analytics systems help people explore and explain data by allowing the creation of both static and interactive visual representations [24, 25, 29, 31, 30, 47]. A basic premise of visualization is that visual information can be processed at a much higher rate than raw numbers and text: as the cliché goes, "A picture is worth a thousand words." Well-designed visualizations substitute perception for cognition, freeing up limited cognitive and memory resources for higher-level problems [35]. There are several visualization tools that give users access to advanced visualization techniques. But the application of visualization technology to large data is non-trivial.

A widely-used method to analyze large data is to take a small subset of the data (often by sub-sampling) and study it with existing (non-scalable) tools. Hypotheses are generated from this sample, which are then tested on the complete data set through confirmatory analysis [17]. This approach has many shortcomings, one of them is the potential bias introduced by the use of small samples. This is further exacerbated by high-dimensional data that comes from multiple sources. Patterns that might be easy to find on the complete data sets might be obscured in small samples.

Working on the whole data set has many advantages but comes at a high computational cost. This creates new challenges for data management systems, since to be effective, visualization tools must be interactive, requiring sub-second response times. In a recent study, Liu and Heer [33] concluded that even relatively short delays

[1] http://www.nycbikemaps.com/spokes/five-boro-bike-tour-sunday-may-1st-2011

in visualization systems can harm user activity, data set coverage, and how many observations and hypotheses are generated. Fekete and Silva [12, 13] argued that although there has been much work on scaling databases for big data, existing technologies do not meet the requirements needed to interactively explore massive or even reasonably sized data sets. Recognizing this limitation, several recent works have started to address the problems of providing efficient support for visualization [49] and interactive queries over large tabular data [2, 5, 26, 27, 32, 34, 48].

In what follows, we present our approach to support interactive exploration of spatio-temporal data, which was implemented in the TaxiVis system.

3.1 The TaxiVis System

TaxiVis was designed to support interactive analysis of NYC taxi data. It implements a visual model that is able to express complex queries and an index structure that enables interactive response times for spatio-temporal queries. We describe these two components below, and discuss how they were combined in the TaxiVis system.

The query model enable users to pose queries over all the dimensions of the data and flexibly explore the attributes associated with the taxi trips. The components of the user interface are shown in Figure 2. Queries can be interactively composed and refined in the Map view (B), as well as generalized by performing parameter sweeps. Query results can be visualized in the data summary view (D) in a variety of ways including time-series plots, histograms, scatterplots and heatmaps. By combining data selection and result visualization in the same environment, TaxiVis allows users to explore multiple data slices while maintaining the spatio-temporal context. The system also implements a number of strategies to render a large number of graphical primitives on a map, as well as the use of adaptive level-of-detail rendering to provide clutter-free visualization of the results (see Figure 3). TaxiVis also implements a number of visualization and interaction techniques to streamline exploration, including, multiple coordinated views and parameter sweeps. The former is illustrated in Figure 3, which shows a comparison of the number of pickups in different neighborhoods on Mondays and Sundays.

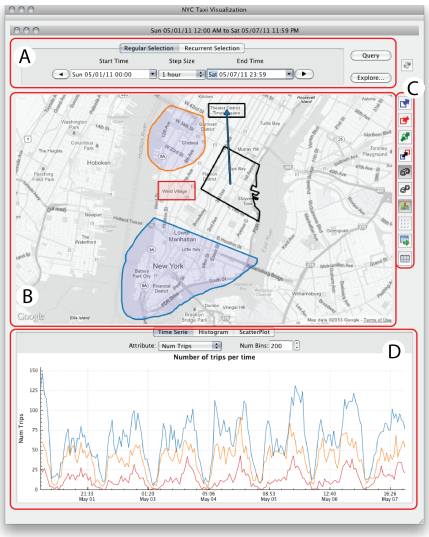

Figure 2: TaxiVis user interface components. (A) Time selection widget, (B) Map, (C) Tool bar, and (D) Data summary.

TaxiVis is currently being used at the NYC Department of Transportation (DoT) and the TLC. The feedback we have received from them was very positive. The analysts stated that "The speed at which the tool permits us to work has saved multiple hours of staff time and has dramatically improved the unit's output and capabilities". We should note that while the original motivation to build TaxiVis was to analyze taxi data, we have used the system to explore other spatio-temporal data sets, including: NYC CitiBike, property ownership [22], 311 complaints [1], geo-tagged tweets, and energy consumption.

3.2 Visual Query Model

To address the usability limitations of existing tools, we designed a new visual query model that supports complex spatio-temporal queries over origin-destination data [14]. Users need not be experts in any textual query language: users specify queries visually and they can iteratively refine their queries through direct manipulation of the results. The model is expressive and supports a wide class of queries, including the query classes for spatio-temporal data defined in Peuquet's triad framework [39].

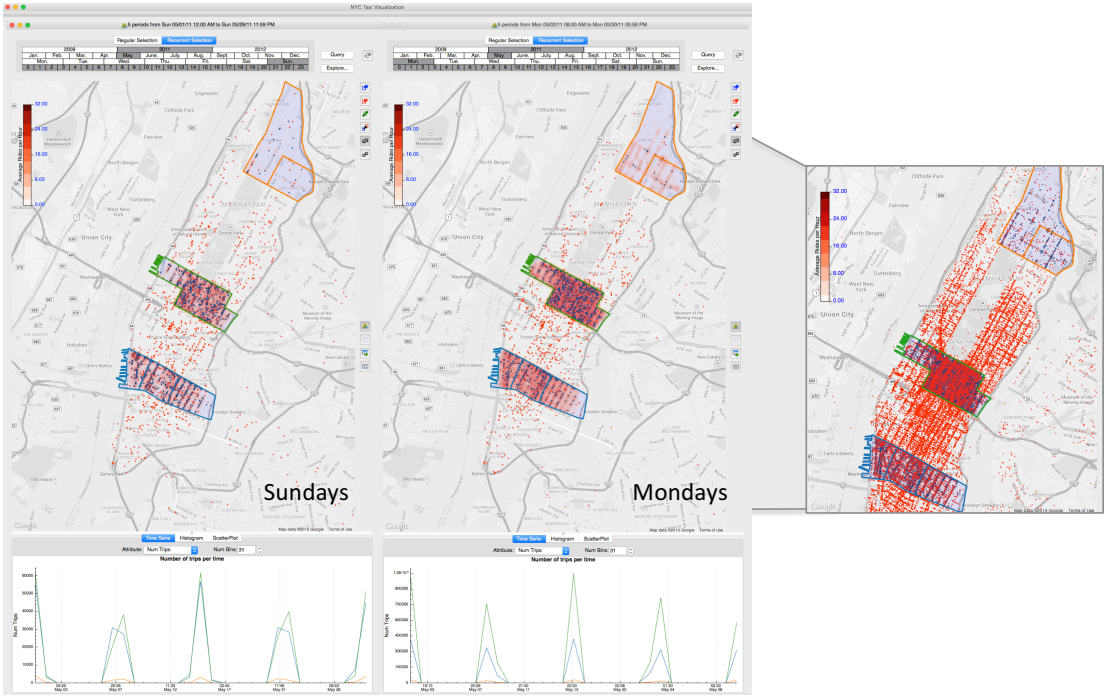

Figure 3: Using multiple views to explore how the number of pickups varies on Sundays and Mondays (in May 2011) for different neighborhoods. While there are many more pickups Midtown (green) throughout the day on Mondays, on Sundays, West, East and Greenwich Village (blue) get busier. This analysis also shows that Harlem (orange) is underserved by taxis. Because queries can return a large number of results, TaxiVis uses adaptive level-of-detail (LOD) rendering to display the results on the map. As shown on the right, without LOD, the map gets cluttered.

In our model, queries are of the following form: **SELECT** * **FROM** trips **WHERE** <constraints>. The general idea is to have users specify the constraints for this query template through visual operations. There are three types of constraints that correspond to the components of the data: *spatial, temporal, and attributes*. Each query is associated with the set of trips contained in its results. Since each trip is uniquely identified by the trip id, *queries can be composed*: users can iteratively refine queries and further explore the results. This has two important implications: it allows the creation of summaries and visualizations while maintaining the spatial and temporal contexts, and enables queries to be applied directly to the derived visualizations. To formalize the process of query composition and properly define query semantics, we use two types of queries: *atomic* queries and *complex* queries, where the latter uses atomic queries as building blocks.

Atomic Queries. An atomic query consists of a set of temporal, attribute and spatial constraints. *Temporal constraints* define intervals that bound the values of the time range of the query. A temporal constraint is specified by an interval $[t_{Min}, t_{Max}]$. A trip satisfies the constraint if $trip.pickup\_time$, $trip.dropoff\_time \in [t_{Min}, t_{Max}]$. It is also possible to have constraints that bound just the pickup or the dropoff time.

An *attribute constraint* can be expressed using equality conditions (for categorical attributes) or interval conditions (for numerical attributes). A trip satisfies an attribute equality constraint associated with a categorical attribute A if for the given value a, $trip.A = a$. If the constraint is associated with a numerical attribute, the trip satisfies the constraint for the interval $[l_A, r_A]$ if $trip.A \in [l_A, r_A]$.

Spatial constraints come in two flavors: single-region and directional constraints. A single-region constraint is defined by a connected spatial region and is associated either with the pickup location (start constraint) or

the dropoff location (destination constraint). A trip satisfies the constraint for region r if *trip.pickup_region* $\in r$ (for start constraints) or *trip.dropoff_region* $\in r$ (for destination constraints). Directional constraints are used to construct queries about origins and destinations. A directional constraint bounds the regions associated with both pickup and dropoff locations. Given source and destination regions, r_{source} and r_{dest}, respectively, a trip satisfies the constraint if *trip.pickup_location* $\in r_{source}$ and *trip.dropoff_location* $\in r_{dest}$.

We define a function called *result* which takes as input an atomic query and returns the set of all *trip* records that satisfy the query constraints. The *result* function determines how queries are evaluated. Atomic queries are *closed under intersection*, *i.e.*, since the query constraints are *closed under intersection* we can combine atomic queries to construct new ones. We do so by taking two atomic queries Q_1, Q_2, and constructing a third query, Q_3, which is given by the intersection of the corresponding constraints. By definition, we have $result(Q_3) = result(Q_1) \cap result(Q_2)$.

Complex Queries. A complex query is constructed by combining a set of atomic queries through disjunction. We evaluate those queries by extending the *result* function inductively. Note that an atomic query is a special case of a complex query, where the query set has a single element. Given two complex queries, Q_1 and Q_2, $result(Q_1 \cup Q_2) = result(Q_1) \cup result(Q_2)$. In general, given an atomic query Q it is not possible to find an atomic query Q' such that $result(Q') = result(Q)^C$ (the complement of $result(Q)$). However, it is always possible to define a complex query Q' that satisfies this condition. Thus, set theoretic operations can be performed on the result of complex queries to build new complex queries.

Visual Representation. Figure 2 illustrates how atomic and complex queries are represented visually in our system. Temporal constraints are specified using time-selection widgets (A), and attribute constraints are defined in a separate view (see [14] for details). Here, to illustrate the semantics of the query model, we focus on spatial views that are defined on the map view (B). Single-region constraints are defined by polygons and directional constraints are defined by arrows. The transparent color in the interior of the polygons define the type of the constraint: blue means start constraint, red means destination constraint (see Figure 2). The colors on polygon borders and arrows identify distinct queries (there are 3 queries – orange, red, and blue). The orange and red queries are atomic queries, consisting of only atomic temporal and spatial constraints. The blue query Q is a complex query, composed by the union of two atomic queries: a single-region start query Q_1 and a directional query Q_2. In an SQL-like textual notation, Q_1 can be represented as:

SELECT * **FROM** trips
WHERE *trip.pickup_time* $\in [05/01/2011, 05/07/2011]$ **AND** *trip.pickup_location* $\in R_1$

where R_1 denotes the blue region selected in the map. And Q_2:

SELECT * **FROM** trips
WHERE *trip.pickup_time, trip.dropoff_time* $\in [05/01/2011, 05/07/2011]$ **AND** *trip.pickup_location* $\in$ NYCNeighborhood('*Gramercy*') **AND** *trip.dropoff_location* $\in$ NYCRegion('*Times Square*')

where NYCNeighborhood and NYCRegion are functions that given a neighborhood name or region name, respectively, returns the corresponding spatial region.

3.3 Query Evaluation

While easy-to-use, the visual interface leads to an important challenge: a user can issue several large queries and visualizations need to be created for their results at interactive speeds. These queries can be complex. For example, in Figure 3, two queries are represented: the query on the left asks for all pickups in three different neighborhoods on all Sundays in May 2011, and the one on the right explores a different time pattern—all Mondays in May 2011. Users can not only select arbitrary regions, but they can also interactively move polygons

Table 1: Summary of experiments with data storage strategies.

	SQLite	PostgreSQL	TaxiVis Storage
Storage space	100GB	200GB	30GB
Index construction time	52h	13h	28m
1k-query	8s	3s	0.2s
100k-query	85s	24s	2s

around the map, thus generating a series of queries.

To support these queries, we first experimented with traditional database systems, both open source and commercial. In spite of extensions for spatial queries, their query performance is not suitable for interactivity, not to mention the fact that they take a considerable length of time to build the spatial indices. Table 1 shows the performance for SQLite and PostgreSQL with PostGIS, with the former being used for in-memory storage. SQLite took 52 hours just to build the indices for data corresponding to a single year of taxi trips, which as mentioned earlier consists of approximately 170 million trips. Moreover, a single atomic spatio-temporal query could take from seconds to tens of seconds to complete, while complex ones such as those specified by the recurrent time selection widget, can take minutes. Finally, another shortcoming of these database systems is their large memory footprint. In our experiments, SQLite and PostgreSQL used more than 100GB of RAM (in memory setup for SQLite) and 200GB, respectively.

In order to address these issues, we have built a light-weight database variant that allows fast queries on all attributes including spatio-temporal constraints. Our implementation is based on a space-partitioning data structure, the *k*-d tree [8], that treats each taxi trip as a point in a *k*-dimensional space. In our implementation, points are only stored in leaves. Our code takes only 30 minutes to build the indices for the full three years of data and uses only 30GB of disk space. At run-time, the whole data structure, including the data points, are mapped to the system virtual memory, therefore, it may operate in-core or out-of-core adaptively, depending on the available resources. In our tests, compared to the database systems mentioned above, the memory usage is considerably smaller, and queries are significantly faster – they can be evaluated within the bounds required by our interactive system. In Table 1, we summarize the results obtained by our our experiments where 1k-query and 100k-query refer to queries returning approximately 1000 and 100,000 trips, respectively.

4 Automatically Finding Interesting Spatio-Temporal Slices

An important challenge in the exploration of large spatio-temporal data is how to identify *interesting* data slices. While TaxiVis simplifies the exploration of the taxi data, it is not practical to exhaustively examine each data slice. Aggregation can help overcome this problem, but it does so at the cost of occluding small or local patterns in the data [3]. For example, events such as the NYC Five Boro Bike tour shown in the maps of Figure 1, affect a relatively small region in the city over a short period of time, and thus may not be visible when the data is aggregated over time or space.

In more recent work, we designed an automated event-detection algorithm based on techniques from computational topology to help *guide* users towards interesting patterns and data slices [9]. The topological representation of large data sets provides an abstract and compact global view that captures different features and leads to enhanced and easier analysis across applications [19, 38]. The advantages of using topology-based techniques are twofold: topological data structures such as contour trees [7] and Reeb graphs [10], which are used to to identify topological features, can be efficiently computed; and unlike existing approaches, topology-based techniques allow for detection of events that can have arbitrary spatial geometry.

(a) (b)

Figure 4: **(a)** Scalar function for a time step corresponding to 9-10am on May 1, 2011. **(b)** Two of the minima of this function correspond to the the path taken by the NYC Five Boro Bike tour in Lower and Midtown Manhattan.

Detecting events in the taxi data. Event detection is accomplished in two steps. First, a time-varying scalar function is derived from the input data. Here, we assume that the temporal dimension is represented as a set of discrete time steps. If we consider the taxi data, we can define the time-varying scalar function as the density of taxis at each point in NYC over hourly intervals. The domain of this function is represented as a graph. Next, the set of events is computed as topological features. In particular, we consider two types of features: *minimum* and *maximum*. Given a single time step in the taxi density function, a minimum of this function represents a region where the density of taxis is lower than its local neighborhood, implying a relative scarcity of taxis in that region. Such events, when in a busy region, could be due to road blocks. For example, Figure 4(a) shows the scalar function corresponding to a time step when the NYC Five Borough Bike tour from Figure 1 occurred. Two of the minima events are shown in Figure 4(b), which correspond to the path taken by the bike tour in Manhattan. Similarly, a maximum represents a region where the density of taxis is higher than that of its local neighborhood.

Event Index. We implemented a visual interface on top of the TaxiVis infrastructure that allows users to explore events at different time steps. In order to guide users towards potentially interesting events from a possibly large number of events, we designed a rudimentary hash-like indexing scheme that groups similar events across time slices. Let an event E be represented as a pair (R,τ), where R denotes the region associated with an event (a subgraph of the domain graph), and τ is a real number that represents the topological significance of E. The pair above, together with the time of event, is used to identify the appropriate bin, called the *event group*, for E as follows. Given two events $E_1(R_1,\tau_1)$ and $E_2(R_2,\tau_2)$, the *graph distance metric* [6], δ, measures the *geometric similarity* between R_1 and R_2:

$$\delta(E_1,E_2) = 1 - \frac{|R_1 \bigcap R_2|}{\max(|R_1|,|R_2|)},$$

where $R_1 \bigcap R_2$ denotes the maximum common subgraph between R_1 and R_2, and $|R|$ denotes the number of nodes in R. The *topological similarity* between two events is defined as:

$$T(E_1,E_2) = |\tau_1 - \tau_2|$$

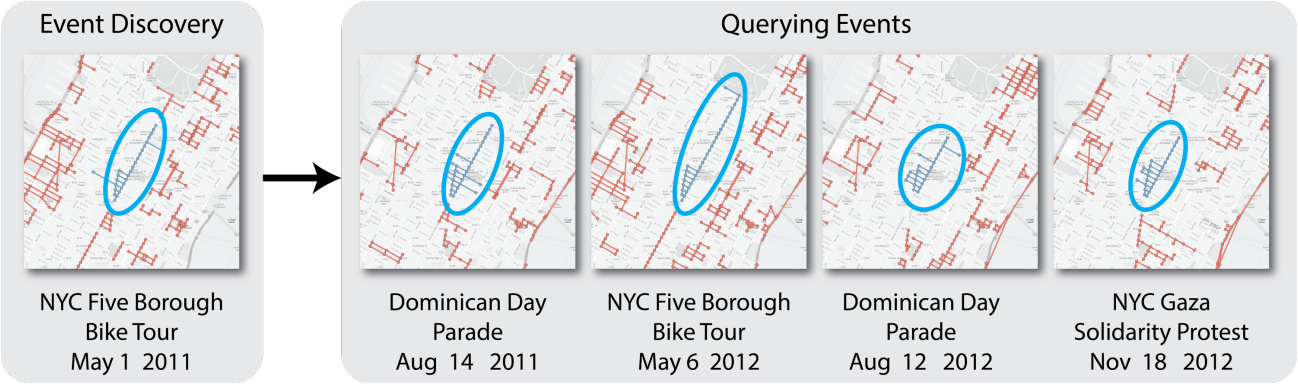

Figure 5: Event-guided exploration of taxi data corresponding to years 2011 and 2012. The user is able to find an event corresponding to the the NYC Five Borough Bike tour that occurred on 1 May 2011 between 8 am and 10 am. Searching for similar events yields the same bike tour that happened in 2012, together with the Dominican day parades for both years. Additionally, we found the Gaza solidarity protest, which was held at the same location.

Two events E_1 and E_2 are in the same event group of the index if

1. $\delta(E_1,E_2) \leq \varepsilon_\delta$ and $T(E_1,E_2) \leq \varepsilon_\tau$, where, ε_δ and ε_τ are user-defined thresholds.
2. E_1 and E_2 occur within the same time period. We used a time period equal to a month in our implementation.

We define two attributes for each event group – range and density. The *range* of an event group is defined as the amount of time between the first and the last event in that group based on their time steps. Its *density* is defined as the number of events of that group that happen per time unit. It measures the time frequency of the events within the group. These two attributes can be used to identify not only periodic events (hourly, daily, and weekly events), but also events with varying frequency (rare events and trends). Thus by exploring properties of event groups through the visual interface, users can effectively identify events of interest. Each event group $\Sigma = \{E_1,E_2,\ldots,E_k\}$, is also associated with a key (R_Σ, τ_Σ), where

$$R_\Sigma = \bigcap_{i\in[1,k]} R_i \quad \text{and} \quad \tau_\Sigma = \sum_{i=1}^{k} \tau_i/k$$

This allows users to search the data for occurrences of a given pattern. This is accomplished by comparing the similarity between the given pattern and the keys associated with the different event groups. Figure 5 illustrates the result of querying the taxi data for events similar to the NYC Five Boro Bike tour.

5 Conclusions

In this paper, we presented an overview of our recent work on techniques to support exploratory analysis and visualization of spatio-temporal urban data. We described TaxiVis, a visual analytics tool and its two main components: a visual query interface that simplifies the process of specifying spatio-temporal queries, and a indexing technique that enables these queries to be evaluated at interactive rates. We also described a topology-based approach to automatic event detection, and how it can help guide users towards interesting time slices. While this work is a step towards scalable and usable visual analytics for spatio-temporal data, there are many open problems we intend to pursue in future work.

Given the growing trend towards transparency, and the large number of open data sets, there is a great opportunity to leverage these data to better understand cities. But this also creates many challenges. Visualization and visual analytics systems have been successfully used to aid users obtain insight from data. But to be effective, visualization systems have to be interactive, requiring sub-second response times [33, 13]. Having been designed for batch queries issued through a text-based or terminal interfaces, existing relational database technologies and business intelligence systems used for OLAP analyses are not suitable backends for these tools [50]. New data management techniques are needed to support interactive visualization [13, 12, 49] Another important challenge comes from the sheer number of data sets available: it is impossible to apply conventional database integration and warehousing techniques where the goal is to establish a single mediated schema. Therefore, we need new methods and tools that help users integrate data on the fly, in a task-oriented manner: as users make discoveries and formulate hypothesis, they should be able to bring in new data that allows them to test these hypotheses. As multiple data sets are integrated, we face additional problems, including the need to support interactive queries that span these data sets, and to visually fuse data sets as different components of a visualization.

Acknowledgments: We thank David Maier for his constructive comments on this manuscript, and the New York City TLC and DoT for providing the data used in this paper and feedback on our results. This work was supported in part by a Google Faculty Award, IBM Faculty Awards, the Moore-Sloan Data Science Environment at NYU, the NYU School of Engineering, the NYU Center for Urban Science and Progress, and NSF awards CNS-1229185 and CI-EN 1405927. Silva has been partially funded by the U.S. Department of Energy (DOE) Office of Biological and Environmental Research (BER).

References

[1] NYC 311. `http://www1.nyc.gov/311`.

[2] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: Queries with bounded errors and bounded response times on very large data. In *Proc. EuroSys*, pages 29–42, 2013.

[3] G. Andrienko, N. Andrienko, C. Hurter, S. Rinzivillo, and S. Wrobel. From movement tracks through events to places: Extracting and characterizing significant places from mobility data. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pages 161–170. IEEE, 2011.

[4] L. Barbosa, K. Pham, C. Silva, M. Vieira, and J. Freire. Structured open urban data: Understanding the landscape. *Big Data*, 2(3), 2014.

[5] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *Proc. IEEE Big Data*, pages 1–8, 2013.

[6] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3):255–259, 1998.

[7] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Comput. Geom. Theory Appl.*, 24(2):75–94, 2003.

[8] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer, 1997.

[9] H. Doraiswamy, N. Ferreira, T. Damoulas, J. Freire, and C. Silva. Using topological analysis to support event-guided exploration in urban data. *IEEE TVCG*, 20(12):2634–2643, 2014.

[10] H. Doraiswamy and V. Natarajan. Computing Reeb graphs as a union of contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):249–262, 2013.

[11] I. Ellen, J. Lacoe, and C. Sharygin. Do foreclosures cause crime? *Journal of Urban Economics*, 74:59–70, 2013.

[12] J.-D. Fekete. Visual analytics infrastructures: From data management to exploration. *IEEE Computer*, 46(7):22–29, 2013.

[13] J.-D. Fekete and C. Silva. Managing data for visual analytics: Opportunities and challenges. *IEEE Data Eng. Bull.*, 35(3):27–36, 2012.

[14] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, 2013.

[15] B. Ferris, K. Watkins, and A. Borning. OneBusAway: Results from providing real-time arrival information for public transit. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1807–1816, New York, USA, 2010. ACM.

[16] A. Feuer. The mayor's geek squad. New York Times, March 24, 2013.

[17] D. Fisher. Incremental, approximate database queries and uncertainty for exploratory visualization. In *Proceedings of the 2011 IEEE Symposium on Large Data Analysis and Visualization*, LDAV '11, pages 73–80. IEEE, 2011.

[18] Freedom of Information Act (FOIA), 2014. http://www.foia.gov.

[19] A. T. Fomenko and T. L. Kunii, editors. *Topological Modeling for Visualization*. Springer Verlag, 1997.

[20] B. Goldstein and L. Dyson. *Beyond Transparency: Open Data and the Future of Civic Innovation*. Code for America Press, San Francisco, USA, 2013.

[21] A. Grossman and A. Sun. MTA swipes show subway trends. http://online.wsj.com, October 19, 2011.

[22] T.-A. Hoang-Vu, V. Been, I. G. Ellen, M. Weselcouch, and J. Freire. Towards understanding real-estate ownership in New York City: Opportunities and challenges. In *Proceedings of the Workshop on Data Science for Macro-Modeling (DSMM)*, 2014. To appear.

[23] J. Höchtl and P. Reichstädter. Linked open data - a means for public sector information management. In *Electronic Government and the Information Systems Perspective*, volume 6866 of *Lecture Notes in Computer Science*, pages 330–343. Springer, Berlin Heidelberg, 2011.

[24] IBM. OpenDX. `http://www.research.ibm.com/dx`.

[25] C. Johnson, H. Pfister, T. Munzner, R. Moorhead, P. Rheingans, and T. Yoo. *NIH-NSF Visualization Research Challenges Report*. IEEE Press, 2006.

[26] U. Jugel, Z. Jerzak, G. Hackenbroich, and V. Markl. M4: A visualization-oriented time series data aggregation. *PVLDB*, 7(10):797–808, 2014.

[27] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed and interactive cube exploration. In *Proc. ICDE*, pages 472–483, 2014.

[28] B. Katz and J. Bradley. *The Metropolitan Revolution: How Cities and Metros Are Fixing Our Broken Politics and Fragile Economy*. Brookings Focus Book. Brookings Institution Press, 2013.

[29] Kitware. Paraview. `http://www.paraview.org`.

[30] Kitware. The visualization toolkit. `http://www.kitware.com`.

[31] Lawrence Livermore National Laboratory. VisIt: Visualize It in Parallel Visualization Application. `https://wci.llnl.gov/codes/visit` [29 March 2008].

[32] L. Lins, J. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE TVCG*, 19(12):2456–2465, Dec 2013.

[33] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2122–2131, Dec 2014.

[34] Z. Liu, B. Jiang, and J. Heer. immens: Real-time visual querying of big data. *Computer Graphics Forum (Proc. EuroVis)*, 32, 2013.

[35] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[36] City of Chicago data portal. `https://data.cityofchicago.org`.

[37] NYC OpenData. `https://nycopendata.socrata.com`.

[38] V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors. *Topological Methods in Data Analysis and Visualization*. Springer, 2010.

[39] D. Peuquet. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers*, 84(3):441–461, 1994.

[40] B. Schaller. Elasticities for taxicab fares and service availability. *Transportation*, 26(3):283–297, 1999.

[41] B. Schaller. A regression model of the number of taxicabs in US cities. *Journal of Public Transportation*, 8(5):63, 2005.

[42] A. E. Schwartz, I. G. Ellen, I. Voicu, and M. H. Schill. The external effects of place-based subsidized housing. *Regional Science and Urban Economics*, 36(6):679 – 707, 2006.

[43] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, H. Wendy, and M. Schraefel. Linked open government data: Lessons from data.gov.uk. *IEEE Intelligent Systems*, 27(3):16–24, 2012.

[44] Taxicab fact book. `http://www.nyc.gov/html/tlc/downloads/pdf/2014_taxicab_fact_book.pdf`, 2014.

[45] J. W. Tukey. *Exploratory Data Analysis*. Pearson, 1977.

[46] UN 2012 world urbanization prospects: The 2011 revision highlights. `http://esa.un.org/unup/pdf/WUP2011_Highlights.pdf`, 2012. Page accessed Jan 2014.

[47] C. Upson et al. The application visualization system: A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4):30–42, 1989.

[48] H. Wickham. Bin-summarise-smooth: a framework for visualising large data. Technical report, had.co.nz, 2013.

[49] E. Wu, L. Battle, and S. R. Madden. The case for data visualization management systems. *PVLDB*, 7(10):903–906, 2014.

[50] K. Zoumpatianos, S. Idreos, and T. Palpanas. Indexing for interactive exploration of big data series. In *Proc. SIGMOD*, pages 1555–1566, 2014.