Data Flow Analysis and Visualization for Spatiotemporal Statistical Data without Trajectory Information

Seokyeon Kim, Seongmin Jeong, Insoo Woo, Yun Jang, Member, IEEE, Ross Maciejewski, Member, IEEE, and David S. Ebert, Fellow, IEEE

Abstract—Geographic visualization research has focused on a variety of techniques to represent and explore spatiotemporal data. The goal of those techniques is to enable users to explore events and interactions over space and time in order to facilitate the discovery of patterns, anomalies and relationships within the data. However, it is difficult to extract and visualize data flow patterns over time for non-directional statistical data without trajectory information. In this work, we develop a novel flow analysis technique to extract, represent, and analyze flow maps of non-directional spatiotemporal data unaccompanied by trajectory information. We estimate a continuous distribution of these events over space and time, and extract flow fields for spatial and temporal changes utilizing a gravity model. Then, we effectively visualize the spatiotemporal patterns in the data by employing flow visualization techniques. The user is presented with temporal trends of geo-referenced discrete events on a map. As such, overall spatiotemporal data flow patterns help users analyze geo-referenced temporal events, such as disease outbreaks, crime patterns, etc. To validate our model, we discard the trajectory information in an origin-destination dataset and apply our technique to the data and compare the derived trajectories and the original. Finally, we present spatiotemporal trend analysis for statistical datasets including twitter data, maritime search and rescue events, and syndromic surveillance.

Index Terms—Spatiotemporal Data Visualization, Kernel Density Estimation, Flow Map, Gravity Model.

1 INTRODUCTION

In cartography, a thematic map is a special-purpose map designed to illustrate particular features or concepts [8], [43] within a dataset. One common theme is the representation of geographical movements of people, ideas, money, energy, or material. Usually movement tables are given and the tables are visualized utilizing lines, arrows, or streaklines (e.g., [5], [27], [39], [48]). However, these movement tables do not exist for much data, even though we know that individuals in the dataset do move. Furthermore, many spatiotemporal event datasets are collected as the movement is already embedded into the datasets; unfortunately, the movement is never explicitly defined in the data.

For example, emergency room records are routinely collected, and such records may contain an underlying notion of disease spread. However, the records themselves have no explicit definition of movements. Instead, each record has only a patient’s address, time of visit, and visit reason. One can imagine that looking at some measure of movement within the spatiotemporal events could provide analysts with insights into various spread patterns. Such spread patterns are not limited only to emergency room events. We can expand this idea to any spatiotemporal event data, such as criminal incident reports, economics, and social trends. In order to find movements within the statistical data, both spatial and temporal patterns need to be explored during the analysis phase. However, it is not easy to extract target movements from statistical data since there are many complicating factors. In previous work, one natural way to analyze such spatiotemporal data is to plot the data on a map and then provide animation control or small multiples views to visualize each time step of the data. This approach provides analysts with insight into the spatial distributions of the data as well as patterns and correlations between these distributions over time. However, scrolling through time steps requires analysts to remember what spatial distributions have occurred at different time steps, and it is not easy to compare one visualization with another visualization on the screen. Moreover, it is hard to extract movements of the data explicitly by direct comparison of two visualizations. There also have been several illustrative and abstract visualization techniques but only direct movement data have been targeted without any flow extraction. In order to overcome these issues, we propose a flow extraction model.

Figure 1 illustrates the concept of potential event flow extraction. Spatiotemporal data is visualized on the map in geographical space in accordance with the time. Events are represented as red heatmaps estimated from the raw data. When the heatmaps are given with two time steps, $t$ and $t + 1$, as in Figure 1 (a), the event flow of the data can be indicated as the straight arrow from the left to the right. In the same way, when the heatmaps are provided over the time steps, $t - 1$ to $t + 1$, the event flow can simply be extracted as the round shaped arrow as shown in Figure 1 (b). Both Figure 1 (a) and (b) are simple cases that have no overlap between events, and the event flows can easily be extracted. However, when applying the actual data, the heatmaps are too complicated to analyze. Figure 1 (c) presents actual heatmaps using real data over the time steps, $t - 2$ to $t + 2$. 
In this work, we present a novel technique for extracting movement information from event-based data sources to create geographic flow maps. As shown in Figure 2, our technique approximates the underlying data distribution over time through the application of a kernel density estimator. This provides us with a continuous functional representation of the data. We, then apply a gravity model to extract flow maps of non-directional statistical data. The mass in the gravity model is obtained from the functional density distribution and the gravity vectors are computed for the flow directions. In this way, we can explore the spread patterns of spatiotemporal event data without any trajectory information, such as disease, crime, social trend. In order to visualize the flow efficiently, we apply line integral convolution (LIC) [7] with animated directional glyphs on the map and oriented line integral convolution (OLIC) [49]. We evaluate our technique using GPS trajectory data, Twitter data, maritime search and rescue events, and syndromic surveillance data. Note that we utilize the term movement and flow interchangeably throughout this work.

Our technique has several benefits for analyzing spatiotemporal data. Since the trajectory information is not easily collected and requires massive amounts of storage for further analysis, we do not make use of the trajectory information in our system. Instead, our technique provides flow maps of event-based statistical datasets. The flow maps are estimated procedurally on the fly and users are able to interactively adjust the parameters in our flow map generation algorithm for more intuitive flow map extractions. The flow maps can be used to understand potential movement paths over time within the statistical datasets. Since certain types of statistical datasets imply movement, it is more advantageous to analyze the data with the movement information, which is the flow map in this work, as compared to conventional visualizations of statistical data distributions. Note that this shows flow trends but does not track movements of the entities creating them.

The major contributions of our paper are as follows:

- A continuous spatiotemporal functional representation for non-directional discrete event-based data using kernel density estimation without trajectory information
- Procedural continuous flow map extraction from spatiotemporal data using the functional representation
- Design of a 3D gravity model to extract potential flow pattern in statistical spatiotemporal data
- Interactive visualization of the flow maps using animated glyphs (particles), LIC, and OLIC

We first review previous work in Section 2 and present an overview of our flow analysis system in Section 3. We describe functional representations for spatiotemporal statistical datasets in Section 4. In Section 5 we present our flow map extraction algorithm, a gravity model, and provide flow map visualization techniques in Section 6. Then, we present results generated by our system in Section 7 and discuss our system in Section 8. Finally, the conclusion and future directions are discussed in Section 9.

2 RELATED WORK

As mentioned earlier, our system generates data flow maps within non-directional spatiotemporal data by employing existing vector field visualization techniques. This section covers relevant topics in spatiotemporal data visualization and vector field visualization.

2.1 Spatiotemporal data analysis

In recent years, many geo-visualization techniques have been developed for exploratory data analysis, and these techniques have been used to inspire creative thinking and provide new insights into previously unknown characteristics of the original data [38]. Some of the techniques focus on data with spatial and temporal information. Tominski et al. [46] use three-dimensional icons to visualize spatiotemporal data on a map which is subdivided based on administrative subdivisions of geographical regions (e.g., federal state). Temporal trends are analyzed based on the administrative subdivisions, encoded in 3D models such as pencils and helixes and displayed on each subdivision. However, this method does not provide temporal trends across subdivisions. Maciejewski et al. [34] propose kernel density estimate heatmaps linked with temporal analysis views to detect hotspots for spatiotemporal data. This work allows the user to detect hotspots of interesting events and analyze a temporal trend by selecting a hotspot on the kernel density estimate heatmap. There is more previous work providing multiple views (e.g., parallel coordinates, matrix and scatter plots) linked with spatial and temporal data views in order to leverage the spatiotemporal data analysis [13], [18].

Another technique to represent spatiotemporal data is the spacetime cube [15], which is constructed by aggregating the event information over the two-dimensional geographical space and additional third-dimension to denote time. The spacetime cube shows its usefulness in analyzing agent-based events with spatial and temporal information in geovisualization [21]. The concept of the spacetime cube is utilized to visualize temporal behaviors of events, such as storytelling and human eye movement [9], [29]. Other types of events, such as health care data, are also applied to the spacetime cube presentation by displaying meaningful glyphs in the cube. Gatalsky et al. [11] visualize a catalog of earthquake events in the spacetime cube using different-size circles in different colors. Kraak [22] applies the spacetime cube to visualize the distribution and propagation of epidemics with additional data analysis views (e.g., parallel coordinates). However, it is still hard for the user to recognize the location and time of a certain event in the spacetime cube. Moreover, it becomes more difficult to analyze the temporal trends for frequently occurring events. Nakaya and Yano [37] use spacetime kernel density estimation on spatiotemporal events and visualize the estimated density results using volume rendering techniques.
Flow maps [5], [12], [14], [32], [39] have been considered an effective means of visualizing spatial interactions (e.g., migration flow) when datasets imply flow between spatially different regions. Such data can form a location-to-location network. However, some event types, such as crime counts, and patient counts, are not applicable for flow maps since there is no network or trajectory information for the origins and destinations. Waldo Tobler [44], [45] presents migration map using the N-squared table of geographical interactions. His insight originates from Tobler’s First Law of Geography: “Everything is related to everything else, but near things are more related than distant things.” Although the N-squared table is useful, Tobler’s work has a lack of insight to time varying data. Andrienko et al. [2] propose spatial generalization and aggregation of massive time varying movement data using glyph shape designed by Tobler. Another approach to generate the flow maps from the statistical datasets is to use a gravity model [1], [3], [4], [19], [28], [30], [40], [47]. A gravity model has been used in many social science applications to explain certain behaviors that are similar to gravitational interaction in Newton’s Law of Gravitation. Recently the gravity model is widely utilized in various research areas including spatial clustering [16], navigation [17], international migration [19], [28], [40], international trade [4], and disease spread [3], [30], [47]. Especially, Alberto Salvo [41] highlights how collusive behavior can magnify the effects of distance with two illustrations, such as the supply of two firms to two northeastern markets in 1996 and the supply of three firms to two southeastern markets in 1999. Kincses and Tóth [20] propose accessibility analysis using gravity modeling, bi-dimensional regression calculations, and GIS visualization. Our work is motivated by the previous gravity model research and we simplify the gravity model to extract flow maps from spatiotemporal statistical data without any trajectory information.

2.2 Vector field visualization

Vector field (flow) visualization has been a well studied topic, and is used to depict flow patterns and magnitudes in simulated or measured vector fields. There have been various approaches [26] to vector field visualization which are classified into direct flow visualization, texture-based flow visualization, geometric flow visualization, and feature-based flow visualization. One well-known texture-based flow visualization technique is Line Integral Convolution (LIC) [7]. It has been shown that LIC fits well in the visualization of 2D flow patterns [25], [26]. Moreover, in order to reveal the directional vector field, Wegenkittl et al. [49] have proposed oriented LIC (OLIC). As geometric flow visualizations, streamlines have been shown to provide a good overview of the flow field [36]. Particle advection techniques provide animated flows by mapping glyph images on particles and rendering geometric shapes of particles [23]. It has been a challenging issue to determine the best flow visualization technique that is well matched to a specific application and domain. As an effort to understand the effectiveness of various flow visualization techniques, researchers performed evaluation studies [10], [24], [31]. Laidlaw et al. [24] describe strengths and weaknesses for six visualization methods including arrow icons, integral curves, wedges, and line-integral convolution (LIC) for 2D vector data through a quantitative evaluation study. Their results show that both expert and novice users better performed given tasks when they used methods with the sign of vectors within the vector field as well as visual representations of integral curves and locations of critical points. Liu et al. [31] evaluate geometry-based and texture-based flow visualization methods of grid-based arrows, evenly spaced streamlines, and LIC variants along with a color wheel and color map for 2D flow visualization. By conducting a user study, they find that a texture-based representation with enhanced LIC creates intuitive perception of the flow, and a geometry-based representation with streamlines may generate visual interpolation as such maximizing the perception of flows. They also describe that a priori familiarity of users with the techniques may affect understanding flows, while mentioning that many participants in their evaluation study are familiar with arrows and streamlines rather than LIC. Based on the results and recommendations from previous research, in this work we utilize particle advection with glyph images.

3 System overview

Most spatiotemporal visual analysis systems utilize a combination of geospatial and temporal visualization. Starting from an overview of geospatial data at a certain time, the user takes a detailed look at temporal information in a location or area of interest through the use of linked views. Such a process has
Figure 3. Data distributions are shown as (a) a heatmap in the current, \( t_0 \), and (b) a heatmap in the future, \( t_0 + 1 \). (c) The flow map from \( t_0 \) to \( t_0 + 1 \) is extracted using the gravity model. The area (I) shows diverging flows and the area (II) presents converging flows when the density decreases and increases, respectively. The flow path is formed along the density changes over time as shown in the green arrow (III). Note that figure (c) is obtained with the kernel \( W = 30, T = 1 \). An example of a 3D gravity kernel for the gravity model is illustrated in (d), where kernel size is \( W = 2 \) and \( T = 1 \) in Equation 5. The \((p, q, r)\) values are shown as 3-tuples in (d). The red arrow is the flow for the green heat map.

proven to be an effective means of exploring spatiotemporal data. However, such exploration needs many snapshots of geospatial data over time and requires the user to switch from a spatial view to a temporal view and vice versa repeatedly.

In order to reduce such cumbersome tasks, we design a novel analysis system to investigate non-directional event based spatiotemporal data. Our flow analysis approach provides spatial movement patterns as well as temporal trends for these datasets. Figure 2 illustrates an overview of our flow analysis. First, events of interest are aggregated based on a user specified time duration (e.g., daily, weekly, monthly). The aggregated discrete data distribution is converted into our functional representations using kernel density estimations. The flow maps are then generated by evaluating a set of our functional representations using a gravity model. In this way, we can extract the flow maps from statistical datasets, and then the flow maps are visualized by adopting vector field visualization techniques. In order to provide a proper spatiotemporal analysis system, we need to overcome the following challenges:

- How do we handle spatial and temporal dimensions together in the functional representations?
- How do we extract flow maps (vector fields) from the scalar data distributions?
- How can we visualize the flow maps for spatiotemporal analysis?

Our solutions to these challenges are introduced in Section 4, Section 5, and Section 6, respectively. Sample visualizations from our flow map analysis system are shown in Figure 3. In the figure, two heatmaps for the current, \( t_0 \), in (a) and the future, \( t_0 + 1 \), in (b) are presented. The flow map between these two heatmaps is visualized in (c). The flow map has three different patterns depending on the data distribution. Diverging and converging flows of Figure 3 are found in the area (I) and (II). The converging pattern (II) is seen in the area where the density increases, whereas the diverging pattern (I) is found when the density decreases. Moreover, a flow path (III) is extracted along the density changes over time as seen in Figure 3 (the green arrow in (c)).

In order to illustrate our algorithm more clearly, we present two test cases under ideal conditions in Figure 4. There are two event locations in (a). The density of the upper left event decreases, whereas, the density of the lower right event increases. This demonstrated the expected diverging and converging patterns, respectively. These flow patterns are obtained by our approach as shown in (a). Figure 4 (b) shows another simple test case where events are moving toward the lower left, as illustrated in the resulting visualization.

4 FUNCTIONAL REPRESENTATION OF SPATIOTEMPORAL DATA

Spatiotemporal data consists of discrete events at geolocations over time. Spatial distribution of the events at each time step can be easily represented using functions as presented in many previous studies. The distribution is encoded with compactly supported kernels and the encoded distribution provides a heatmap of data values. In order to obtain spatiotemporal distributions, it is necessary to combine two different dimensions: location and time. However, it is not intuitively easy to extend this representation...
Figure 5. Parameter comparisons for \( a_0 \) and \( a_1 \) in Equation 5. (a) presents three heat map visualizations from \( t_0 - 1 \) to \( t_0 + 1 \). The flow maps are extracted by varying the values, \( a_0 \) and \( a_1 \), in (b), (c), and (d). When \( a_0 \) increases, the flow patterns are observed as similar to the potential flow at \( t_0 \). On the other hand, when \( a_1 \) increases, the flow patterns are extracted as more temporal patterns of the neighboring steps from \( t_0 - 1 \) to \( t_0 + 1 \) and this tends to ignore the events at \( t_0 \).

The kernel bandwidth can be fixed or varied depending on the method employed. The kernel bandwidth influences the magnitude of the kernel, i.e., kernels with large bandwidths have smaller heights. We use a variable bandwidth for the the kernel, \( K_s \), proposed by Silverman [42]. The bandwidth \( (h_{x,i}) \) of the kernel \( K_s \) placed on the point \((x_i, y_i)\) is proportional to the distance from the \( i \)th sample to the \( k \)th nearest neighbor.

5 Flow Map Extraction

Flow maps in cartography are defined by Phan et al. [39] as “a mix of maps and flow charts, that show the movement of objects from one location to another, such as the number of people in a migration, the amount of goods being traded, or the number of packets in a network”. Usually flow maps are obtained from datasets with direction or trajectory information. However, most statistical datasets do not have the directional information. In this Section, we propose a gravity-based model to extract the potential flow map.

5.1 Gravity model

The generic form of the gravity model between spatial locations, \( i, j \), is presented as follows:

\[
g_{(i,j)} = \frac{(m_i^{a_0} \cdot m_j^{a_1})}{d_{ij}^2},
\]

where \( m_i \) and \( m_j \) are the masses at \( i \) and \( j \), \( d_{ij} \) is the distance between \( i \) and \( j \), \( a_0 \) and \( a_1 \) are the control parameters affecting masses. In many fields, researchers have estimated these \( a_0 \) and \( a_1 \) statistically using many data variables. For example, Karemera et
The spatio-temporal flow map extraction model using the gravity model produces only local flows. Figure 6 (d) shows the flow map visualization with the gravity-based flow extraction. Al. [19] present a gravity model analysis of international migration to North America and they use the number of migrants, distance, population, income, inflation, unemployment, language, etc. as data variables to estimate influential parameters in their model. In this work, we have statistical datasets that consist of geolocations and events over time; therefore, we modify the gravity model to estimate the flows within the statistical data. We could use other estimation parameters combined with this in future work.

### 5.2 Gravity-based flow extraction model

In order to estimate the event flows using the gravity model, we replace the mass in Equation 3, \( m \), with \( \hat{f}_{2D}(x,y) \) as presented in the following.

\[
g_{ij}(t) = \frac{\hat{f}_{2D}(x_i,y_j)^{a_0} \cdot \hat{f}_{2D}(x_j,y_i)^{a_1}}{d_{ij}^2} \tag{4}\]

The spatio-temporal flow map extraction model using the gravity model is presented as follows:

\[
F_{\text{Map}}_{3D}(x,y,t) = \sum_{p=W}^{W} \sum_{q=W}^{W} \sum_{r=-T}^{T} (p,q,r) \cdot \frac{\hat{f}_{2D}(x,y)[p,r]^{a_0} \cdot \hat{f}_{2D}(x,p,y)[r]^{a_1}}{d_{ij}^2}, \tag{5}\]

where \( W \) is the kernel size in the spatial axes, \((x,y)\), \( T \) is the kernel size along the time axis, \((t)\), and \( d_{ij} \) is the euclidean distance between \((x,y,t)\) and \((x_p,y_q,t_r)\). The temporal trend is determined by the multiple adjacent distributions along the time axis. Note that the multiplication of \((p,q,r)\) results in the directional information in the flow map. Figure 3 (d) illustrates an example of the gravity kernel when \( W = 2 \) and \( T = 1 \) in Equation 5. In that case, the x-axis and y-axis range is \(-2\) to \(2\) and the t-axis range is \(-1\) to \(1\). Note that the discrete kernel values are represented in the 3-tuples.

The influences of the parameters, \( a_0 \) and \( a_1 \), are compared in Figure 5. Three heatmaps for the consecutive time steps are shown in (a). The flow maps are extracted about \( t_0 \) with the kernel (\( W = 80, T = 1 \)) by varying the values, \( a_0 \) and \( a_1 \), in Figure 5 (b), (c), and (d). When \( a_0 \) increases, the flow patterns are similar to the potential flow at \( t_0 \), whereas, Figure 5 (d) presents the flow patterns only between \( t_0 - 1 \) and \( t_0 + 1 \) and ignore the event density at \( t_0 \). Since the effects of the parameters, \( a_0 \) and \( a_1 \), are dominant, we use \((1,1)\) for \( a_0 \) and \( a_1 \) in order to extract the global flow patterns including all time steps within the kernel ranges. However, one can extract the flow map by adjusting \( a_0 \) and \( a_1 \) to stress certain time steps.

As a different flow map extraction model, the flow can be simply obtained by the difference between heatmaps in two different time steps. The difference-based flow extraction model is presented as follows:

\[
F_{\text{Map}}_{\text{diff}}(x,y)|_{t_0, t_1} = (\hat{f}_{2D}(x,y)|_{t_1}) - (\hat{f}_{2D}(x,y)|_{t_0}) \tag{6}\]

However, \( F_{\text{Map}}_{\text{diff}}(x,y) \) extracts only local flow patterns that do not represent the global flows. Figure 6 shows the comparison between difference-based flow extraction and gravity-based flow extraction. Figure 6 (a) and (b) are the heatmaps at the current time, \( t_0 \), and the future, \( t_1 \). The major changes between Figure 6 (a) and (b) are found in area (I) and area (II). Figure 6 (c) presents the flow map visualization with the difference-based flow extraction. As shown in Figure 6 (c), several converging flows toward the centers of high heatmap values in the future, \( t_1 \). However, it is not possible to extract the global movements since the difference-based model produces only local flows. Figure 6 (d) shows the flow map using the gravity-based flow extraction. The global flow patterns are extracted as the curve from Figure 6 area (I) to area (II).
Table 1
Comparison of CPU and GPU computing time

<table>
<thead>
<tr>
<th>computing type</th>
<th>model</th>
<th>total points</th>
<th>number of timesteps</th>
<th>gravity parameters W</th>
<th>T</th>
<th>$T_{\text{total}}$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>$FMap_{diff}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>1,353</td>
</tr>
<tr>
<td></td>
<td>$FMap_{diff}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>34,742</td>
</tr>
<tr>
<td></td>
<td>$FM_{Map_{3D}}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>$FM_{Map_{3D}}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>34,606</td>
</tr>
<tr>
<td></td>
<td>$FM_{Map_{3D}}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>5,224</td>
</tr>
<tr>
<td></td>
<td>$FM_{Map_{3D}}$</td>
<td>3,056</td>
<td>31</td>
<td>-</td>
<td>-</td>
<td>34,606</td>
</tr>
</tbody>
</table>

Table 2
Performance of GPU computing time for different datasets

<table>
<thead>
<tr>
<th>data</th>
<th>total points</th>
<th>number of timesteps</th>
<th>gravity parameters W</th>
<th>T</th>
<th>$T_{\text{total}}$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Bowl</td>
<td>4,659</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>34,742</td>
</tr>
<tr>
<td>Boston Marathon</td>
<td>5,224</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>34,742</td>
</tr>
<tr>
<td>SAR</td>
<td>23,416</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>34,742</td>
</tr>
<tr>
<td>syndromic</td>
<td>693,878</td>
<td>24</td>
<td>-</td>
<td>-</td>
<td>34,742</td>
</tr>
</tbody>
</table>

6 Visualization system of spatiotemporal data

In order to analyze spatiotemporal trends of statistical data, the flow maps are extracted as introduced in Section 5. In this work, the flow maps are overlaid on a map to reveal movement patterns within a geographical space and the flow fields are visualized with vector field visualization techniques. Figure 7 illustrates the visualization pipeline for the flow map. The computation for the algorithm is performed and, then, the system transfers the vector fields to either a stand-alone client software or a web-based software for the vector field visualization. Our system provides three different types of visualization including OLIC, arrow glyph, and particle visualization similar to LIC. We present the flow map visualization techniques and details of the system implementation in this section.

6.1 Flow map visualization

In flow visualization using the particle advection, initial particle seeding is one of the challenging issues to obtain the best illustration of flow patterns within a vector field [48]. Some previous work uses flow feature points from fluid simulation [51]. However, such approaches are not applicable to our flow maps computed in Section 5 since our input data is not physics-based simulation data but a set of spatiotemporal discrete events. Moreover, our particle advection scheme is different from typical particle advection methods in that we deal with the transition of areas with high density areas or high density changes and consider two-dimensional projection of the patterns in the flow maps onto a geographical map. Many statistical datasets have more importance in the areas with high density changes since they may have meaningful trends to be analyzed. Such areas are usually located around hotspots. We initially place the particles in the region of interest, such as high density or high velocity areas, using the importance-driven method proposed by Bürger et al. [6]. We randomly select one high density area and place a new particle in a random position $(x, y)$ within the selected area. The new particle is advected along the flow map and dies when the life time is over. The particles are visualized as animated directional glyphs in our system. Similarly, OLIC are applied only to non-zero vector fields, which are extracted in the areas whose density values are not zero. Therefore, we can project these visualization results onto a map where actual events occur.

6.2 Web-based flow map visualization

Our system also provides a web-based flow map visualization. When the system generates a flow map using our model, the vector fields are transmitted to not only the local client software but also the web server. The web-based flow map visualization system consists of JavaScript and several APIs, such as D3.js, Backbone.js, When.js. The web server visualizes the 3D globe on the web according to the vector fields with the D3 projections. The server, then, attaches minimum geographical information including roads, country boundaries, and lakes, using the TopoJSON. The web-based flow map visualization provides animated particles and the color of a particle varies accordingly as the particle ages. If the target vector field area is too small to be visible enough, our system allows a user to apply an additional map on the different map layer located between the globe layer and the particle layer.

6.3 System implementation

In order to build our system, we utilize hardware accelerated computations and visualizations. The functional representations in Section 4 and the flow map extractions in Section 5 are implemented using NVIDIA CUDA. Once the functional representations are obtained in the system, we calculate the flow maps in CUDA again. Thereafter, the flow maps are transferred to shaders for visualizations. We use a multi-pass technique to obtain several layers including map rendering, heat map rendering, particle rendering, point sprite rendering, LIC and OLIC rendering, etc., and the layers are blended for the final results. For the vector field visualization, a geometry shader is used for the particle rendering and point sprite rendering in order to create glyphs. Table 1 and Table 2 show the computing performance according to our model parameters. Data load and histogram generation time, $T_{\text{load}}$, KDE preparation time, $T_{\text{pre}}$, computing time for KDE, $T_{\text{diff}}$, computing time for the gravity model, $T_{\text{grav}}$, and data transfer time from GPU to CPU, $T_{\text{tran}}$, are compared, respectively. In Table 1, GPU performance is about 30 times faster than CPU $FMap_{diff}$. Note that computing $FMap_{3D}$ on CPU is not possible in our experiment. For the $FMap_{3D}$ computation, as W increases, the computing
Figure 8. The flow map visualizations of GPS trajectory data. (a)-(e) show the data distributions starting from (a) and moving toward (e). (f) presents the flow map extracted from our gravity model with three time steps. (g) is the flow map with five time steps. The difference between (f) and (g) is found at intersections marked in the rectangular boxes. The flow map is visualized with OLIC.

time becomes longer due to the increase of the number of cells in the gravity kernel. Table 2 shows the performance of GPU computing for different datasets that are used in Section 7. We have tested our flow analysis system on commodity workstations, whose specifications include an Intel i7 CPU, 16GB CPU memory, and GeForce GTX 660 with 2GB GPU memory.

7 DATA EXPLORATION

Our system extracts the flow maps and presents the spatiotemporal trends on a geographical map. In order to evaluate our system, we use various datasets and extract different flow maps. We present the flow map visualizations and the discussion of the flow maps in this section. In Section 7.1, we validate our technique with a GPS trajectory data as a systematic evaluation. However, in Section 7.2, Section 7.3, and Section 7.4, we apply the technique to statistical spatiotemporal data including twitter data, maritime search and rescue events, and syndromic surveillance. Since we are focused on data flow pattern analysis for non-directional spatiotemporal statistical data, our system utilizes discrete event data without trajectory information, such as twitter ID or movements.

7.1 Origin-destination data analysis

In order to validate our technique, we present flow map visualizations with origin-destination data containing actual flows. We use GPS trajectory data and we apply our technique to the dataset after discarding the trajectory information in the data. We, then, apply our flow map extraction model and compare extracted patterns with the original trajectory information. The GPS trajectory data were collected by 182 GPS users in GeoLife project of Microsoft Research Asia from April 2007 to August 2012 [52]. Figure 8 (a) - (e) present the data visualizations with every 5 minutes GPS trajectory data with the heatmaps. The flow maps, (f) and (g), are visualized with OLIC for the flow vectors extracted by our gravity model. In order to make the data analysis fair, we aggregate the data points within certain time range and generate new data with multiple time steps. (a)-(e) are the distributions of five consecutive time steps and each time step contains all data points within the time range. Then we apply our gravity model to extract the flow map. The flow map in (f) is generated with three time steps and one in (g) is extracted with five time steps. Both show the data flow directions with OLIC technique. The difference is found at the intersections marked in the rectangular boxes. The flows at the intersections in (f) are following the street shapes, whereas, the flow directions at the intersections in (g) are diagonal. If we compute the flow map with many time steps at the same time, the vector fields are representing the data flow directions for all time steps instead of just a single time step, which tells the effect of the gravity kernel size along the temporal axis, $T$, in Equation 5. Note that the individual OLIC visualizations over several time steps are composed within our system to reveal the flow map for entire time steps. In this example, we can demonstrate that our flow map extraction technique reproduces the actual trajectory directions and the flow maps can be abstracted for multiple time steps depending on the gravity kernel sizes.

7.2 Twitter data analysis

Recently, social media services, e.g, Twitter, offers a freely accessible database of user-generated reports. As many people use GPS enabled mobile communication devices, these reports are able to capture important local events observed by an active and ubiquitous community. We have collected and analyzed Tweet messages to investigate the message flows in our system. We analyzed two different cases, 2015 Super Bowl and Boston Marathon bombing in 2013. Note that we do not utilize any trajectory information from the social media data.

The Super Bowl was held in Glendale, Arizona, on February 1st, 2015. Before investigating this case, we expected that many fans came to the stadium to watch the game and they might have been using their mobile phones to broadcast their status during the game day through Twitter. As expected, a lot of Tweets were generated during the day, and we decided to explore movements of the fans on the day. We aggregated the Twitter data by hour from 12:00 to 20:00 and extracted the flow map every hour. There are
Figure 9. The flow maps during Super Bowl in 2015. Three hot spots are marked in the information; the stadium, malls, and remote parking area. The tweet messages related with the game are analyzed between 12:00 and 20:00. The game started at 16:30 and the converging flow patterns were found around the stadium from 12:00 to 16:00. Then, the flow patterns toward the remote parking area from 16:00 to 17:00 were discovered due to the lack of parking spots around the stadium. After the game, the flow patterns were detected as people moved toward the malls.

Our second twitter study focused on The Boston Marathon bombings and subsequent related shooting incidents that began on April 15, 2013 when two pressure cooker bombs exploded during the Boston Marathon at 14:49, killing 3 people and injuring an estimated 264 others [50]. We started to analyze Twitter messages from 13:00 to 17:00 as we assumed that there were random patterns before the marathon athletes showed up and eventually people gathered together along the marathon course. As seen in Figure 10 (a), (b), and (c), there were mixed patterns, such as
7.3 Coast guard search and rescue data analysis

Our next application is the Search And Rescue (SAR) data that is collected by all U.S. Coast Guard stations [35]. There are two different types of the data: response cases (call for action) and response sorties (resources deployed to respond to the call, e.g., a boat or an aircraft). With this data, the Coast Guard is particularly interested in determining the spatial and temporal distribution of response cases and their associated sorties (a boat or an aircraft allocated to respond to an incident) for all SAR operations conducted in the Great Lakes. The SAR data from 2004 to 2011 is used for this study. Figure 11 (a), (b), and (c) for winter, spring, and summer, respectively, show the density distributions of the SAR cases in Lake Michigan, Lake Huron, and Lake Erie. As seen in the figures, more SAR cases happen during summer. Moreover, more SAR cases occur near the small cities including Muskegon, Ludington, Traverse City, Sarnia, during summer, whereas, most SAR cases are found near the big cities during winter. This observation is found as the flow map in (d). This implies that the safety or coast guards should be allocated more near the small cities in summer along the flow patterns.

7.4 Synthetic syndromic data analysis

Another example is syndromic surveillance, which focuses on detection of adverse health events. We use a synthetic syndromic surveillance data generated by Maciejewski et al. [33], consisting of a series of outbreaks over time moving from Indianapolis to Northern Indiana. Figure 12 illustrates how the outbreaks change over time as heatmaps in (a)-(c). The outbreaks occurred around random and gathering movements at 12:00, 13:00, and 14:00. Then the bombs exploded at 14:49PM and right after that, people escaped from the bombing sites, which are marked in Figure 10 (d). The flow patterns showed that all people moved away and our flow map model caught these patterns. After two hours, people slowly came to the bombing sites to investigate or watch the scene. This is shown in Figure 10 (f).

random and gathering movements at 12:00, 13:00, and 14:00. Then the bombs exploded at 14:49PM and right after that, people escaped from the bombing sites, which are marked in Figure 10 (d). The flow patterns showed that all people moved away and our flow map model caught these patterns. After two hours, people slowly came to the bombing sites to investigate or watch the scene. This is shown in Figure 10 (f).

random and gathering movements at 12:00, 13:00, and 14:00. Then the bombs exploded at 14:49PM and right after that, people escaped from the bombing sites, which are marked in Figure 10 (d). The flow patterns showed that all people moved away and our flow map model caught these patterns. After two hours, people slowly came to the bombing sites to investigate or watch the scene. This is shown in Figure 10 (f).
Indianapolis on July 26, 2006 and propagated to near areas and Northern Indiana areas. We compute the flow map of the data by placing the center of gravity on (b) with the kernel (W=50, T=3), which indicates that our system computes the flow map for the seven time steps. Flow paths from Indianapolis to North Indiana along local highways and state border between Indiana and Illinois are extracted as the disease spreading in Figure 12 (d) and (e), which are not easily imagined only from the heatmaps. Small flow patterns are also visible toward some small cities in the suburb of Indianapolis. In addition, there are major flow paths toward Fort Wayne in the northeast Indiana, which is actually a future (t0 + 1) event at the time (t0) in (b). The major flow is formed near Fort Wayne because the amount of event change is large even when the density near Fort Wayne is low. Notice that the flow patterns from high density to high density do not appear since there is no event change. In addition, the pattern from high density to low density appears as diverging flow, whereas, the pattern from low density to high density are shown as converging flow.

8 Discussion
We have developed an event data flow analysis and visualization system with novel flow map extraction algorithms that can be applied to any spatial statistical data without trajectory information. In this section, we discuss requirements and limitations of our approach.

8.1 Statistical data
Our system requires exact geographical information, however, most statistical data contains district level location, such as state and city, without latitude and longitude information. Since this keeps our system from building accurate functional representations, it is not easy to extract flow maps due to the ambiguity of the event locations. Also if there is nonuniform event distribution along the time axis, it is difficult to determine the time aggregation level for the data flow analysis. Another requirement for the data in our system is that there should be enough time steps to extract the flow map. Since the gravity model analyzes the past, current, and future time steps, our system needs at least three different time steps to extract the flow map. Due to this, it is not possible to obtain the flow maps for the first and last time steps.

8.2 Flow patterns
Although our algorithm extracts the flow map from the statistics data without any trajectory information, there are a few cases that our system is not able to handle properly for the flow map extraction. The special cases are discussed in this section.

The first case is that an event moves toward one direction and then comes back to the original location as seen in Figure 13. The density distributions are shown on the left from (t0 − 2) to (t0 + 2). As seen in the distributions, the event passes through the same location with two opposite directions over time. Since the temporal gravity kernel size is 2, total five time steps are used to compute the flow map. In this case, the flow map over the area that an event visits multiple times within the temporal kernel size becomes zero from Equation 5. This is shown on the right of Figure 13 and the flow patterns only in the non-overlap area are extracted. In order to prevent this limitation, we can change the temporal aggregation level during the preprocessing for the functional representation so that we avoid the multiple visits within the temporal kernel size. The other solution is that we change the temporal kernel size to avoid the multiple visits with opposite directions. Considering the computation performance as seen in Table 1 and Table 2, adjusting the temporal kernel size would have more benefits since the computing time for the gravity model is much longer.

Another case happens when different objects are handled under the same density distribution. For example, there are two cars. One car moves to a location and stops there. The other car starts to move from the location to another location after the first car stops. Since we compute the event distribution for both cars together, there is no means to differentiate the objects and our system generates the continuous flow map for the cars. In order to overcome this issue, we must analyze the different objects separately and overlay all flow maps together. We plan to investigate more efficient approach to handle this case in the future.

Figure 14 (a) shows another case. (a)-(I), (a)-(II), and (a)-(III) are the density distributions and our system extracts the flow map as seen in (a)-(IV). We can see that the event is moving toward the west of Cleveland over time. However, our system shows also flow patterns toward the east of Cleveland as presented in the red rectangle. This happens when the events occur sparsely in the spatial space and our KDE does not cover the entire geographical space sufficiently. Similar cases are found in Figure 9 and Figure 10. Although we can tell that this flow is correctly estimated for some cases, we still consider that the flow is not well extracted from the data. This sparsity issue will be studied further in the future in order to remove the erroneous flow patterns at the outer edge of the kernels.

Figure 14 (b) presents unwanted flow patterns. Three event data distributions are shown in (b)-(I), (b)-(II), and (b)-(III) and the flow map extracted from the distributions about t0 in (b)-(IV). As seen in (I), there are four blue boxes whose density values decrease over time. In this case, diverging flow patterns are extracted from the blue box areas. However, the problem is found in the low density area surrounded by the blue boxes. Ideally, no flow pattern is supposed to be extracted in the area but as seen in (VI), there are flow patterns that are produced by the diverging flows from the blue boxes. Although these flow patterns are visualized in our system, their flow speeds are very slow and the patterns are noisy similar to saddle points for flows from all directions. Therefore, it is possible to differentiate these flow patterns from other accurate flows. Nonetheless, we consider that these flows are not properly computed. Since we do not use any trajectory information, it is not easy to formulate topology structures and remove these flow patterns. We will search for an improved algorithm to embed...
topology information to improve the flow map extraction in the future.

8.3 Visualization techniques

Our system provides three different visualizations for the flow map including OLIC, arrow glyph, and dense particle tracing similar to LIC. OLIC is useful to visualize sparse flow patterns as seen in Figure 8 but this is not appropriate for complicated and dense flow patterns. Arrow glyph visualization allows the user to focus on our flow map extracted by our algorithm but it is difficult to analyze detail flow maps. Additionally, the quality of the visualization for the arrow glyphs depends on the number of glyphs in the beginning. Dense particle tracing visualization is useful when we analyze the overall flow map patterns. However, the dense particle tracing visualization conveys much more detail information although we want to analyze the abstract flow maps. In addition, currently, we offer two different visualization systems, stand-alone SW and web-based SW, however, the issue of coordinates for mapping occurs as seen Figure 12 (d) and (e). The mapping on the web-based visualization SW is implemented using the 3D globe algorithm, whereas, that on the stand-alone software is implemented by the hard-coded edge point. The difference between these coordinate mappings produces the slight different visualizations. This will be investigated as future work.

8.4 Expansion to other domains

Our flow map extraction algorithm enables the data flow analysis and produces insight to the movement patterns for the spatiotemporal statistical data. Currently, there are many devices that generate data with spatiotemporal information. Therefore, it is possible to extend this algorithm to many different domains, such as, social and natural science. However, as mentioned earlier, we do not include domain characteristics during the flow map extraction, such as, additional network, transportation information, natural directional phenomena including weather condition. We plan to apply the domain features to improve our algorithm in the future.

8.5 Miscellaneous issue

As summarized in Table 1, the computation for the flow map using FMap^3D is more expensive than one using FMap^diff. Although we mentioned that the gravity model produces the flow map more adequately, we recommend that the user starts to analyze the flow map with FMap^diff since the flow maps are extracted and visualized fairly fast on GPU. In order to accelerate the FMap^3D computation, we will study the approximation of the gravity model as future work.

9 CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel spatiotemporal data analysis technique. We extracted flow maps from discrete spatiotemporal statistical data by evaluating the continuous functional representations. We employed a two-dimensional kernel density estimation to approximate the underlying data distribution and applied a gravity model that generates flow maps. We evaluated the flow map extraction model with two trajectory datasets. We have also demonstrated our flow map analysis and its visualization using four different types of event-based data. Our results showed the benefits of employing the flow maps for the trend analysis of spatiotemporal data. Our technique can be used to understand potential movement paths and trends over time within the statistical datasets, such as, criminal incident reports, economics, and social trends. As future work, we are going to investigate advanced flow map extraction models based on further statistical analysis of the data including constraint conditions. We also plan to handle noisy data distributions that cause random flow patterns. Furthermore, we will apply illustrative flow visualization techniques for better perception of the flow. We will extend our flow analysis techniques to multivariate spatiotemporal data analysis as well.

REFERENCES

