

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

Seokyeon Kim, Seongmin Jeong, Insoo Woo, Yun Jang<sup>✉</sup>, *Member, IEEE*,  
Ross Maciejewski<sup>✉</sup>, *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 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 streak-lines (e.g., [5], [27], [39], [49]). 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 was 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.

Fig. 1 illustrates the concept of potential event flow extraction. Spatiotemporal data is visualized on the map in

- S. Kim, S. Jeong, and Y. Jang are with Sejong University, Seoul, South Korea. E-mail: {ksy0586, jsm}@sju.ac.kr, jangy@sejong.edu.
- I. Woo is with Intel Folsom, Folsom, CA 95630. E-mail: insoo0712@gmail.com.
- R. Maciejewski is with Arizona State University, Tempe, AZ 85287-0001. E-mail: rmaciej@asu.edu.
- D.S. Ebert is with Purdue University, West Lafayette, IN 47907. E-mail: ebertd@purdue.edu.

Manuscript received 2 May 2016; revised 20 Jan. 2017; accepted 24 Jan. 2017.  
Date of publication 8 Feb. 2017; date of current version 26 Jan. 2018.

Recommended for acceptance by G. Andrienko.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TVCG.2017.2666146

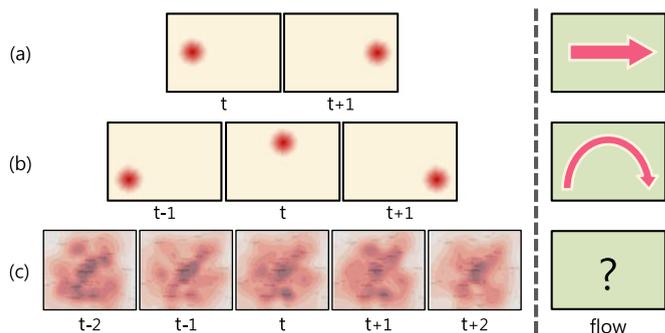


Fig. 1. Illustration of potential flow map extraction. When event heatmaps for two consecutive time steps are given as in (a), we can indicate that the event is relocated from the left to the right. Similarly, there are three events over time and the event flow is easily imagined as in (b). However, when complicated event heatmaps are provided, it is not easy to extract the event flows in (c).

geographical space in accordance with 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 Fig. 1a, 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 Fig. 1b. Both Figs. 1a and 1b 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. Fig. 1c 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 Fig. 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 requiring

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) [50]. 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 trajectory information is not easily collected and requires massive amounts of storage for further analysis, we do not make use of 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

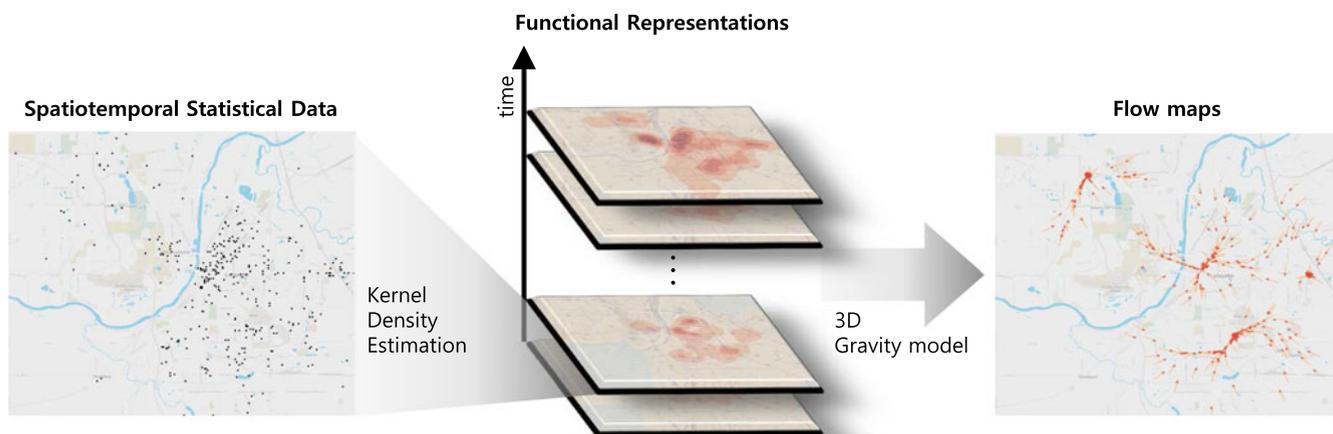


Fig. 2. System overview. Discrete spatiotemporal data is represented as a continuous function (KDE) and flow maps are extracted using 3D gravity model for temporal trends (movement flows).

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. [47] 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 coordinate, 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 an 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 [45], [46] presents migration maps using the N-squared table of geographical interactions. His insight originates from his First Law of Geography; “*Everything is related to everything else, but near things are more related than distant things*” [44]. 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], [48]. 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], [48]. For example, 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 [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. [50] 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 for 2D vector data through a quantitative evaluation study. Their results show that both expert and novice users better performed given tasks when they

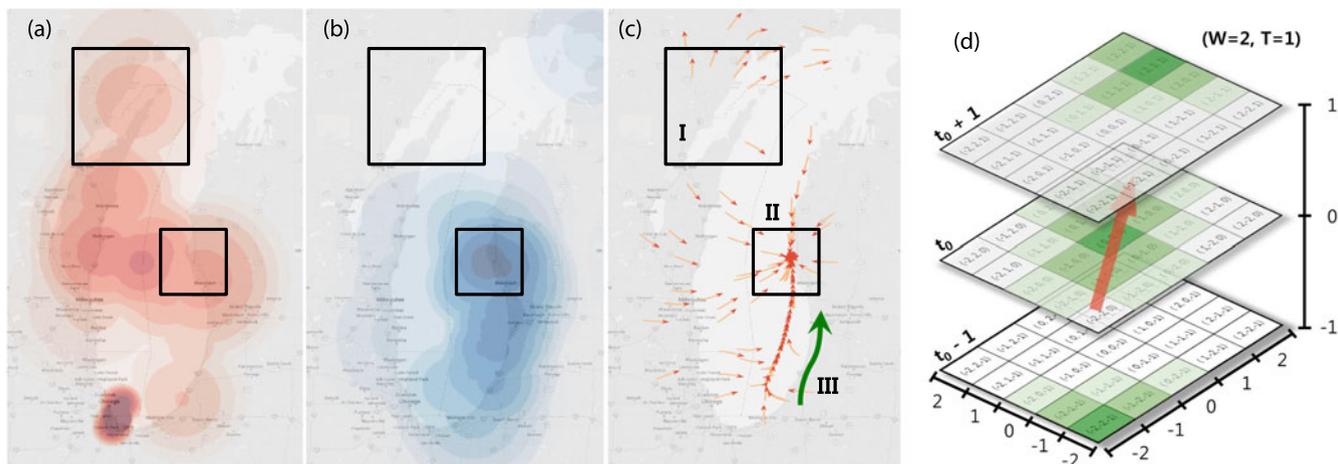


Fig. 3. Data distributions are shown as (a) a heatmap in the current,  $t_0$ , and (b) a heatmap in the future,  $t_0 + 1$ . (c) The flow map from  $t_0$  to  $t_0 + 1$  is extracted using the gravity model. The area (I) shows diverging flows and the area (II) presents converging flows when the density 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 heatmap.

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 of 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, 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 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. Fig. 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 functional representations using kernel density estimation. 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 Sections 4, 5, and 6, respectively. Sample visualizations from our flow map analysis system are shown in Fig. 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 Fig. 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 Fig. 3 (the green arrow in (c)).

In order to illustrate our algorithm more clearly, we present two test cases under ideal conditions in Fig. 4. There are two event locations in Fig. 4a. The density of the upper left event decreases while the density of the lower right event increases. This demonstrate expected diverging and converging patterns, respectively. These flow patterns are obtained by our approach as shown in Fig. 4a. Fig. 4b shows the 2nd 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. The spatial distribution of the events at

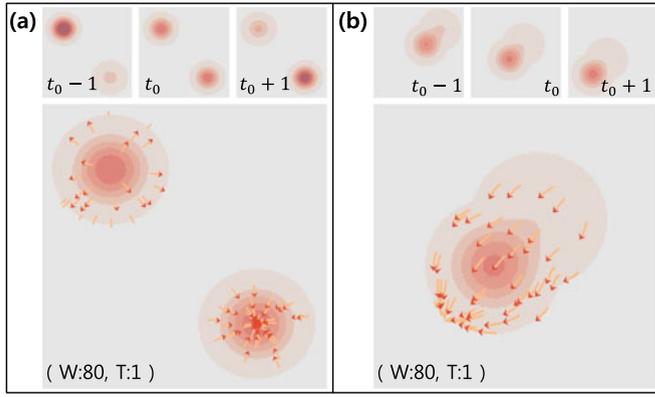


Fig. 4. Two simple event patterns. (a) Presents diverging and converging patterns. (b) Shows the flow map using our approach when the events move toward the left lower corner.

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 to spatiotemporal data since geolocation and time are often on different scales.

In this work, we utilize a simple representation of discrete spatiotemporal data that is an array of spatial distributions over time. In order to obtain continuous representations from discrete data, the kernel density estimation (KDE) approach is applied to the data distribution. Initially, we have combined the spatial dimension and temporal dimension together to analyze flows, but this approach showed only either diverging or converging flow patterns. Therefore, each time step is encoded with 2D kernels separately, and the functional representation is written as follows:

$$\hat{f}_{2D}(x, y) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_{s,i}^2} K_s \left( \frac{x - x_i}{h_{s,i}}, \frac{y - y_i}{h_{s,i}} \right), \quad (1)$$

where  $\hat{f}_{2D}(x, y)$  is the density estimate at a given location,  $(x, y)$ ,  $N$  is the total number of samples, and  $h_{s,i}$  is the bandwidth of  $K_s$ . We use the *Triweight kernel* for  $K_s$ :

$$\mathbf{K}_{s,t}(\mathbf{u}) = \frac{35}{32} (1 - \mathbf{u}^2)^3 \mathbf{1}_{(\|\mathbf{u}\| \leq 1)}. \quad (2)$$

Note that any kernel can be used to represent discrete data. However, one has to make sure that the first derivative of a kernel exists within the entire spatial data range, otherwise, the first derivative field of Equation (1) will be discontinuous or incorrect. For example, the kernels having sharp edges at the end of kernel, such as the *Epanechnikov kernel* or *Cosine kernel*, result in discontinuous vector fields when kernels overlap. Since the *Triweight kernel* covers a narrow range around an event, it creates less uncertain functional representations compared to wider range kernels, such as Gaussians. Also, the *Triweight kernel* requires less compute time compared to kernels with exponential functions.

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 kernel,  $K_s$ , proposed by Silverman [42]. The bandwidth ( $h_{s,i}$ ) 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 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:

$$\mathbf{g}_{(i,j)} = (m_i^{a_0} \cdot m_j^{a_1}) / d_{ij}^2, \quad (3)$$

where  $m_i$  and  $m_j$  are the masses at  $i$  and  $j$ ,  $d_{ij}$  is the distance between  $i$  and  $j$ , and  $a_0$  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 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:

$$\mathbf{g}_{(i,j)} = \frac{\left( \hat{f}_{2D}(x_i, y_i) \right)^{a_0} \cdot \left( \hat{f}_{2D}(x_j, y_j) \right)^{a_1}}{d_{ij}^2}. \quad (4)$$

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

$$\mathbf{FMap}_{3D}(\mathbf{x}, \mathbf{y}, \mathbf{t}) = \sum_{p=-W}^W \sum_{q=-W}^W \sum_{r=-T}^T (p, q, r) \cdot \frac{\left( \hat{f}_{2D}(x, y) \Big|_t \right)^{a_0} \cdot \left( \hat{f}_{2D}(x_p, y_q) \Big|_{t_r} \right)^{a_1}}{d_{ij}^2}, \quad (5)$$

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

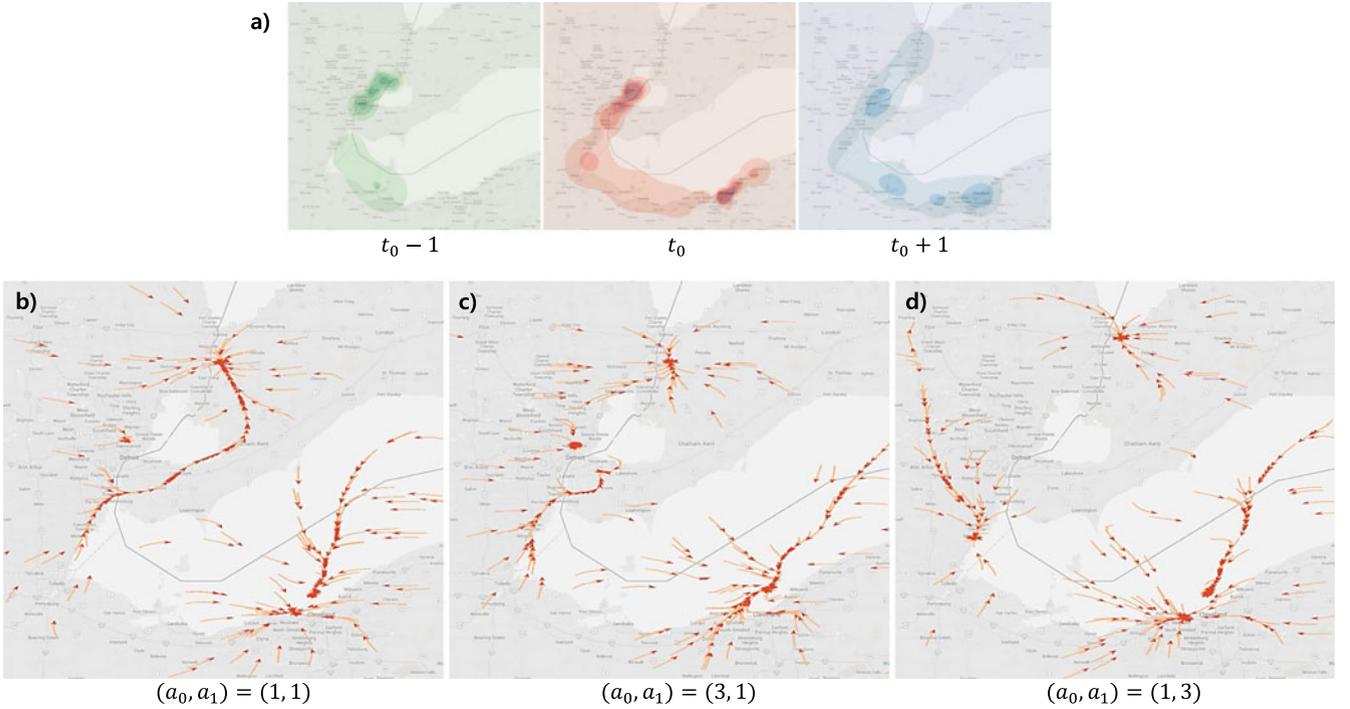


Fig. 5. Parameter comparisons for  $a_0$  and  $a_1$  in Equation (5). (a) Presents three heat map visualizations from  $t_0 - 1$  to  $t_0 + 1$ . The flow maps are extracted by varying the values,  $a_0$  and  $a_1$ , in (b), (c), and (d). When  $a_0$  increases, the flow patterns are observed as similar 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  $t_0$ .

in the directional information in the flow map. Fig. 3d illustrates an example of the gravity kernel when  $W$  is 2 and  $T$  is 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 Fig. 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 Figs. 5b, 5c, and 5d. When  $a_0$  increases, the flow patterns are similar to the potential flow at  $t_0$ . On the other hand, when  $a_1$  increases, our model tends to produce the temporal flow patterns over the relationship only between the neighboring steps. This tends to reduce the effect of the time step  $t_0$ . Fig. 5c shows the flow patterns similar to the potential flow at  $t_0$ , whereas, Fig. 5d 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:

$$\mathbf{FMap}_{\text{diff}}(\mathbf{x}, \mathbf{y})|_{(t_0, t_1)} = \left( \hat{f}_{2D}(x, y)|_{t_1} \right) - \left( \hat{f}_{2D}(x, y)|_{t_0} \right). \quad (6)$$

However,  $\mathbf{FMap}_{\text{diff}}(x, y)$  extracts only local flow patterns that do not represent the global flows. Fig. 6 shows the comparison between difference-based flow extraction and gravity-based flow extraction. Figs. 6a and 6b are the heatmaps at the current time,  $t_0$ , and the future,  $t_1$ . (c) Presents the flow map visualization using the difference-based flow extraction, whereas, (d) shows the flow map visualization with the gravity-based flow extraction.

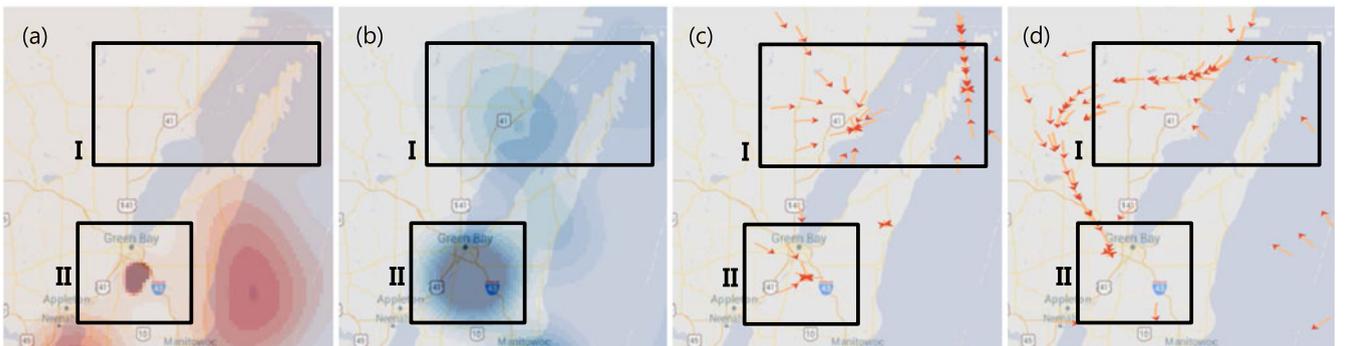


Fig. 6. Comparison between the difference-based and the gravity-based flow map extraction. (a) and (b) present heatmaps at the current time,  $t_0$ , and the future,  $t_1$ , respectively. (c) Presents the flow map visualization using the difference-based flow extraction, whereas, (d) shows the flow map visualization with the gravity-based flow extraction.

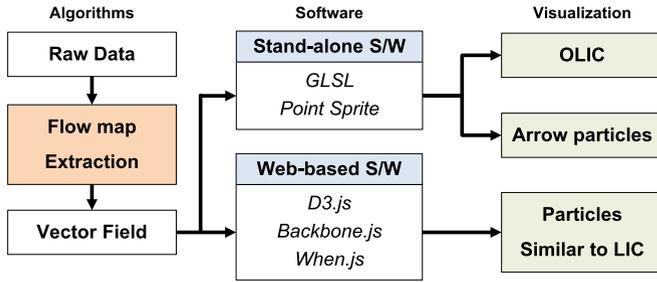


Fig. 7. Visualization pipeline. The vector field is computed using the functional representation from the raw data with KDE and gravity model. The vector fields are visualized on either stand-alone client software or web-based software. Our system provides three different types of visualizations including OLIC, arrow glyphs, and particle tracing similar to LIC.

between Figs. 6a and 6b are found in area (I) and area (II). Fig. 6c presents the flow map visualization with the difference-based flow extraction. Fig. 6c shows several converging flows toward the center of the 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. Fig. 6d shows the flow map using the gravity-based flow extraction. The global flow patterns are extracted as the curve from Fig. 6 area (I) to area (II).

## 6 VISUALIZATION SYSTEM OF SPATIOTEMPORAL DATA

In order to analyze spatiotemporal trends in 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. Fig. 7 illustrates the visualization pipeline for the flow map. The computation for the algorithm is performed, and the system transfers the vector fields to either a stand-alone software client or a web-based client 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 the system implementation details in this section.

### 6.1 Flow Map Visualization

In flow visualization using particle advection, initial particle seeding is one of the challenging issues to obtain the best illustration of flow patterns within a vector field [49]. Previous work uses flow feature points from fluid simulation [52]. 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 projections 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 geometry 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. Tables 1 and 2 show the computing performance according to our model parameters. Data load and histogram generation time,  $T_{load}$ , KDE preparation time,  $T_{pre}$ , computing time for KDE,  $T_{kde}$ , computing time for the gravity model,  $T_{grav}$ , and data transfer time from GPU to CPU,  $T_{tran}$ , are compared, respectively. In Table 1, GPU performance is about 30 times faster than CPU  $FMap_{diff}$ . Note that computing  $FMap_{3D}$  on the CPU is not possible in our experiment. For the  $FMap_{3D}$  computation, as  $W$  increases, the computing time becomes longer due to the increase in 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, 16 GB CPU memory, and GeForce GTX 660 with 2 GB GPU memory.

TABLE 1  
Comparison of CPU and GPU Computing Time

computing type	model	total points	number of timesteps	gravity parameters		$T_{load}$ (ms)	$T_{pre}$ (ms)	$T_{kde}$ (ms)	$T_{grav}$ (ms)	$T_{tran}$ (ms)	total (ms)
				W	T						
CPU	$FMap_{diff}$	3,056	31	-	-	93	514	34,135	-	-	34,742
	$FMap_{diff}$			-	-	96	518	247	-	189	1,050
GPU	$FMap_{3D}$	3,056	31	1	1	99	730	243	82	199	1,353
				10	1	90	699	209	3,967	191	5,157
				30	1	108	589	247	33,469	192	34,606
				50	1	98	757	243	86,913	192	88,204

TABLE 2  
Performance of GPU Computing Time for Different Datasets

data	total points	number of timesteps	gravity parameters		$T_{load}$ (ms)	$T_{pre}$ (ms)	$T_{kde}$ (ms)	$T_{grav}$ (ms)	$T_{tran}$ (ms)	total (ms)
			W	T						
Super Bowl	4,659	24	30	2	49	60	35	4,796	33	4,973
Boston Marathon	5,224	10	20	1	80	379	119	1,163	14	1,755
SAR	23,416	4	120	1	205	553	738	210,577	229	212,302
syndromic	693,878	86	50	3	3,542	1,782	6,897	135,564	271	148,056

## 7 DATA EXPLORATION

Our system extracts the flow maps and presents the spatio-temporal 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 Sections 7.2, 7.3, and 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 [53]. Figs. 8a, 8b, 8c, 8d, and 8e present the data visualizations with every 5 minutes GPS trajectory data with the heatmaps. The flow maps, Figs. 8f and 8g, 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

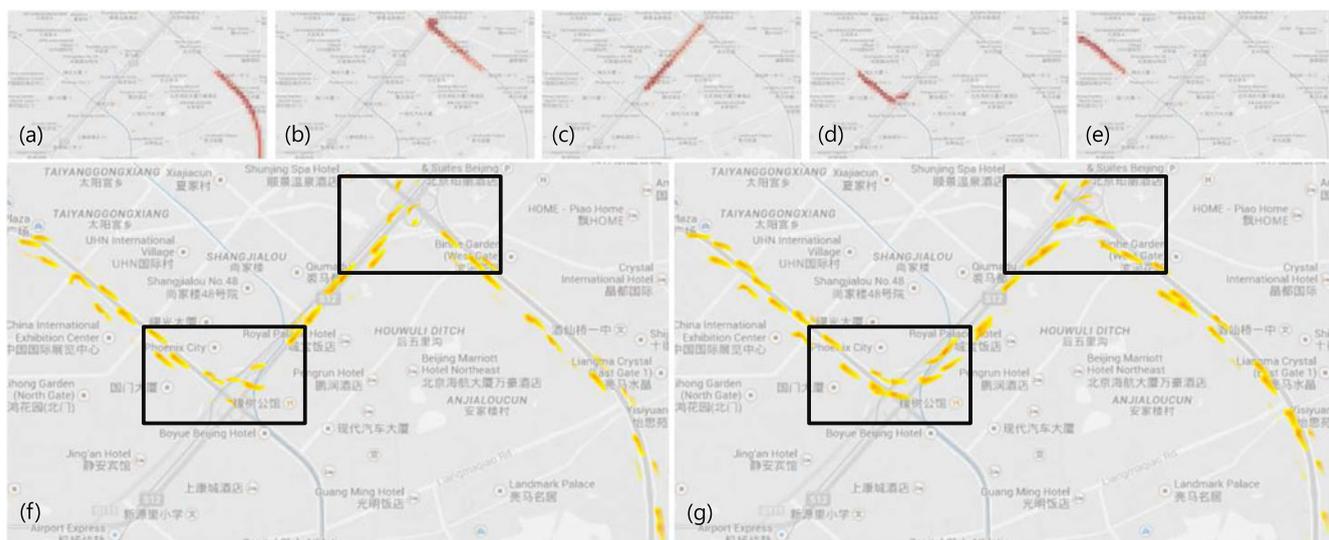


Fig. 8. The flow map visualizations of GPS trajectory data. (a)-(e) show the data distributions starting from (a) and moving toward (e). (f) presents the flow map extracted from our gravity model with three time steps. (g) is the flow map with five time steps. The difference between (f) and (g) is found at intersections marked in the rectangular boxes. The flow map is visualized with OLIC.

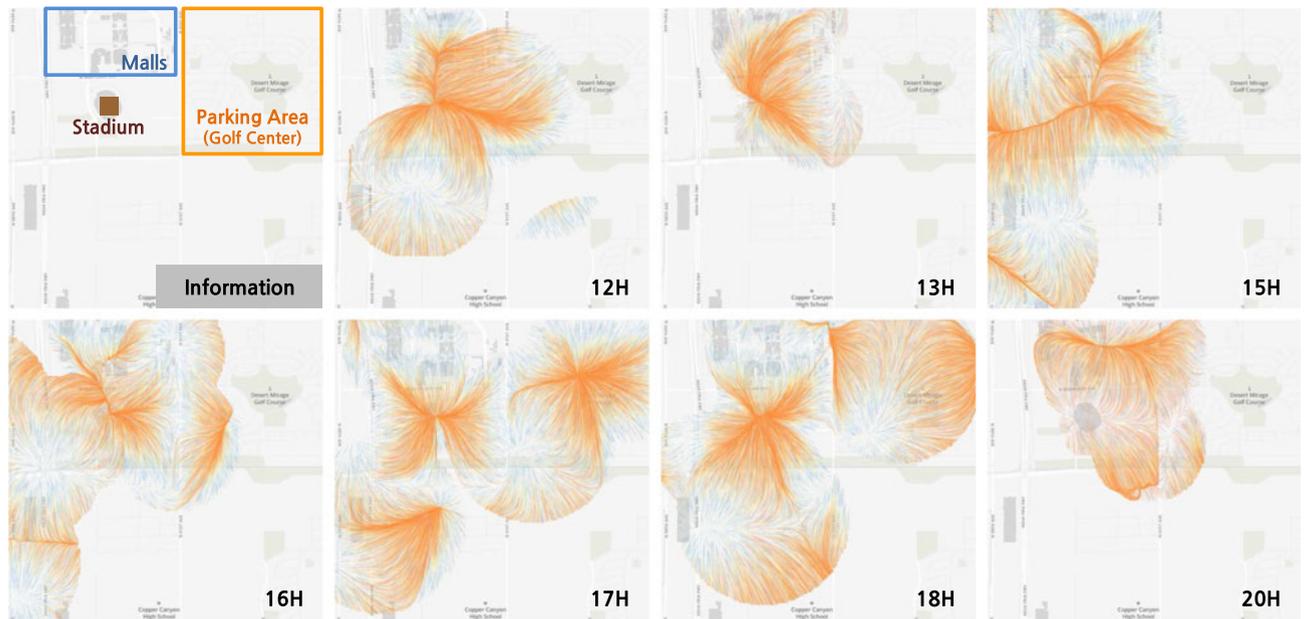


Fig. 9. The flow maps during super bowl in 2015. Three hot spots are marked in the information; the stadium, malls, and remote parking area. The tweet messages related with the game are analyzed between 12:00 and 20:00. The game started at 16:30 and the converging flow patterns were found around the 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.

certain time range and generate new data with multiple time steps. Figs. 8a, 8b, 8c, 8d, and 8e 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 Fig. 8f is generated with three time steps and one in Fig. 8g 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 Fig. 8f are following the street shapes, whereas, the flow directions at the intersections in Fig. 8g 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, the 2015 Super Bowl and the 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 three hot spots, such as the stadium, malls, and a remote parking area as indicated in Fig. 9. People started to move to the stadium from 12:00 until the game starting time, which was 16:30. As time went on, the movement flows were dominant toward the stadium where there were many available parking spaces until 15:00. Then from 16:00, the major flow was found in the remote parking area, which is apart from the main stadium. This indicates that there was no available parking spaces near the stadium, and people started to move to other parking spaces in the remote parking areas. Interesting patterns were found around 20:00. There was a flow from the stadium to the malls, which might indicate that people went to the malls again for food and drinks or picking up their cars.

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 [51]. 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 Figs. 10a, 10b, and 10c, there were mixed patterns, such as random and gathering movements at 12:00, 13:00, and 14:00. Then the bombs exploded at 14:49 PM and right after that, people escaped from the bombing sites, which are marked in Fig. 10d. The flow patterns showed that all people moved away and our flow map model caught these

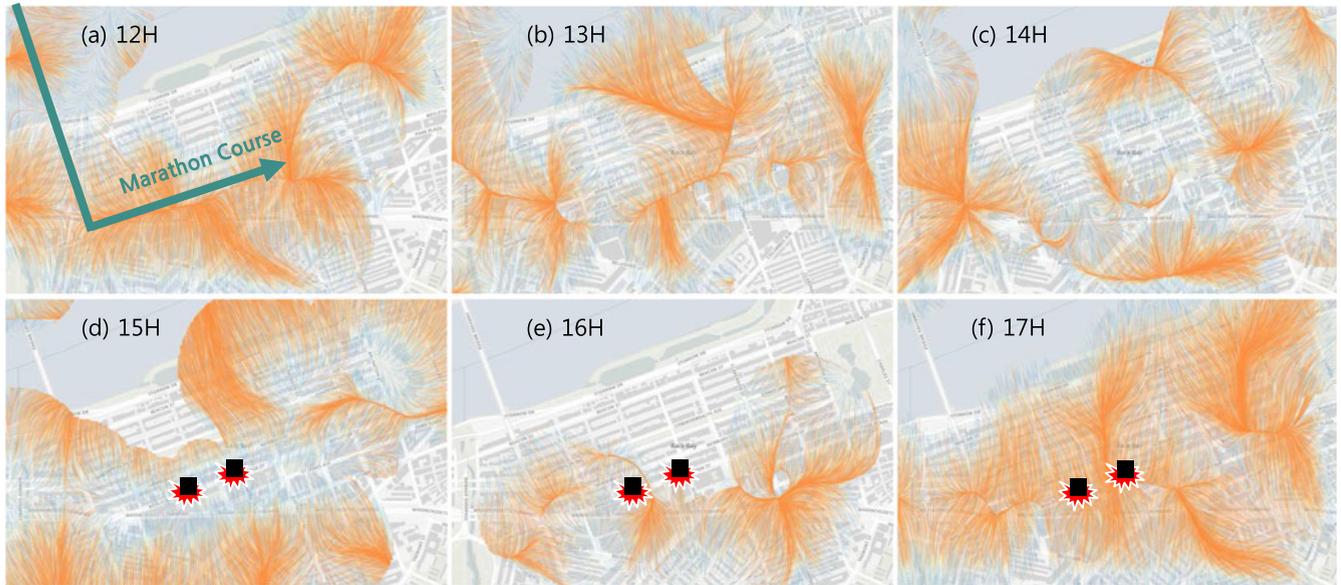


Fig. 10. The flow maps of the Boston Marathon bombing in 2013. The tweet messages are analyzed between 13:00 and 17:00. The bombs exploded at 14:49 and the escape flow patterns are found right after the bomb explosion in (c). Random patterns and gathering patterns at the marathon course are found in (a) and (b). People started to gather again at the bombing sites two hours later in (d).

patterns. After two hours, people slowly came to the bombing sites to investigate or watch the scene. This is shown in Fig. 10f.

### 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. Figs. 11a, 11b, and 11c 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 following small cities of Muskegon, Ludington, Traverse City, Sarnia, during summer; whereas, most SAR cases are found near several large ports during winter. This observation is highlighted in the flow map in (d). This implies that the safety or coast guard resources should be allocated across all these area flow patterns in summer and focused near major highlighted shipping ports during winter.

### 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. Fig. 12 illustrates how the outbreaks change over time as heatmaps in (a)-(c). The outbreaks occurred around Indianapolis on July 26, 2006 and propagated to near areas and Northern Indiana

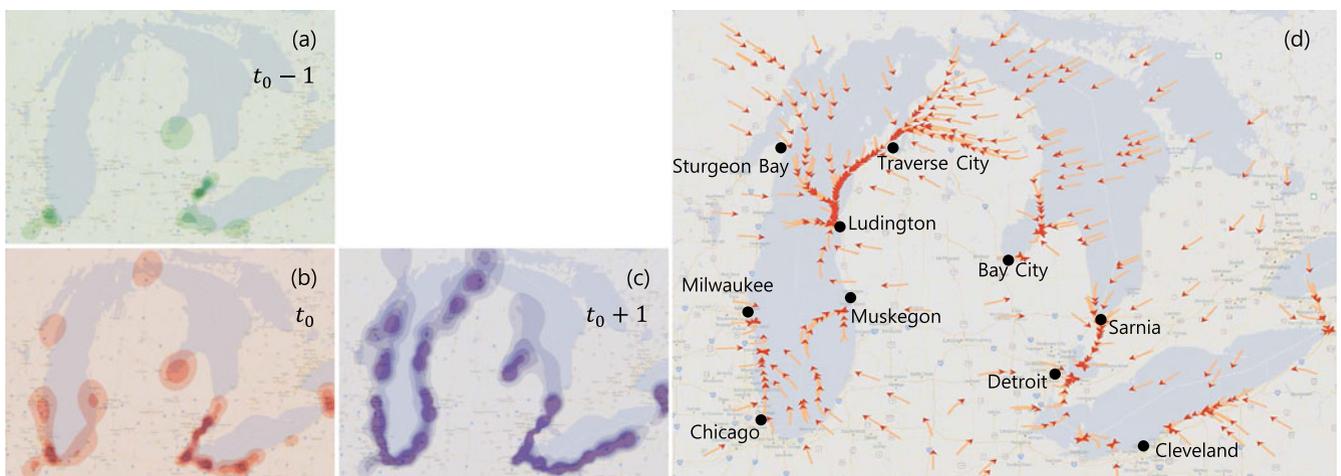


Fig. 11. (a) is a heatmap in the past,  $t_0 - 1$ , (b) in the current,  $t_0$ , and (c) in the future,  $t_0 + 1$ , for the search and rescue case. (d) is the flow map visualization and the flow trends are extracted based on the density distributions (a, b, c) from the past to the future.

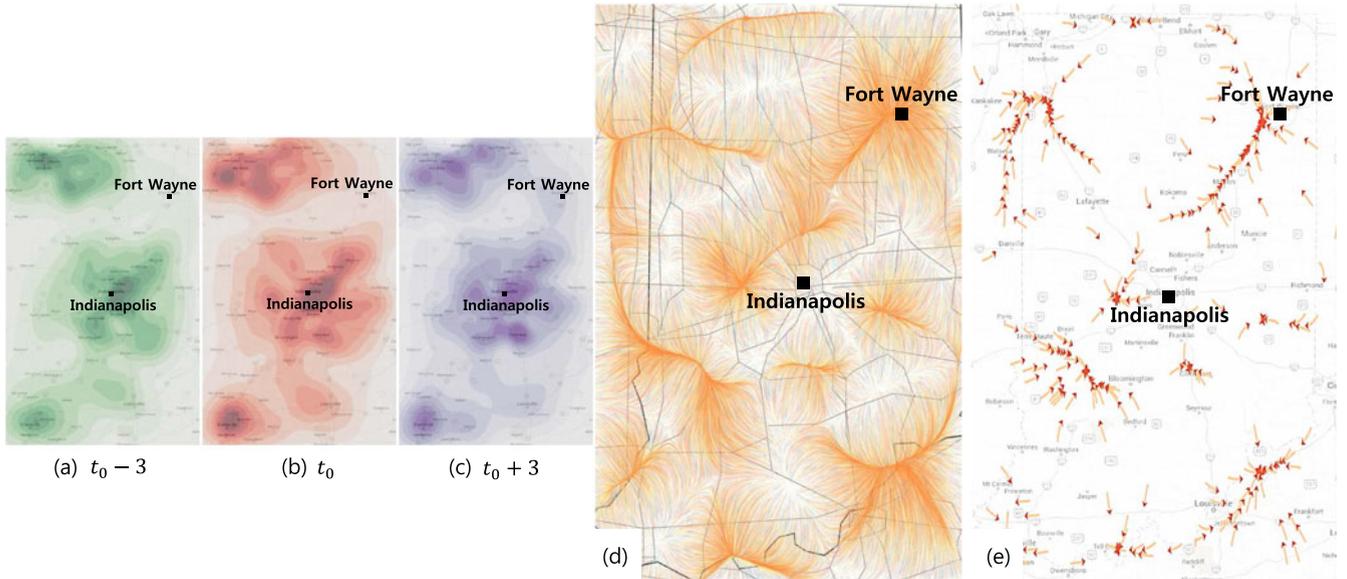


Fig. 12. Outbreaks of synthetic syndromic data. (a), (b), and (c) show the heatmaps of the data distribution and the corresponding flow map is presented in (d) and (e). Flow paths from Indianapolis to Northern Indiana are extracted as the disease spreading direction.

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 Northern Indiana along local highways and state border between Indiana and Illinois are extracted as the disease spreads in Figs. 12d and 12e, which is not easily imagined only from the heatmaps. Small flow patterns are also visible toward some small cities in the suburbs of Indianapolis. In addition, there are major flow paths toward Fort Wayne in the northeast Indiana, which is actually a future ( $t_0 + 1$ ) event at the time ( $t_0$ ) 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

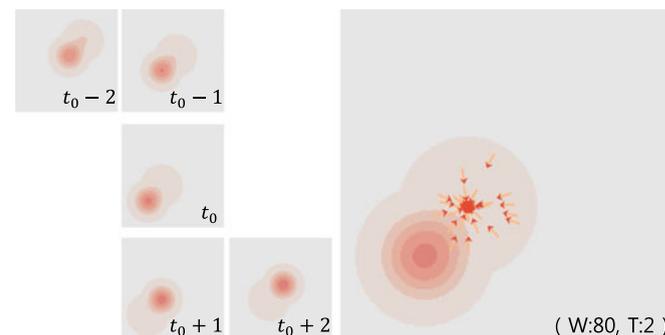


Fig. 13. The event distribution moves to the bottom left ( $t_0 - 2$  and  $t_0 - 1$ ) and then moves back to the top right ( $t_0 + 1$  and  $t_0 + 2$ ). Our algorithm generates the flow map only for the non-overlap area if the overlap area is placed within the temporal kernel size.

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 flow maps from statistical data without any trajectory information, there are a few cases where our system is not able to properly extract flow maps. These special cases are discussed in below.

The first case is when an event moves towards one direction and then comes back to the original location as seen in Fig. 13. The density distributions are shown on the left from ( $t_0 - 2$ ) to ( $t_0 + 2$ ). 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, a total of 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 (Equation (5)). This is shown in the right of Fig. 13 where the flow patterns only in the non-overlap area are extracted. In order to prevent this limitation, we can change

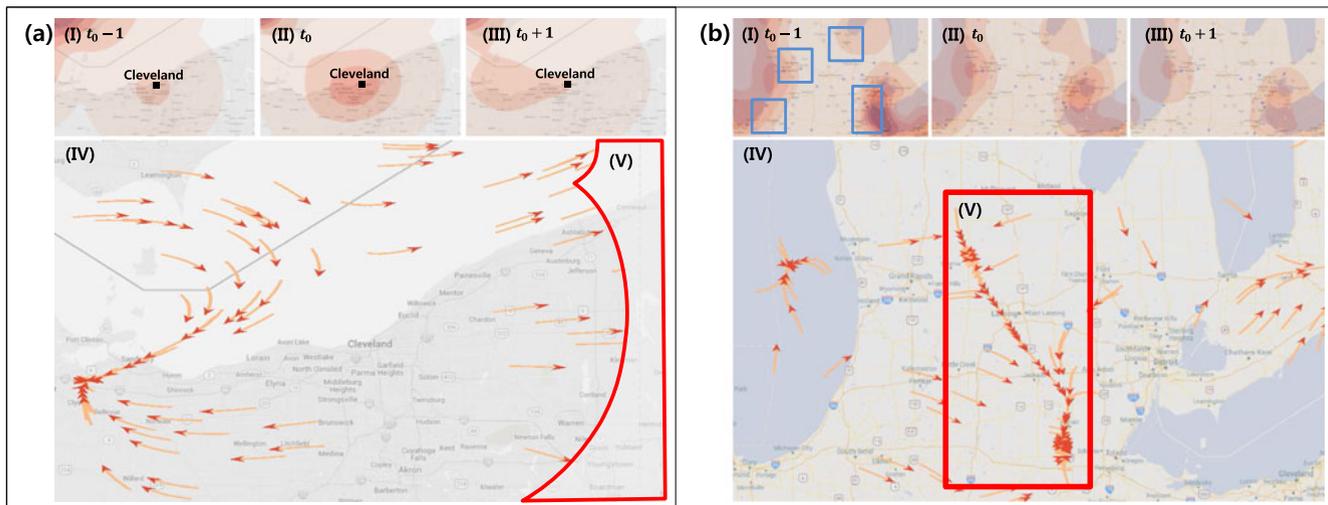


Fig. 14. (a)-(I), (a)-(II), and (a)-(III) present heat maps from  $t_0 - 1$  to  $t_0 + 1$ . (a)-(IV) is the flow map from the heat maps. (a)-(V) presents the zero vector field area, such as the kernel edges. We stop advecting the vector field at the kernel edges since there is no actual field beyond the edges. (b)-(I), (b)-(II), and (b)-(III) are different heat maps. Incorrect flow patterns are extracted in the area in (b)-(V) surrounded by the high density areas (blue boxes). All diverging flows from the blue box areas are merged in the red box area but it is not properly estimated since there is no density change in the area.

the temporal aggregation level during the functional representation preprocessing so that we avoid the multiple visits within the temporal kernel size. The other solution is to change the temporal kernel size to avoid the multiple visits with opposite directions. Considering the computation performance, as seen in Tables 1 and 2, adjusting the temporal kernel size would have more benefits since the computing time for the gravity model is much longer.

Another case occurs when different objects are handled under the same density distribution. For example, when 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 approaches to handle this case in the future.

Fig. 14a 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 also shows 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 Figs. 9 and 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.

Fig. 14b 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  $t_0$  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 should be extracted in this 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 topological 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 Fig. 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 detailed information even when we want to only analyze the abstract flows. In addition, currently, we offer two different visualization systems, stand-alone SW and web-based SW; however, a small mapping coordinate issues occur as seen Figs. 12d and 12e. The mapping in the web-based visualization SW is implemented using a 3D globe algorithm; whereas, that in the stand-alone software is implemented by the hard-coded edge points. The difference between these coordinate mappings produces slightly different visualizations. This will be investigated in future work.

## 8.4 Expansion to Other Domains

Our flow map extraction algorithm enables data flow analysis and produces insight into the movement patterns within 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 sciences. However, as mentioned earlier, we do not include domain characteristics during flow map extraction, such as, additional network, transportation information, natural directional phenomena (e.g., weather conditions). This currently limits the benefits of the system and we plan to apply these 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 accurately, 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 the GPU. In order to accelerate the  $FMap_{3D}$  computation, we will study approximating 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.

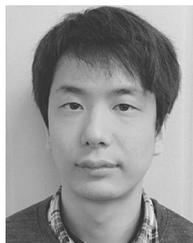
## ACKNOWLEDGMENTS

This work was supported in part by the Ministry of Science, ICT and Future Planning, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-R2718-16-0017) and the grant (R0190-15-2016, Development of Complex Fast Stream Big Data Processing based on In-memory Technology in Distributed Environment) supervised by the IITP (Institute for Information & communications Technology Promotion), and the National Science Foundation grant number 1350573. We would like to thank the reviewers for their valuable suggestions and comments, which helped to improve the presentation of this work. Yun Jang is the corresponding author.

## REFERENCES

- [1] J. E. Anderson, "The gravity model," Working Paper 16576, National Bureau of Economic Research, 2010.
- [2] N. Andrienko and G. Adrienko, "Spatial generalization and aggregation of massive movement data," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 205–219, Feb. 2011.
- [3] J. M. Barrios, W. W. Verstraeten, P. Maes, J.-M. Aerts, J. Farifteh, and P. Coppin, "Using the gravity model to estimate the spatial spread of vector-borne diseases," *Int. J. Environ. Res. Public Health*, vol. 9, no. 12, pp. 4346–4364, 2012.
- [4] J. H. Bergstrand, "The gravity equation in international trade: Some microeconomic foundations and empirical evidence," *Rev. Econ. Statist.*, vol. 67, no. 3, pp. 474–481, 1985.
- [5] K. Buchin, B. Speckmann, and K. Verbeek, "Flow map layout via spiral trees," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2536–2544, Dec. 2011.
- [6] K. Bürger, P. Kondratieva, J. Krüger, and R. Westermann, "Importance-driven particle techniques for flow visualization," in *Proc. IEEE Pacific Vis. Symp.*, 2008, pp. 71–78.
- [7] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," in *Proc. Conf. Comput. Graph. Interactive Tech.*, 1993, pp. 263–270.
- [8] B. Dent, *Cartography: Thematic Map Design*. New York, NY, USA: WCB/McGraw-Hill, 1999.
- [9] R. Eccles, T. Kapler, R. Harper, and W. Wright, "Stories in geotime," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2007, pp. 19–26.
- [10] A. Forsberg, J. Chen, and D. Laidlaw, "Comparing 3D vector field visualization methods: A user study," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1219–1226, Nov./Dec. 2009.
- [11] P. Gatalsky, N. Andrienko, and G. Andrienko, "Interactive analysis of event data using space-time cube," in *Proc. Int. Conf. Inf. Vis.*, 2004, pp. 145–152.
- [12] D. Guo, "Flow mapping and multivariate visualization of large spatial interaction data," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1041–1048, Nov./Dec. 2009.
- [13] D. Guo, J. Chen, A. MacEachren, and K. Liao, "A visualization system for space-time and multivariate patterns (VIS-STAMP)," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 6, pp. 1461–1474, Nov./Dec. 2006.
- [14] D. Guo and X. Zhu, "Origin-destination flow data smoothing and mapping," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2043–2052, Dec. 2014.
- [15] T. Hägerstrand, "What about people in regional science?" *Papers Regional Sci.*, vol. 24, no. 1, pp. 6–21, Dec. 1970.
- [16] M. Indulska and M. E. Orlowska, "Gravity based spatial clustering," in *Proc. 10th ACM Int. Symp. Advances Geographic Inf. Syst.*, 2002, pp. 125–130.
- [17] W. Javed, S. Ghani, and N. Elmquist, "Gravnav: Using a gravity model for multi-scale navigation," in *Proc. Int. Work. Conf. Adv. Visual Interfaces*, 2012, pp. 217–224.
- [18] M. Jern, T. Astrom, and S. Johansson, "Geoanalytics tools applied to large geospatial datasets," in *Proc. Int. Conf. Inf. Vis.*, 2008, pp. 362–372.
- [19] D. Karemera, V. I. Oguledo, and B. Davis, "A gravity model analysis of international migration to North America," *Appl. Econ.*, vol. 32, no. 13, pp. 1745–1755, 2000.
- [20] Á. Kincses and G. Tóth, "The application of gravity model in the investigation of spatial structure," *Acta Polytechnica Hungarica*, vol. 11, no. 2, pp. 5–19, 2014.
- [21] M. Kraak, "The space-time cube revisited from a geovisualization perspective," in *Proc. Int. Cartographic Conf.*, 2003, pp. 1988–1996.
- [22] M.-J. Kraak and P. Madzudzo, "Space time visualization for epidemiological research," in *Proc. 23rd Int. Cartographic Conf.*, 2007, pp. 302–314.
- [23] J. Krüger, P. Kipfer, P. Kondratieva, and R. Westermann, "A particle system for interactive visualization of 3D flows," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 6, pp. 744–756, Nov./Dec. 2005.
- [24] D. H. Laidlaw, et al., "Comparing 2D vector field visualization methods: A user study," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 1, pp. 59–70, Jan./Feb. 2005.
- [25] R. S. Laramée, et al., "Applications of texture-based flow visualization," *Eng. Appl. Comput. Fluid Mech.*, vol. 2, no. 3, pp. 264–274, 2008.
- [26] R. S. Laramée, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf, "The state of the art in flow visualization: Dense and texture-based techniques," *Comput. Graph. Forum*, vol. 23, pp. 203–221, 2004.

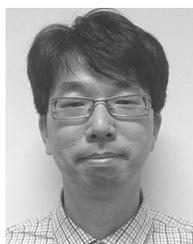
- [27] S. J. Lavin and R. S. Cerveny, "Unit-vector density mapping," *Cartographic J.*, vol. 24, no. 2, pp. 131–141, 1987.
- [28] J. J. Lewer and H. Van den Berg, "A gravity model of immigration," *Econ. Lett.*, vol. 99, no. 1, pp. 164–167, 2008.
- [29] X. Li, A. Çöltekin, and M.-J. Kraak, "Visual exploration of eye movement data using the space-time-cube," in *Proc. Int. Conf. Geographic Inf. Sci.*, 2010, pp. 295–309.
- [30] X. Li, H. Tian, D. Lai, and Z. Zhang, "Validation of the gravity model in predicting the global spread of influenza," *Int. J. Environ. Res. Public Health*, vol. 8, no. 8, pp. 3134–3143, 2011.
- [31] Z. Liu, S. Cai, J. E. Swan II, R. J. Moorhead II, J. P. Martin, and T. J. Jankun-Kelly, "A 2D flow visualization user study using explicit flow synthesis and implicit task design," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 5, pp. 783–796, May, 2012.
- [32] W. Luo, A. M. MacEachren, P. Yin, and F. Hardisty, "Spatial-social network visualization for exploratory data analysis," in *Proc. ACM SIGSPATIAL Int. Workshop Location-Based Social Netw.*, 2011, pp. 3:1–3:4.
- [33] R. Maciejewski, et al., "Generating synthetic syndromic-surveillance data for evaluating visual-analytics techniques," *IEEE Comput. Graph. Appl.*, vol. 29, no. 3, pp. 18–28, May/June, 2009.
- [34] R. Maciejewski, et al., "A visual analytics approach to understanding spatiotemporal hotspots," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 2, pp. 205–220, Mar./Apr. 2010.
- [35] A. Malik, R. Maciejewski, B. Maule, and D. Ebert, "A visual analytics process for maritime resource allocation and risk assessment," in *Proc. IEEE Conf. Visual Analytics Sci. Technol.*, 2011, pp. 221–230.
- [36] O. Mattausch, T. Theußl, H. Hauser, and M. E. Gröller, "Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines," in *Proc. Spring Conf. Comput. Graph.*, 2003, pp. 213–222.
- [37] T. Nakaya and K. Yano, "Visualising crime clusters in a space-time cube: An exploratory data-analysis approach using space-time kernel density estimation and scan statistics," *Trans. GIS*, vol. 14, no. 3, pp. 223–239, 2010.
- [38] D. J. Peuquet and M.-J. Kraak, "Geobrowsing: Creative thinking and knowledge discovery using geographic visualization," *Inf. Vis.*, vol. 1, pp. 80–91, 2002.
- [39] D. Phan, L. Xiao, R. Yeh, and P. Hanrahan, "Flow map layout," in *Proc. IEEE Symp. Inf. Vis.*, 2005, pp. 219–224.
- [40] R. Ramos and J. Surinach, "A Gravity Model of Migration between ENC and EU," IZA Discussion Papers 7700, Institute for the Study of Labor (IZA), 2013.
- [41] A. Salvo, "Trade flows in a spatial oligopoly: Gravity fits well, but what does it explain?" *Can. J. Econ./Revue Canadienne d'Économique*, vol. 43, no. 1, pp. 63–96, 2010.
- [42] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL, USA: Chapman & Hall/CRC, 1986.
- [43] T. A. Slocum, R. B. McMaster, F. C. Kessler, and H. H. Hugh, *Thematic Cartography and Geovisualization*. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2009.
- [44] W. Tobler, "A computer movie simulating urban growth in the detroit region," *Econ. Geography*, vol. 46, no. 2, pp. 234–240, 1970.
- [45] W. Tobler, "Movement mapping," Retrieved April 30, 2016, <http://www.csiss.org/clearinghouse/FlowMapper/>
- [46] W. R. Tobler, "Experiments in migration mapping by computer," *Amer. Cartographer*, vol. 14, no. 2, pp. 155–163, 1987.
- [47] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3D information visualization for time dependent data on maps," in *Proc. Int. Conf. Inf. Vis.*, 2005, pp. 175–181.
- [48] J. Truscott and N. M. Ferguson, "Evaluating the adequacy of gravity models as a description of human mobility for epidemic modelling," *PLoS Comput. Biol.*, vol. 8, no. 10, 2012, Art. no. e1002699.
- [49] G. Turk and D. Banks, "Image-guided streamline placement," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 453–460.
- [50] R. Wegenkittl, E. Gröller, and W. Purgathofer, "Animating flow fields: Rendering of oriented line integral convolution," in *Proc. Comput. Animation*, 1997, pp. 15–21.
- [51] Wikipedia. Boston marathon bombings–Wikipedia, the free encyclopedia, Wikipedia, New York, NY, USA, 2013, Accessed on: Mar. 31, 2015.
- [52] L. Xu, H. Dinh, E. Zhang, Z. Lin, and R. Laramée, "A distribution-based approach to tracking points in velocity vector fields," in *Proc. Comput. Vision Pattern Recog.*, 2009, pp. 2663–2670.
- [53] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 791–800.



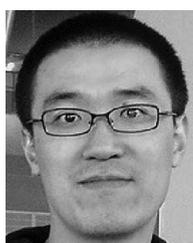
**Seokyeon Kim** received the BS degree in computer engineering from Sejong University, South Korea, in 2014. He is currently working toward the PhD degree at Sejong University. His research interests include computer graphics, data visualization, and volume rendering.



**Seongmin Jeong** received the BS degree in computer engineering from Sejong University, South Korea, in 2016. He is currently working toward the MS degree at Sejong University. His research interests include flow map visualization and visual analytics.



**Insoo Woo** received the BS degree in computer engineering from Dong-A University, South Korea, in 1998 and the PhD degree from the School of Electrical and Computer Engineering, Purdue University, in December 2012. He is currently a graphics software engineer with Intel Corporation. Prior this, he was employed as system software engineer with NVIDIA. His research interest include GPU-aided techniques for computer graphics and visualization.



**Yun Jang** received the bachelors degree in electrical engineering from Seoul National University, South Korea, in 2000, and the masters and doctoral degrees in electrical and computer engineering from Purdue University, in 2002 and 2007, respectively. He is an associate professor of computer engineering with Sejong University, Seoul, South Korea. He was a postdoctoral researcher with CSCS and ETH Zürich, Switzerland, from 2007 to 2011. His research interests include interactive visualization, volume rendering, visual

analytics, and data representations with functions. He is a member of the IEEE.



**Ross Maciejewski** is an assistant professor with Arizona State University. His primary research interests include the areas of geographical visualization and visual analytics focusing on public health, dietary analysis, social media, and criminal incident reports. He is a fellow of the Global Security Initiative at ASU and the recipient of an NSF Career Award (2014).



**David S. Ebert** received the PhD degree in computer science from Ohio State University. He is a professor in the School of Electrical and Computer Engineering, Purdue University, a University faculty scholar, director of Purdue University Rendering and Perceptualization Lab, and director of the Purdue University Regional Visualization and Analytics Center. His research interests include novel visualization techniques, visual analytics, volume rendering, information visualization, perceptually based visualization, illustrative visualization, and procedural abstraction of complex, massive data. He is a fellow of the IEEE and the IEEE Computer Society, and a member of the IEEE Board of Governors.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).