# Adaptively Exploring Population Mobility Patterns in Flow Visualization

Fei Wang, Wei Chen, Ye Zhao, Tianyu Gu, Siyuan Gao, and Hujun Bao

***Abstract*—Thanks to the ubiquitous cell phone use, we have never been so close to uncover population mobility patterns in urban area. While some researches utilize cellphone call records to mine population patterns, few works aim to depict population movement in adaptively spatial and temporal representations, i.e., from a community, a district in the city over an hour, a day to a week. In this paper, we construct a system which deciphers, transforms, queries and visualizes the records from millions of users in a city. In particular, we design a data structure, namely MobiHash, which collects phone call records over base stations and indexes them by utilizing a Voronoi division of the urban space. MobiHash supports responsive data queries so that users can interactively retrieve trajectories reflecting population flows in areas of interest. Moreover, population movement are represented as vector fields to reduce visual clutter and occlusions. Because of sparse moving points, a novel radiation model is proposed to interpolate population passing zones. Case studies and experts' feedback validate the utility and efficiency by comparing population moving patterns in different times by using our system.**

***Index Terms*—Population Mobility Pattern, Visual Query, Flow Visualization, Cell Phone Data.**

## I. Introduction

A Population mobility pattern (e.g., the gathering and scattering pattern) informally represents a event or incident that involves a large group of people, which form durable and stable areas with a comparable high or low density. Population mobility patterns are well studied in many aspects, for example, predicting non-trivial group incidents (e.g., traffic jams, earth quakes, public gatherings) [1], modeling transport services [2]–[4], analyzing functional areas of a city [5], and even designing and evaluating mobile applications.

The rapid expansion of wireless infrastructure in recent years has offered opportunities to record population mobility behaviors. In this paper, we utilize a real big data set of phone call records with base station information in a city to design a system for a study of city-wide population mobility. Our data comprises both the city's base stations and movement of cell phone holders among these stations. Therefore, our data reflects spatial locations of cell phones as well as human trajectories which reflecting urban population dynamics.

Displays showing a large amount of trajectories may suffer from visual clutter and occlusions. Simplifying or aggregating the spatial data are widely used to reduce visual clutter by decreasing the opacity of minor flows [6] or the size of the represented data [7], [8]. Density fields with different radii kernels can be combined to expose simultaneously large-scale patterns and fine features [9], [10]. These approaches mainly present global patterns of movement, however, local details which impact traffic situations may be omitted. Vector field techniques used in fluid dynamics have been well employed in migration and general movement mapping [11], [12] because smooth and continuous lines of vector fields are able to outline movement details and specific topological patterns such as sources, sinks and vortexes, etc.

In this paper, the idea is utilizing vector fields to adaptively represent the mobility patterns of population in a city and in communities or districts inside the city. The vector fields are visualized through flow visualization tools so that mobility behaviors in different sizes of space and time can be identified. For example, if an expert cares about traffic situations on a bridge in one day, s/he can brush the bridge on map and filter data of that day. While s/he wants to know the population movement for a big area in a specific hour, s/he can select the big area (up to the whole city) and filter the data of the specific hour. While the generated raw vector fields have high resolution, by using vector visualization tools with different visual resolutions (e.g., density of streamlines), both detailed movement in the former scenario and global flows in the latter scenario would be presented.

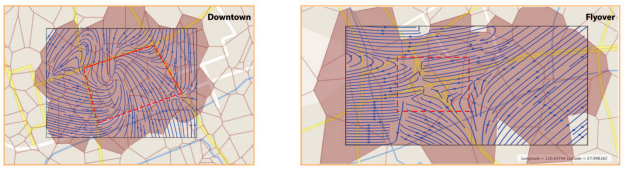


Fig. 1. Left: Crowd congregating into the red square could be detected in the flow visualization; Right: Traffic flows outlining a flyover in the red rectangle could be identified.

The adaptive approach to uncovering population mobility patterns potentially has twofold challenges, i.e., querying the big, spatial and dynamic data sets and visualizing sparse phone records in various space and time. To overcome these challenges, we first design a dynamic spatial data structure MobiHash to manage the calling records on the spatial do-

Fei Wang is with Key Laboratory of High Performance Computing and Stochastic Information Processing (MOE), College of Mathematics and Computer Science, Hunan Normal University, Changsha, P. R. China. He is also with Smart System Institute, National University of Singapore.
E-mail:wolflyecn@gmail.com

Wei Chen (corresponding author), Tianyu Gu, Siyuan Gao, and Hujun Bao are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, P. R. China.
E-mail: {chenwei@cad.zju.edu.cn, bao@cad.zju.edu.cn}

Ye Zhao is with Department of Computer Science, Kent State University, Kent, Ohio, United States.
E-mail:zhao@cs.kent.edu

mains of the city. In particular, we split the city's space into cells through a Voronoi diagram, given the locations of all the mobile stations. In fact, it provides an effective spatial partition of the city, since the distribution of mobile stations reflects the city's structural and human activity features. For example, it is dense in heavily used area and sparse in suburbs.

Additionally, we provide two visualization techniques to allow users explore data: (1) the dynamic flows of population movement on the map; (2) the statistics of cell phone usage over time in interested city regions. Flows among spatial divisions represent spatial patterns of population movement.

We demonstrate and evaluate the proposed method in real-world applications if applied in intelligent transportation systems. For example, if population congregates at a square during a short time, traffic situation forecasting should be reported immediately to avoid dangerous events from congestions, e.g., 2014 Shaihai stampede. Map errors (missing roads, junctions, etc.) are great challenges in automatic generation of road network map [13]. The flow detecting application considers flows variation to present local situations of traffic sensitive areas. People in all directions crowding into the red square of Fig. 1(left) could be detected by our flow visualization. In the field of road network map, we address similarity between flows and the road network. Fig. 1(right) displays a flow visualization, whose outline in the red rectangle well matches a flyover on map.

We summarize our main contributions as follows:

- Designing new representation and visualization techniques to utilize the cell-phone-data-based movement data for exploring population mobility patterns;
- Dividing the city domain by Voronoi diagram based on real-world base station locations, so that the data is efficiently managed via a spatial structure;
- Utilizing an efficient data structure, MobiHash, to offer immediately data query.

The rest of this paper is organized as follows. Related work is summarized in Section II. Then we introduce our data and system architecture in Section III. Section IV gives definitions for basic concepts and describes how to manage movement data with MobiHash index. The proposed interpolation method is given in Section V and our visual interface is described in Section VI. We present case studies and experts' feedbacks in Section VII, followed by a conclusion and future work in Section VIII.

## II. Related Work

Existing approaches to discovering population mobility patterns cover three categories: statistical methods [14], [15], data-mining methods [16], and visual analytics methods [8], [17]. Statistical methods and data-mining methods are typically used to investigate specific mobility models in a city. Numerous data mining methods have also been proposed for the study of human mobility, e.g., modeling cellphone users movement; predicting where cellphone users will travel next; and identifying cell-phone users' important locations. Most works relevant to our approach locate in visual analytics and data management areas.

### A. Visual Analytics of Movement

A wide variety of studies in transportation and visual analytics for analysis of movement data have been proposed. A recent survey summarizes existing visualization methods for traffic data analysis [18]. Global and local traffic patterns are uncovered and traffic situations behind the patterns are analyzed, such as major traffic routes detection [19], traffic jams analysis [20], and tidal flows reasoning [21]. Andrienko et.al. [22] review existing methods, tools, and procedures and present an illustrated structured survey of the state of the art concerning the analysis of movement data. In particular, AllAboard [23] uses cellphone data to support visually exploring urban mobility for transit optimization. Wang et al. [24] study human travel patterns on mobile phone records. Both works employ many visualization technics, such as OD maps, density maps, vector field maps and particle graphs to complete different tasks.

Movement between regions can be represented as a graph, where regions are graph nodes and flows between them are treated as weighted directed edges [25]. Displays showing multiple trajectories may suffer from visual clutter and occlusions. Simplifying or aggregating techniques are proposed to reduce visual clutter [6]–[8], [26]. Edge bundling groups spatially close trajectories [27], [28] to show origin-to-destination movement.

While the graph model is widely employed, density fields built using kernels with different radii can be combined into one field to expose simultaneously large-scale patterns and fine features. Willems et al. [9] propose a specific kernel density estimation method for trajectories, which interpolates trajectory points involving the speed and acceleration. Scheepens et al. [10] further process multiple density field and suggest a scripting-based architecture for creation, transformation, combination, and enhancement of movement density fields.

Flow maps modeling movement as vector fields have been well employed in migration and general movement mapping [11], [12], [29]. Researchers on trajectory clustering analysis utilize flow visualization to show speed and direction of movement [30]. Related problems of deriving flow datasets have been studied based on computer vision and geometric algorithms [31], [32]. Although we employ the vector field model, we point out that their contexts are different because our dataset is irregular and experts need to observe data adaptively.

### B. Data Models for Query

Researchers have used various methods to organize and query trajectory data. Ferreira et al. [33] propose a visualized tool to query trip records with spatial, temporal and other user-specified attributes. Nanocubes [34] can also be used to reveal real-time spatio-temporal data sets. However, those methods are developed for large region origin-destination (OD) queries. SemanticTraj [35] textulizes trajectories into documents and employs text search engine to query with semantic hints.

Moreover, using approximations as minimum bounding rectangles (MBRs), octagons and regular grid cells are widely used in indexing spatial-temporal data. At the cost of more

computational resources, high resolution of spatial grid can provide better matches on querying trajectories over spatial division. Multidimensional indexing methods like 3D R-Tree [36] and HR-Tree [37] are popular approaches to trajectory indexing. On the other hand, systems like SETI [38] and TrajStore [39] notably reduce the size of bounding boxes for large trajectories through building time index for divided grids.

In this paper, we propose a novel data model to topologically link phone records in flows and could support querying data in unequal time spans.

## III. Data and System Overview

### A. Cell-phone Call Data

With the help of our collaborator, one of China mobile communication service companies, we have an opportunity to study cell-phone data from a city in China from December 1, 2013 to March 31, 2014. The city is covered by 28,000 base stations serving for 9 million 18-64 years old users (China National Bureau of Statistics, 2010). Therefore, the available data are sufficient to represent the main population mobility patterns of the city. Even though cell phone's locations are inaccurate, base station's locations are always accurate and we can infer the covering region of a base station. Thus, a base station is regarded as a minimum observation unit to distinguish a user's position, called a cell. And two kinds of data, namely, call detail record (CDR) and cell detail list (CDL), are studied in this work.

**Call Detail Record (CDR)** When a mobile phone holder calling in a cell, entering or leaving a cell, a CDR will be stored. A CDR comprises *phone ID*, *station ID*, *record time*, and *state*, i.e. <*phone_id, station_id, time, state*>, where the *state* attribute indicates a user's behavior, i.e. entering or leaving a station. The size of CDR is very large, and more than 100 million records are generated in one day. Moreover, the granularity of CDR differs severely both in time and in space because some citizens call frequently in movement while others do not call even during a day.

**Cell Detail List (CDL)** A CDL record lists the cell related information, formally <*cell_id, cell_latitude, cell_longitude, direction, city_id*>.

For the mobile network, mobile devices usually connect to the closest station regarding the strength of signal [40]. Due to the Voronoi tessellation rules, base stations from CDL are specified as seeds to generate a Voronoi diagram. The city is partitioned into Voronoi cells based on distances to seeds (see Fig.1). A interesting pattern is that base stations are more dense in the downtown than in the countryside, which is consistent with the population density.

To further verify the reasonability of the space division, we aggregate phone records on one day in cells of one functional zone. We find that the data size of each cell of the functional zone is almost equivalent, therefore, we call the Voronoi tessellation an adaptive division of the city in space domain.

### B. System Architecture and Overview

Our system's architecture, shown in Fig. 2 consists of two components: a user interface for data visualization and interaction, and a data service part for data management, query processing and data analysis. Firstly, experts select relevant time and region items. The data service interface receives those settings and dispatch them to a trip query module. The trip query module transforms settings into query constraints and execute querying on a distributed data management module via our novel index. Then the querying results are analyzed and encoded into graphic nodes to transfer into the visualization module, which will display the graphic nodes to represent mobility patterns. Experts can investigate views and carry on iterative interactions.

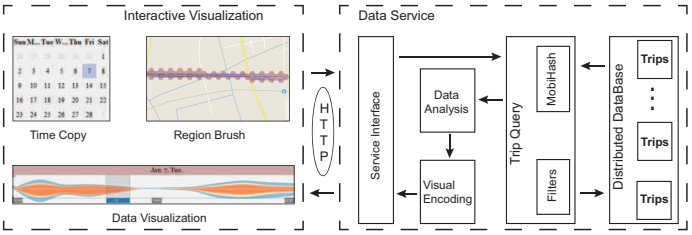


Fig. 2. System overview. Our system includes two parts: Interactive Visualization and Data Service.


Fig. 3. User interface. (a) Time copy widget; (b) Flow visualization in streamlines; (c) Interaction tools; (d) Temporal stack graph

**Interactive Visualization**. Mobility patterns in temporal and spatial spaces are visualized in two fashions. A flow visualization (See Fig. 3(b)) is employed to accentuate people movement characters on map, e.g., moving directions, moving regions, and numbers of moving populations. Trips are transformed into vector fields not only to highlight population movement but also to reduce clutter and occlusions. On the other hand, trends of moving population are depicted in a stack graph (See Fig. 3(d)). For interactive query, a temporal filter (See Fig. 3(a)) and a spatial filter (See Fig. 3(c)) are introduced in Section V and VI.

**Data Service**. Our sample CDR data exceeds 25 GB in one day. For the trip query module, it needs transforming settings from query widgets into query constraints, and searching trips from the massive amount of CDR data as well. Therefore, it still needs to shape an efficient data model to manage the big data of 4 months.

Three measures are carried out: 1) a population movement data model is abstracted from the sparse and irregular cell-phone records; 2) a bi-directional linking hash index is built to find flows under query constraints within hundreds of mil-

liseconds; 3) a distributed column-based in-memory database is employed to store and retrieve the original data.

## IV. Identifying Population Movement and Managing Flows

We identify cell phone users who are moving from one cell to another and model movement in one day as trips. Flows are managed by a hash index to support efficient queries over possibly selected cells and time slots.

### A. Definitions

The major challenge when identifying movement in mobile phone data lies in the sparse and irregular records, where user displacements (consecutive distinct recorded locations) are usually observed in a long period (e.g., the first location is observed at 8:00 am and next location is observed at 6:00 pm). To extract population movement more accurately, we only record displacements occurring within an appropriate time window. Formally, population mobility items can be defined as follows.

**Definition 1 (Population Movement)** *Population Movement* means a group of objects $M = o_1, o_2, \cdots, o_k$ shifting together from one place to another, where $k > N$ and $N$ is a parameter (default as 15 based on the characteristics of our data set).

**Definition 2 (Movement Point)** A *Movement Point* $p$ is a four tuple, i.e., $p$ = (*o, loc, t, s*), where each point consists of a moving object id, a geospatial coordinate set, a timestamp, and a state.

**Definition 3 (Trip)** A *Trip* $T$ is a sequence of time-series movement points, which records consecutive locations and states of a moving object. Regarding population movement in real life, We assume people move from one station to another station in a short period (60 minutes), following an observation from [4]. A flow can be described as $T$: $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, where $p_{i+1}.t - p_i.t < 60$ minutes and $p_i.loc \neq p_{i+1}.loc$. Therefore, population mobility means a group of objects passing the same stations during a time interval.

### B. Identifying Population Movement

As stated before, CDRs are irregular both in time and in space. Time intervals between consecutive CDRs varies severely from a few seconds to several days. A two-phase approach is taken to identify a movement. In the first phase, points frequently (<6 seconds) switching between 2 stations, called ping-pong handover in telecommunication, are removed from the raw data. In the second phase, points that don't satisfy the definition of a trip, i.e. $p_{i+1}.t - p_i.t < 60$ minutes and $p_i.loc \neq p_{i+1}.loc$, would be removed because they are not movement points of a trip. Points that are beyond the city area are also removed regarding the station's locations which they belong to.

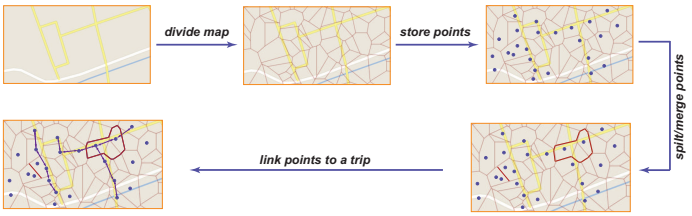


Fig. 4. The 4-step procedure of building the MobiHash index.

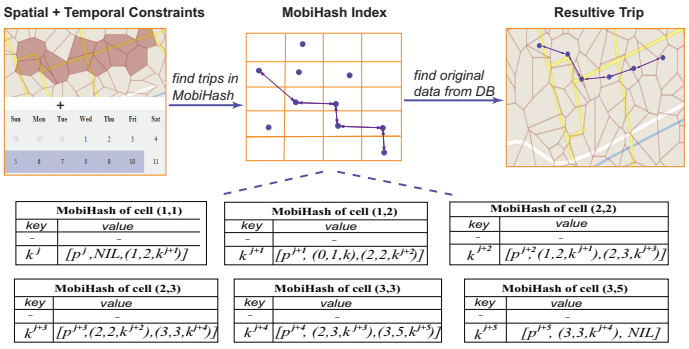


Fig. 5. The procedure of querying with the MobiHash index.

### C. Query Population Trips with Index

To look for a subset of records from a large dataset, using a proper index will reduce the time of searching dramatically.

**Building Index**

The grid-file [41] index splits a space evenly into a grid where each cell of the grid refers to a small set of points. Similarly, we split the data set in a space-time cube, whereas MobiHash subdivides the map into Voronoi cells. A labeled bucket is related to one cell and stores a collection of points located in the cell. If points in a bucket exceed the threshold the bucket will split, on the contrary, neighbor buckets will merge together. MobiHash indexes these records in buckets labeled with a cell's id and a time slot. Consecutive points of a trip connect sequently in two directions in order to accelerate finding a trip. To explain simply, Fig. 4 presents the index's building process which is only divided in the spatial domain.

After a 4-step procedure, MobiHash can be presented as a hierarchical container (see Fig. 5). From the top level to the bottom level, they are Voronoi cells, time slots, buckets respectively. And points are stored as key-value data and retrieved via the key as a hash code. To reduce collisions, the cell's id, the moving object's id are merged to generate a unique 32-bit code in a bucket. Meanwhile, the value includes five elements, i.e. a point's id, last cell's coordinates, next cell's coordinates, the key of the last point, the key of the next point.

**Querying Trips**

Fig. 5 depicts the procedure of querying in three steps: input query constraints, search trips in the index and query the original points in database. Once experts manipulate the spatial and temporal widgets, the values are transformed into multiple query ranges. These ranges constrain searching spaces while traversing in the index. The time ranges can be a single

time slot or multiple periodic time slots, such as from 6:00 am to 10:00 pm on every Monday in January, 2014.

Scanning a bucket and traversing a trip will be conducted when searching in the index. While a movement point is located in the query ranges, neighbour points of the same trajectory will be checked. Those neighbour points are found via the last cell's coordinates, the next cell's coordinates and the keys of the last point and the next point. As shown in Fig. 5, if Point $k^{j+1}$ is found, a cursor will move forward and backward to find records with the same id until points exceed the query ranges.

Once finding trips, points' ids will be returned which are used to retrieve the original data. Retrieving original data is vital to experts because original data support them changing their requirements in an iterative exploration. In order to reduce data retrieval time, a column-based in-memory database, MonetDB, is employed to store the CDR data. We marshal the data on 5 machines in horizontal partitioning regarding the time. Each machine has 22 tables where each table stores data of 1 day, i.e., totally storing data of $(22 \times 5) = 110$ days.

### D. Discussion

The *MobiHash* index is an extension of our previous work [21] where we made a benchmark and our index outperforms 3D R-Tree and SETI. The complexity of the query is $\mathcal{O}(c_1 * c_2 * m) + \mathcal{O}(1) = \mathcal{O}(m+1)$, where $m$ is the number of points in a bucket and $c_1$ is the number of cells, and $c_2$ is the length of a trajectory segment, and both $c_1$ and $c_1$ are small constants. Since the data is split in a time-space cube, we can adjust the size of 3-dimension cell such that there is no more than $10^5$ points in one bucket. For example, we set the time interval of a bucket as 24 hours. To reduce the size of the index, MurmurHash3 is used to generate 32-bit hash values such that the size of a key-value record is no more than 180 bytes.

## V. A K-Means Radiation Model for Vector Fields

People move randomly among base stations, whose trips form a complicated network. On the other hand, our data are sparse for far distance trips. If painting many criss-cross lines on the map directly, visual clutter and occlusions will be severe and impede interactivity. In this section, we introduce a novel interpolation model to transform trips into vector fields.

### A. K-Means Radiation Model

To investigating population movement, experts often focus on main routes of trips and their effects on transportation. With respect to this assumption, we only consider several main directions, although people moving towards diverse destination stations. We use the k-Means algorithm to cluster similar directions to form major vectors. A major vector radiates from the station's center to its neighbor zones.

Given a movement network in a time interval, the procedure of vector field generation can be concluded as four steps (see Fig. 6): 1) each movement from one station to another is recorded. 2) we calculate a vector of each station regarding every movement and then find several major vectors of each station (Section V.B). 3) we subdivide the selected region into quad cells evenly. 4) a synthetic vector field is generated according to major vectors, which approximately describes the backbone of the movement network (Section V.C).

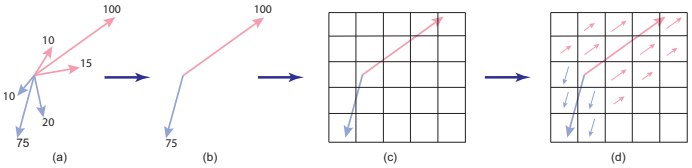


Fig. 6. The procedure of vector field generation.(a) calculating movement in stations, (b) clustering major vectors in k-Means, (c) subdividing the selected region evenly, (d) synthesizing vector fields from major vectors.

### B. Basic Vectors and Major Vectors

**Definition 4 (Movement Intensity)** A *movement intensity* denotes how many objects move from a movement point $p_i$ to another one $p_j$ at a time interval, which can be described as a 3-tuple, i.e., $< \iota, \tau, \ell >$, where $\iota$ is a departure station, $\tau$ is a destination station, and $\ell$ is the number of population as an intensity. Two vectors, namely basic vectors, in a movement can be calculated: a fan-in vector of a destination station; a fan-out vector of a departure station.

As people in a station moving forward to distinct stations, there are many basic vectors in a station. However, we accentuate population mobility patterns, therefore we only consider movement involving a group of people. We cluster basic vectors into several major vectors according to vectors' angles and intensities via k-Means algorithm (k=3). Then, each major vector is normalized to represent the direction of a group of people movement and each major vector is related to an intensity. The major vectors express a global vision of population movement.

### C. Synthetic Vector Field

We developed a method to reconstruct a synthetic vector field in neighbor regions of a station from its major vectors. The method is based on two observations in real life, which is consistent with population flows:

1) people in a small area tend to have similar moving direction under an effect of a vector;

2) people who gather or scatter in a local area form special topological structures such as sources, sinks and vortexes, etc.

To derive vectors in the area around a station, we subdivide the selected region into quad cells. The size of a cell is set to ensure different stations in different cells. For a major vector $v_i$ (a vector at station $p_i$), let $q$ be any quad cell around $p_i$. To present the special topological structures, we differentiate two vectors which represent flows entering a station and those leaving. Suppose the angle between the direction from $p_i$ to $q$ and $v_i$ is $\theta$. For the leaving vectors, we only take in account quad cells ($q$) if $\theta$ is less than $60^o$. Meanwhile, we take in account quad cells if $\theta$ is larger than $120^o$ for the entering vectors. The synthetic vector at a quad cell $q$ is determined as follows.

$$v(q)=\frac{v_i}{\|v_{p_i,q}\|},$$
$$\|v_{p_i,q}\|=(f(\theta_{v_{p_i,q},v_i})+g(\|v_{p_i,q}\|))*\ell$$

where f and g are quadratic decay functions, and $\theta$ is an angle between $v(q)$ and $v_i$. The equations described the truth that the intensity at $p$ reduces as the deviation of $q$ from $p_i$ and of $v_{p_i,q}$ from $v_i$.

As major vectors have different intensities, their effects may overlap in some areas. Computing the average orientation of vector and rotations is non-trivial due to the need to account for the periodicity of rotation. Olson [42] provides a new geometric interpretation and produces significantly more accurate results for computing average orientation using the squared arc-length algorithm.

To put it in a nutshell, the interpolation is depicted in algorithms in Appendix.

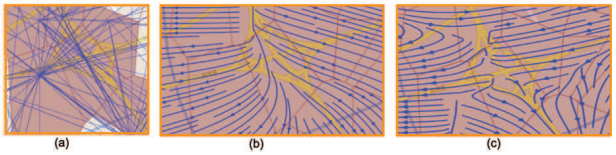


Fig. 7. Visualizing trips in different vector field techniques. (a) The original trips on an overpass. (b) Vector Field k-Means Fitting outlines the main flows. (c) Our approach displays the detailed flows.

### D. Discussion

Due to perceptual and cognitive limitations and performance issues, visually analysing trips is difficult [43]. In addition, the connection between stations will deviate from the real situation a lot if there exists some loss or incompleteness on trips. There are several potential approaches to visualize trips as follows.

**Edge Bundling.** To bind or cluster edges in the same direction and small distance is one of the most important approaches to reduce visual cluttering. These methods, however, are only suitable for static graphs. The number of trips is presented very well, but directions and speeds disappear in edge-bundling views. Therefore, edge bundling is not suitable for visualizing population flows because they change over time.

**Topological Reconstruction.** A variety of methods, such as feature-based graph [8] and kernel density estimation [44], have been employed to simplify large-scale graphs and then visualize them through traditional methods. Howbeit, such methods usually convey a global vision of movement. In fact, experts not only overview global movement but investigate local details to find how people gather or scatter over time.

**Vector Field k-Means Fitting.** Vector Field k-Means Fitting (VFKM) [32] use vector fields to cluster trips by similarity of trips. Each trip would be approximately tangent to one of the fitting vector fields. For population trips between stations, they are spatially sparse and are irregular in time. Moreover, people moving directions may be totally distinct. Fitting such trips in vector fields is almost infeasible. In Fig. 7 our method displays detailed flows passing an overpass. Meanwhile, VFKM only outlines the main flows because of its approximate computation.

**Multi-step Visualization.** Multi-step visualization employs many technics, such as OD maps, density maps, vector field maps and particle graphs to complete different tasks. Although it has been well applied in exploring urban mobility [23] and human travel patterns [24], switching between different views costs much human physical memory during interactive exploration. We propose a novel interpolation model on sparse mobile phone data to fuse a vector field map and a heatmap in one view and allow experts to compare views of different time intervals.

## VI. Visual Design and User Interface

In this section we describe the visual design and the user interface (see Fig. 3). To facilitate operating the user interface, a three-step interaction is introduced in Fig. 3(c), i.e., selecting time spans, brushing a region, and checking views of query results. While checking views, a lane on the stack graph is dragged to compare flows in different hours. If not satisfied, the view can be removed, and a new interaction will be started from the first step.

### A. Visual Design

To explore the mobility data, we design a flow visualization and a stack graph presenting patterns distributing on map and changing over time, respectively.

**Flow Visualization.** Texture-like methods, such as Spot Noise [45] and LIC [46] produce dense field images showing the flow features in fine-grain detail. However, we employ a streamline method to show population mobility patterns on map because of a straightforward direction cue and low computational cost.

In Fig. 3(b), lines denote population movement among stations, where arrows indicate the direction. Behind the flow, a heatmap displays the density of population on map in a time interval.

**Temporal Stack Graph.** The basic idea of the temporal view is not only displaying how population flows vary over time, but also comparing population flows at different speeds.

After brushing the calendar and Vonoroi cells, a stack graph shows how the flows change over time, from which the trend of population movement will be identified (see Fig. 3(d)). The horizontal axis indicates selected time period. The vertical axis indicates the number of people. Three colors are used to encode flows in different speed ranges. A color ribbon indicates the number of people within a corresponding speed range over time. Legends are displayed under the horizontal axis to indicate time slots. A lane is designed to allow experts to select time slots and investigate a flow visualization.

### B. Time Copy Widget

Calendars, timelines and time lists provide single time selection methods, however selecting multiple time intervals is quite inconvenient because it needs many operations. Time copy widget (see Fig. 3(a)) is designed to reduce the number of operations. The widget comprises two components: a calendar and a time slider. First, Experts brush dates on the calendar to

select single or multiple dates, such as five dates of a week, the same days in a month, and several random dates, etc. Then they can drag the time slider in two directions to determine a time span of interest. Later, the start and ending timestamps are listed on both sides of the slider. Thus, multiple time slots can usually be selected within only two operations.

### C. Region Brushing Tool

Experts usually want to select regions (e.g., paths, districts or streets) to investigate variations of flows. A region brushing tool is provided to select appropriate cells on map. The main problem lies in how to evaluate which cell the brush locates in. It is time-consuming to evaluate in brute force. We preprocess relations between rectangles and Voronoi cells to reduce the interaction time. Above all, the map is divided evenly and each rectangle includes several Voronoi cells. When a brushing starts, we will figure out which rectangle a brushing point locates in.

## VII. Case Studies and Experts' Feedback

In this section, we present two case studies, using real data, where our proposed methodology is used to explore urban data. The original data size is about 2.4T in text format. Since only considering moving populations, we remove static points and points swapping in neighbor stations. After preprocessing, data are stored in MonetDB distributed on 5 machines. To show that our system can effectively discover the population mobility patterns, two cases are described below.

### A. Population movement scenario at downtown

Due to the high density of populations and buildings, the analysis of population flows in downtown mainly relies on manual operation. Fig. 8 displays the population mobility patterns of different times obtained by brushing downtown area and applying the time copy widgets.

Feb. 4, 2014 is the 5th day of the Chinese Spring Festival. In the temporal view, population kept steady till 9:00 am. During holidays citizens sleep longer and go on street later than usual. Furthermore, the population assembled mainly within a few zones around 9:00 am. The heatmap in Fig. 8 represents the number of populations and the arrows represent moving directions. With the increase of populations at 10:00 am, the assembling zones started to enlarge (see Fig. 8(b)). The number of assembling zones increased with the appearance of the peak of population flows about 13:00 pm in Fig. 8(c). It is obvious to infer that the assembling people scattered gradually on 20:00 pm from the direction of the arrows in Fig. 8(d). Till 23:00 pm, the assembling people has dismissed with only few highlight dots shown in the Fig. 8(e).

### B. Traffic situations on a bridge

Dongou bridge connects downtown area and the residential area. It is a 6-lane road and is busy all day. By brushing the bridge covered cell and selecting the day of Feb. 1, flows of each hour show the traffic situations on the bridge. The temporal view shows that the population passing the bridge

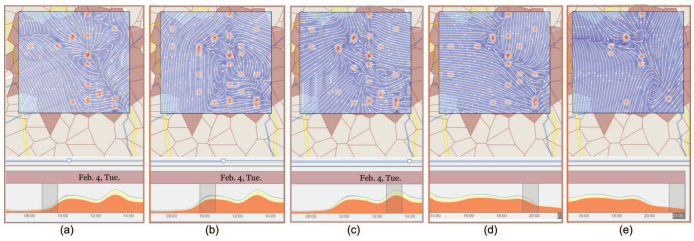


Fig. 8. population mobility patterns at downtown. From (a) to (e), subgraphs show that people gather and scatter at downtown from early morning to midnight.

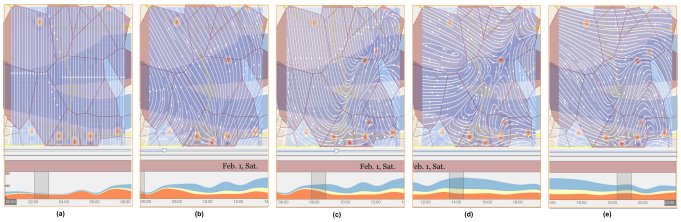


Fig. 9. Traffic situations on a bridge in a day. From (a) to (e), each subgraph indicates a mobility pattern of an hour.

in a high speed ($\geq 20km$) is much larger than population in downtown. At 2:00 am, few people run to and from the bridge, which is displayed by some straight upward and downward arrows on the diagram above (see Fig. 9(a). About 06:00 am, more and more people cross the bridge from the north to the south toward downtown, see Fig. 9(b). Red dots on the south of the bridge indicate vehicles are assembling there. In Fig. 9(c) population flows in two directions increase at 8:00 am. The traffic situation is complicated at 14:00 pm or so with no obvious regularity shown on the diagram (see Fig. 9(d)). In the rush hour, vehicles are departing downtown towards uptown while some might-be tourists set out from the island to downtown (see Fig. 9(e)). There is no crowded traffic phenomenon that appears on the bridge in the day.

### C. Feedback from transportation experts

Following the initial deployment, we invited two transportation experts to use our system. Having discussed the concept of the population flows with them, we wanted to know how they felt the system matched their expectations. They felt that the visualizations *"highlight the population mobility patterns"* [Expert_1] and provide *"a variety of ways to understand the content"* [Expert_2]. Further comments from the experts reveal a difference in how they value the various views within the interface. For Expert_2 who has used similar tools before, *"brushing the map eases the formulation of queries"*. He compared those ways of querying data with how our system extends upon them:

*"In commonly map-based search interfaces, it is difficult to formulate queries that simultaneously combine temporal and spatial constraints in free styles. The system responds more quickly than former systems though the data set is huge."*

For Expert_1, the system holds the most value as:

*"Some interesting patterns are uncovered via flow visualization. It is the first time that we observe traffic flows changing*

*over time on a flyover from real-life data. The patterns are useful for estimating designs of flyovers."*

However, the experts provided negative feedbacks on the interface, which they felt observing traffic flows on roads is not flexible enough, because Voronoi cells cover larger zones than roads. On the other hand, they thought CRD data does not include all transport situations because drivers don't run cars while calling. Asked to speculate about future directions of using visualization to explore the transportation, Expert_1 is considering the *"possibility of integrating other data sets, such as floating car data, public transportation data"*.

## VIII. Conclusion and Future Work

In this work, we have introduced an adaptive exploratory system that aims to make it possible for population mobility patterns seekers to orient themselves in space and over time. Our general design considerations led us to define the system that:

- Provides a flow visualization that can be used to discover population mobility patterns.
- Allows query specification through dynamic manipulation of temporal and spatial ranges.
- Offers responsive results that immediately change according to user's interacctive operations.

To complete the design, we have developed a k-Means radiation model as a novel interpolation to transform population movement into a vector field which is visualized as flows, and MobiHash as a novel mechanism to index trips and reduce loaded data during query refinements. The data is divided by Voronoi diagram based on real-world base station locations. Furthermore, we have implemented brushing as an interaction technique that displays tempo-spatial relatedness between visual elements.

We demonstrate our system via two applications to intelligent transportation systems. Moreover, two real-world exploratory studies earn positive reactions of domain experts towards discovering population mobility patterns with our system. We also invited experts from the transportation domain to comment on our system in practice and they gave us positive feedback.

Although the flow visualization clearly displays patterns of population movement, it is presented in a rectangle view because a vector field is mapped on an image. The mapping will cause some distortion of the flow visualization. We will further develop a mapping from the vector field to the geographic map. In the current realization, our system is limited to cell phone records due to the Voronoi diagram subdivision. As experts suggestion, more data subdivisions and interactive widgets are needed to support multiple data sources. Besides, a performance benchmark of MobiHash needs to be addressed in future, though it satisfies interactive exploration requirements.

## IX. Acknowledgements

We would like to express gratitude to experts from Wuxi Mingda Traffic Technology Consultation Co., Ltd for their active participation and to collaborators to provide valuable data and to anonymous reviewers for comprehensive comments at various stages of drafting. This work is supported by the Construct Program of the Key Discipline in Hunan Province, China and the Scientific Research Fund of Hunan Provincial Education Department (15B137). Wei Chen is supported by National 973 Program of China (2015CB352503), NSFC (61232012, 61422211, and U1609217). This work is partly supported by US NSF grants 1535031 and 1637242.

## References

[1] S. Isaacman, R. Becker, R. Cáceres, et.al., "Human mobility modeling at metropolitan scales," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 239–252.

[2] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, and T. He, "coride: Carpool service with a win-win fare model for large-scale taxicab networks," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '13. New York, NY, USA: ACM, 2013, pp. 9:1–9:14.

[3] D. Zhang, J. Huang, Y. Li, F. Zhang, C. Xu, and T. He, "Exploring human mobility with multi-source data at extremely large metropolitan scales," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 201–212.

[4] P. Wang, T. Hunter, A. M. Bayen, and M. C. Gonza'lez, "Understanding road usage patterns in urban areas," *Scientific Reports*, 2013.

[5] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 186–194.

[6] J. Wood, A. Slingsby, and J. Dykes, "Visualizing the dynamics of london's bicycle-hire scheme," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 46, no. 4, pp. 239–251, 2011.

[7] D. Guo, "Flow mapping and multivariate visualization of large spatial interaction data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1041–1048, 2009.

[8] T. von Landesberger, F. Brodkorb, P. Roskosch, N. Andrienko, G. Andrienko, and A. Kerren, "Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 11–20, Jan 2016.

[9] N. Willems, H. Van De Wetering, and J. J. Van Wijk, "Visualization of vessel movements," in *Computer Graphics Forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 959–966.

[10] R. Scheepens, N. Willems, H. van de Wetering, et.al., "Composite density maps for multivariate trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2518–2527, 2011.

[11] W. R. Tobler, "Experiments in migration mapping by computer," *The American Cartographer*, vol. 14, no. 2, pp. 155–163, 1987.

[12] W. R. Tobler, "A model of geographical movement," *Geographical Analysis*, vol. 13, no. 1, pp. 1–20, 1981.

[13] W. Shi, S. Shen, and Y. Liu, "Automatic generation of road network map from massive gps, vehicle trajectories," in *12th International IEEE Conference on Intelligent Transportation Systems*, Oct 2009, pp. 1–6.

[14] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns," *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[15] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[16] R. Ganti, M. Srivatsa, A. Ranganathan, and J. Han, "Inferring human mobility patterns from taxicab location traces," in *Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, USA: ACM, 2013, pp. 459–468.

[17] Y. Ma, T. Lin, Z. Cao, C. Li, F. Wang, and W. Chen, "Mobility viewer: An eulerian approach for studying urban crowd flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–10, 2015.

[18] W. Chen, F. Guo, and F. Y. Wang, "A survey of traffic data visualization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 2970–2984, Dec 2015.

[19] Z. Wang, T. Ye, M. Lu, X. Yuan, et. al., "Visual exploration of sparse traffic trajectory data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1813–1822, 2014.

[20] Z. Wang, M. Lu, X. Yuan, et. al., "Visual traffic jam analysis based on trajectory data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2159–2168, 2013.
[21] F. Wang, W. Chen, F. Wu, Y. Zhao, et. al., "A visual reasoning approach for data-driven transport assessment on urban roads," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 103–112.
[22] N. Andrienko and G. Andrienko, "Visual analytics of movement: An overview of methods, tools and procedures," *Information Visualization*, vol. 12, no. 1, pp. 3–24, 2013.
[23] G. Di Lorenzo, M. Sbodio, F. Calabrese, et. al., "Allaboard: visual exploration of cellphone mobility data to optimise public transport," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 2, pp. 1036–1050, 2016.
[24] Y. Wang, Z. Li, L. Li, Y. Zhang, J. Hu, J. Zhang, and W. Guo, "Visualization analysis for urban human traveling behavior based on mobile phone data," in *15th COTA International Conference of Transportation Professionals*, 2015.
[25] G. Andrienko, N. Andrienko, P. Bak, D. Keim, and S. Wrobel, *Visual analytics of movement*. Springer Science & Business Media, 2013.
[26] S. Van Den Elzen and J. J. Van Wijk, "Multivariate network exploration and presentation: From detail to overview via selections and aggregations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2310–2319, 2014.
[27] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea, "Skeleton-based edge bundling for graph visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2364–2373, 2011.
[28] D. Holten and J. J. Van Wijk, "Force-directed edge bundling for graph visualization," in *Computer graphics forum*, vol. 28, no. 3. Wiley Online Library, 2009, pp. 983–990.
[29] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd, "Flow map layout," in *Proceedings of the 2005 IEEE Symposium on Information Visualization*, ser. INFOVIS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 29–37.
[30] D. R. Brillinger, H. K. Preisler, A. A. Ager, and J. G. Kie, "An exploratory data analysis (eda) of the paths of moving animals," *Journal of statistical planning and inference*, vol. 122, no. 1, pp. 43–63, 2004.
[31] J. C. Nascimento, M. A. Figueiredo, and J. S. Marques, "Trajectory analysis in natural images using mixtures of vector fields," in *2009 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 4353–4356.
[32] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva, "Vector field k-means: Clustering trajectories by fitting multiple vector fields," in *Computer Graphics Forum*, vol. 32, no. 3pt2. Wiley Online Library, 2013, pp. 201–210.
[33] N. Ferreira and J. e. a. Poco, "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.
[34] L. Lins, J. T. Klosowski, and C. Scheidegger, "Nanocubes for real-time exploration of spatiotemporal datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2456–2465, 2013.
[35] S. Al-Dohuki, Y. Wu, F. Kamw, J. Yang, et. al., "Semantictraj: A new approach to interacting with massive taxi trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 11–20, 2017.
[36] Y. Theodoridis, M. Vazirgiannis, and T. Sellis, "Spatio-temporal indexing for large multimedia applications," in *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, 1996, pp. 441–448.
[37] M. A. Nascimento and J. R. Silva, "Towards historical r-trees," in *Proceedings of the 1998 ACM symposium on Applied Computing*. ACM, 1998, pp. 235–240.
[38] V. P. Chakka, A. C. Everspaugh, and J. M. Patel, "Indexing large trajectory data sets with seti," *Ann Arbor*, vol. 1001, no. 48109-2122, p. 12, 2003.
[39] P. Cudre-Mauroux, E. Wu, and S. Madden, "Trajstore: An adaptive storage system for very large trajectory data sets," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. IEEE, 2010, pp. 109–120.
[40] A. Sevtsuk and C. Ratti, "Does urban mobility have a daily routine? learning from the aggregate data of mobile networks," *Journal of Urban Technology*, vol. 17, no. 1, pp. 41–60, 2010.
[41] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: An adaptable, symmetric multikey file structure," *ACM Transactions on Database Systems (TODS)*, vol. 9, no. 1, pp. 38–71, 1984.
[42] E. Olson, "On computing the average orientation of vectors and lines," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3869–3874.
[43] K. Vrotsou, H. Janetzko, C. Navarra, G. Fuchs, D. Spretke, F. Mansmann, N. Andrienko, and G. Andrienko, "Simplifly: A methodology for simplification and thematic enhancement of trajectories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 1, pp. 107–121, Jan 2015.
[44] O. D. Lampe and H. Hauser, "Interactive visualization of streaming data with kernel density estimation," in *2011 IEEE Pacific Visualization Symposium*. IEEE, 2011, pp. 171–178.
[45] J. J. Van Wijk, "Spot noise texture synthesis for data visualization," *ACM Siggraph Computer Graphics*, vol. 25, no. 4, pp. 309–318, 1991.
[46] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 1993, pp. 263–270.

**Fei Wang** received the Ph.D. degree in Computer Science and Technology from Zhejiang University, China, in 2016. His research interests include information visualization and visual analytics.

**Wei Chen** is a professor in State Key Lab of Computer Aided Design and Computer Graphics at Zhejiang University, P.R.China. From June 2000 to June 2002, he was a joint Ph.D student in Fraunhofer Institute for Graphics, Darmstadt, Germany and received his Ph.D degree in July 2002. His Ph.D advisors were Prof.Qunsheng Peng, and Prof.Georgios Sakas. From July. 2006 to Sep. 2008, Dr. Wei Chen was a visiting scholar at Purdue University, working in PURPL with Prof.David S. Ebert. In December 2009, Dr.Wei Chen was promoted as a full professor of Zhejiang University. He has performed research in visualization and visual analysis and published 27 IEEE/ACM Transactions and IEEE VIS papers. His current research interests include visualization, visual analytics and biomedical image computing.

**Ye Zhao** received B.S. and M.S. degrees in computer science from the Tsinghua University of China in 1997 and 2000. He further received his PhD degree in computer science from the Stony Brook University in 2006. He is an associate professor in the Department of Computer Science at the Kent State University, Ohio, USA. He received Google Faculty Research Award in 2011. In Oct 2015, he was honored as Scholar of the Month by KSU. His current research projects include visual analytics of urban transportation data, multidimensional, text, and animated information visualization, patient-specific computational hemodynamics modeling, etc.

**Tianyu Gu** received the B.E.E. degree from Zhejiang University, China, in 2016. His research interests include quantitative social science and visual analytics.

**Siyuan Gao** received the B.S. degree from Zhejiang University, China, in 2016. He is a Ph.D. student of Yale University. His research focuses on a broad aspects including machine learning, data visualization and biomedical imaging.

**Hujun Bao** is a professor in the State Key Laboratory of Computer Aided Design and Computer Graphics and college of Computer Science and Technology, Zhejiang University. He received a B.Sc. degree in mathematics and Ph.D. degree in applied mathematics from Zhejiang University in 1987 and 1993 respectively. Prof. Bao lead the 3D graphics computing group in the lab, which mainly makes researches on geometry computing, 3D visual computing, real-time rendering, and their applications. His research goal is to investigate the fundamental theories and algorithms to achieve good visual perception for interactive digital environments, and develop related systems.

## APPENDIX

In this section, we introduce the procedure of interpolation in 3 algorithms. Algorithm 1 illustrates main steps, i.e., calculating movement intensities (line 2 - 5), finding major vectors, and synthesizing a vector field. Algorithm 2 presents how to construct major vectors at each station. We consider both population entering and leaving, and only movement intensities greater than 15 are population movement (line 4 - 9). Here we set 15 as default, but it can be changed by experts regarding their requirements. We use k-Means (k = 3) to cluster vectors in one station as main vectors. Algorithm 3 describes how to form a vector field from main vectors. As noted in Section V, main routes will impact their neighbor zones. We first divide the brushed region into rectangle cells (line 1). Then, we calculate a main vector's impacting zones regrading angles ($<$ 60) between the main vector and the vector from rectangle cells to the station (line 3 - 10).

**algorithm 1** Interpolation Using a k-Means Radiation Model

**Input:** $\Delta = \{T\}$, // a trip set
*bbx*, // the geo-location bounding box of selected stations
$(\omega, \hbar)$, // the grid size
$\kappa$ // the number of cluster
**Output:** a vector field of $\omega \times \hbar$
1: $I \leftarrow \{\}$ // initialize the movement intensity set
// $s_{ij}$ is a movement intensity from station $i$ to station $j$
2: **for** each trip $T$ in $\Delta$ **do**
3: calculate every movement intensity $s_{ij}$ of each station
4: add $s_{ij}$ into $I$
5: **end for**
6: [$V^{in}$, $V^{out}$]←find_major_vectors($I$), $S$, $\kappa$)
7: $VF$←synthesize_vector_field($V^{in}$, $V^{out}$, *bbx* $(\omega, \hbar)$)

**algorithm 2** find_major_vectors

**Input:** $I$, // intensity set
$S$, // ID set of stations
$\kappa$ // the number of cluster
**Output:** $V^{in}$ and $V^{out}$
// determine basic vectors regarding movement
1: **for** each station k **do**
2: add $\lambda_k^{in}$ into $V^{in}$, add $\lambda_k^{out}$ into $V^{out}$
3: **end for**
4: **for** each intensity $s_{ij}$ in $I$ **do**
5: **if** $s_{ij}.ell \geq 15$ **then**
6: calculate a vector $\alpha$ regarding station i and j
7: add $\alpha$ into $\lambda_i^{out}$, add $\alpha$ into $\lambda_j^{in}$
8: **end if**
9: **end for**
10: find major vectors via $\kappa$-Means of $\lambda_i^{in}$ and $\lambda_i^{out}$, $\kappa$ = 3
11: return $V^{in}$, $V^{out}$

**algorithm 3** synthesize_vector_field

**Input:** $V^{in}$, $V^{out}$, *bbx*
$(\omega, \hbar)$, // the grid size
**Output:** $VF$, // a vector field of $\omega \times \hbar$
1: divide the region into $\omega \times \hbar$ cells via *bbx*
2: initialize $\omega \times \hbar$ arrays, $VF$, $\Lambda$
// radiate vectors from major vectors
3: **for** each major vector $\nu$ in $V^{in}$ and $V^{out}$ **do**
4: **for** each quad cell $q$ around $p_i$ **do**
5: **if** ($\nu$ is a move_out vector $\wedge$ $\cos(\theta) > 0.5$)
|| ($\nu$ is a move_in vector $\wedge$ $\cos(\theta) < -0.5$) **then**
6: $v_q \leftarrow \nu / \|v_{p_i q}\|$
7: **end if**
8: add $v_q$ into $\Lambda_{i+q}$
9: **end for**
10: **end for**
// calculate vectors at overlapping quad cells
11: **for** each $\Lambda_i$ **do**
12: calculate a mean value $\varpi$ using the squared arc-length algorithm
13: $VF_i \leftarrow \varpi$
14: **end for**
15: return $VF$