Visual Fidelity Guaranteed Sampling for Large Trajectory Data Visualization

Category: Research

Paper Type: algorithm/technique

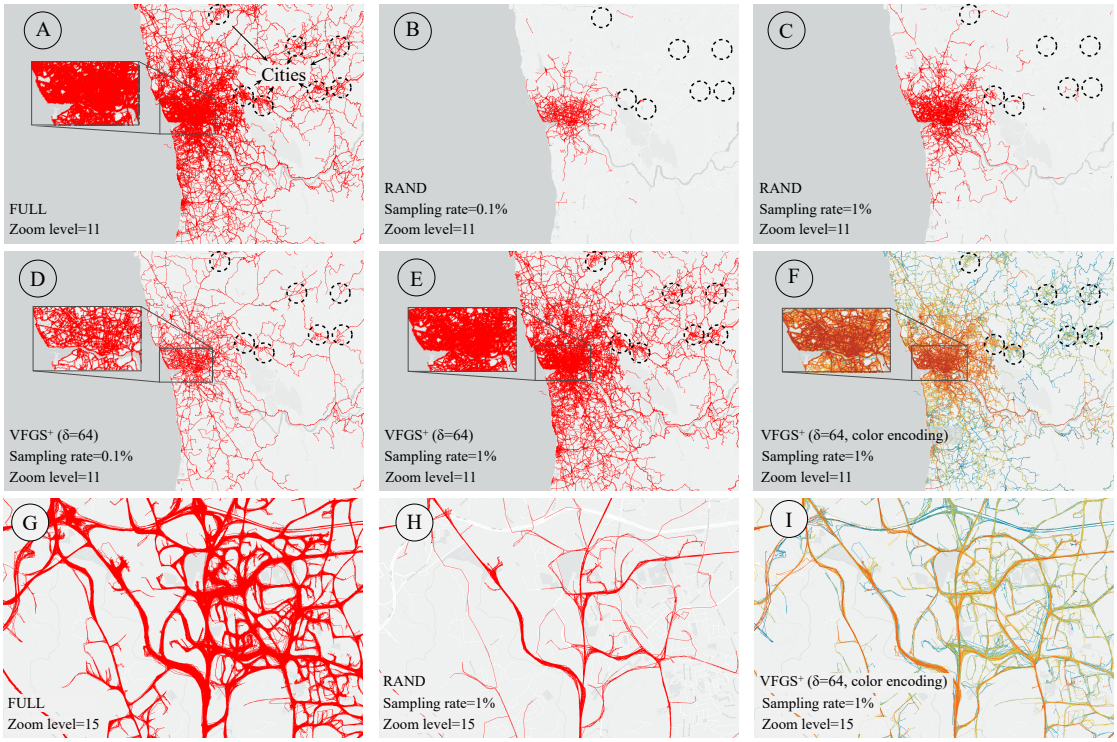


Fig. 1. Comparison of random sampling and VFGS$^+$. (i) A is the visualization of the Porto taxi trajectory dataset at zoom level 11. B and C are visualizations of random sampling with sampling rate 0.1% and 1%, respectively. D, E, F are visualizations of our proposed VFGS$^+$ with different parameters. (ii) G, H, and I are visualizations of the whole Porto dataset, random sampling with sampling rate 1%, and VFGS$^+$ with sampling rate 1% at zoom level 15, respectively.

Abstract— Visualizing large-scale trajectories is the core subroutine in many smart city applications, e.g., traffic management, route recommendation. However, it is very challenging as (i) the large trajectory data size, (ii) the limited rendering capability of graphics device, and (iii) visual clutter in data visualization. In this work, we propose a visual fidelity guaranteed sampling (VFGS) approach for large-scale line-based trajectory visualization. Specifically, we first define a novel fidelity loss function to capture the visual difference between two visualization results. We then prove that it is NP-hardness to select a sized-k subset of trajectories with minimal visual fidelity loss. Next, we devise an approximate algorithm VFGS with a suite of optimization techniques, which returns visual fidelity quality guaranteed visualization result efficiently. Moreover, we further improve the visual effectiveness of VFGS by reduce visual clutter in large data visualization.We conduct extensive experimental studies to demonstrate the effectiveness of our proposal on real-world trajectory datasets. In addition, qualitative user studies further illustrate the superiority of our approach in various applications, e.g., region center identification, reachable route inspection, traffic flow comparison.a Visualizing large trajectory data surfers from limited rendering capability and visual clutter issues. Sampling can effectively mitigate the issues, yet existing methods generally adopt random sampling strategy that has an attenuating effect on visual fidelity. In this work, we propose a visual fidelity guaranteed sampling (VFGS) algorithm for line-based visualization of large trajectory data. We first define a novel fidelity loss function to capture the visual difference between two visualizations. We prove that it is NP-hard to select a sized-k subset of trajectories with minimal visual fidelity loss. Next, we devise an approximation algorithm VFGS with a suite of optimization techniques, which returns fidelity-guaranteed visualizations efficiently. Moreover, we improve the effectiveness of VFGS by taking human perception and cluttering degree into consideration. We conduct extensive experimental studies to demonstrate the effectiveness of our method on real-world trajectory datasets. Qualitative user studies further illustrate the superiority of VFGS in various applications, e.g., traffic flow detection, and reachable route discovery.

Index Terms—Trajectory visualization, quality guaranteed sampling, visual fidelity

1 Introduction

Nowadays, the ubiquitous location-acquisition devices lead to an explosive increase of movement data (e.g., GPS trajectories) for urban moving objects, e.g., vehicles, sharing bikes, and pedestrians. Trajectory visualization has been employed in many smart city applications, e.g., traffic management [38], urban planning [35], route recommendation [47], and location-based services [26, 46]. Line-based trajectory visualization, i.e., connecting the locations of a moving object by poly-

Table 1. Visualization rendering cost of GTX 1080

No. of trajectories	No. of GPS points	Rendering time (s)
1,000	32,648	0.016
10,000	331,583	0.143
100,000	3,262,278	1.416
1,000,000	32,660,845	13.950

lines, is a popular and conventional visualization method [18]. However, large-scale line-based trajectory visualization is challenging. The reasons are (i) large trajectory data size, (ii) limited rendering capability of graphics device, and (iii) visual clutter in large data visualization. We elaborate them as follows:

Large trajectory data size: The trajectory data size is extremely huge. For example, Shenzhen has 24,237 taxis and generates more than 82.8 million GPS locations (e.g., taxi trajectories) in each day [7]. In New York City, there are over 13,000 taxis that averagely carry over 1.0 million passengers and make 500,000 trips per day [20].

Limited rendering capability of graphics device: Rendering refers to the use of the hardware device (e.g., GPU) in the generation of visualizations. However, due to the hardware limitation, the rendering capability of modern commodity GPU is limited. We did a benchmark experiment to evaluate the rendering capability of NVIDIA GeForce GTX 1080 with 8GB video memory. We varied the number of trajectories from 1,000 to 1 million, which are randomly selected from Porto taxi trajectory dataset [5].The experimental results are summarized in Table 1. Obviously, it cannot support interactive visual exploration in large-scale trajectory dataset, e.g., it needs 13.95 seconds to render 1 million trajectories with 32.66 millions GPS points (almost 40% of Shenzhen taxi trajectory GPS points in one day). Moreover, the rendering cost is linear with the input data trajectories. Thus, it is impractical to visualize billion-level GPS points via the commodity GPUs.

Visual clutter in data visualization: Visual clutter is a common issue in data visualization [10]. Fig. 1(A) is the visualization result of the full Porto taxi trajectory dataset. Intuitively, the region shown in the embedded figure of A suffers visual clutter issue seriously, i.e., the road network almost cannot be recognized in it, which hinders the abilities of human-users to explore the dataset and identify the underlying data insights.

To overcome the above challenges, several visualization approaches have been proposed in the literature. Unfortunately, none of them could address these three challenges simultaneously and perfectly. In particular, the spatial aggregation based approaches [37, 44] preprocess the massive movement data, and visualize the results after preprocessing. The aggregation based methods ignored the visual clutter in raw spatial data as they only visualize the aggregated/preprocessed results. In other words, their visualization results may lose the detail information in raw data. In recent years, many visualization research works are proposed to address visual clutter, e.g., edge bundling [36, 45] and density map [25, 34]. However, these works neither focus on line-based trajectory visualization nor designed for large-scale trajectory dataset.

Sampling techniques are de-facto standards for large-scale data analysis in both database and visualization community. In general, it samples a subset of data from the raw large-scale dataset, then it could be rendered efficiently by the graphics device. For example, ScalaR [13] employs a areduction layer between the visualization layer and the data management layer. The reduction layer embedded a uniform random sampling algorithm to sample data randomly when the query results are large enough. It then reduces the amount of data to be visualized. However, the uniform random sampling method does not work well in our large trajectory data visualization problem as it does not have any guarantees about the sampling results. Take Fig. 1(B) and (C) as an example, they are the visualization results of uniform random sampling method on Porto taxi trajectory dataset with sampling rate 0.1% and 1%, respectively. Visually, both visualized results cannot capture the overview of the input data, as shown in Fig. 1(A).

In database community, Park et al. devised a visualization-aware sampling algorithm (VAS) for large-scale scatter points visualization in [29], which offers theoretical quality guarantee on the visualization result. However, the VAS techniques cannot be adapted to our problem as (i) trajectory data is more complex than scatter points (e.g., varying lengths, lack of compact representation), and (ii) the formulated visualization quality measure function in [29] is only for scatter points, it cannot be used to measure the quality of trajectory visualization results.

In this work, we propose visual fidelity guaranteed sampling approaches for the line-based trajectory visualization problem, which kills three birds (i.e., the above three challenges) by one stone. The technical challenges of our proposals are (i) how to define visual fidelity guarantee theoretically? (ii) how to devise an efficient sampling algorithm which offers visual fidelity guarantee on the visualization result, and (iii) how to overcome the visual clutter in large trajectory visualization. Specifically, we first define a novel the visual fidelity loss function between two visualization results formally. With the visual fidelity loss function, we then prove it is NP-hardness to select a sized-k subset of trajectories which has the minimal visual fidelity loss. Next, we devise an approximate algorithm (VFGS) which returns a sized-k subset of trajectories and offers theoretical visual fidelity guarantee on the returning result. Last, we address the visual clutter issue explicitly and overcome it by taking data distribution and human perception capability into consideration in the advance approach (VFGS$^+$). Fig. 1(D) and (E) show the visualization results of our proposal VFGS$^+$ on Porto taxi trajectory dataset with the sampling rate 0.1% and 1%, respectively. Obviously, the visualization fidelity of them (comparing with the visualization result of full dataset in Fig. 1(A)) is much better than the uniform random sampling visualization results with the same sampling rate, see (B) and (C) in Fig. 1. Fig. 1(F) is the returning result of our proposal which colors the trajectories (with different representativeness)according to the trajectory representativeness. It has the same input parameters of Fig. 1(E). Intuitively, the visual clutter in Fig. 1(A) and (E) are alleviated shown as Fig. 1(F). In addition, our proposals are robustness with different zoom levels. Fig. 1(G), (H), and (I) depict the visualization results of the full Porto dataset, the returning result of uniform random sampling RAND and the returning result of VFGS$^+$ with color encoding at zoom-level 15, e.g., we can obtain them by zooming in the visualization result in Fig. 1(A), (C), and (F), respectively. Intuitively, the visualization result of our proposal VFGS$^+$ in Fig. 1(F) outperforms the uniform random sampling method RAND in Fig. 1(H) significantly. It even performs better than Fig. 1(G), the visualized result of the full dataset, as it reduces visual clutter in G by using color encoding scheme to capture the representativeness of different roads.

The contributions of this paper are:

- We formulate the visual fidelity guaranteed sampling problem for large trajectory data visualization, and prove it is (NP hard)NP-hard in Section 3.
- We devise an approximation algorithm for it with a suite of optimization techniques, e.g., submodularity, lazy computing in Section 4.
- We propose an advanced approach to further enhance the effectiveness of our approximate algorithm, which (address)address the visual clutter by introducing perception tolerance parameter, and encodes the representativeness of each (road)trajectory by different colors in Section 5.
- We conduct extensive experiments on real-world trajectory datasets to demonstrate the superiority of our proposals in Section 6. Especially, we conduct qualitative user studies to show (their) the effectiveness on three real-world applications.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 formulates our problem and analyze its hardness. Section 4 provides an approximate solution for it, together with a suite of optimization techniques. Section 5 proposes an advanced solution for our problem. Section 6 elaborates our extensive experimental studies and our findings in detail. Section 7 concludes this work and highlights the promising future directions.

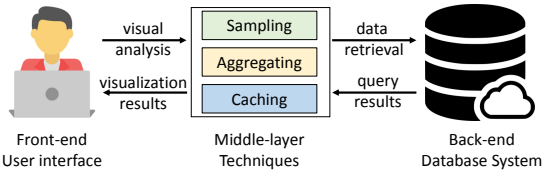


Fig. 2. Interactive visualization system architecture for large dataset

2 Related Work

In this section, we survey previous work and focus on the most relevant pieces. Section 2.1 and 2.2 summarize the related works in trajectory visual analysis and interactive data visualization for large dataset, respectively.

2.1 Trajectory Visual Analysis

Trajectory is the most common representation of the object movements. Each trajectory consists of a sequence of spatial locations (i.e., GPS points). To support the understanding and analyzing of the trajectory dataset, many visualization and visual analytic systems are proposed in the literature. As stated in [18], existing trajectory visual analysis techniques can be classified into three categories according to visualization form, i.e., point-based visualization, line-based visualization and region-based visualization. We briefly introduce the research works in these three categories, and refer the interested readers to a recent survey [18] for detail discussions.

The point-based visualization captures the spatial distribution overview of the GPS points in the moving object trajectories. Many density-based methods, e.g., kernel density estimation, are applied in point-based visualization methods [14,15,27,40,43]. These point-based visualization methods reduce the visual clutter in large amount data by sacrificing the detail information of trajectories, e.g., the sequence order of GPS points. However, the point-based visualization result cannot identify the movement of the individual object and reveal the moving details [18], e.g., direction and path. The region-based visualization approaches divide the whole region into sub-regions in advance, then visualize the aggregated information in each sub-region [21, 37, 39]. These methods demonstrated their effectiveness to capture the macro-patterns. In this work, we focus on the line-based visualization methods, which are widely used in visual analysis applications. It uses polylines to show the trace of the object movements. Through this, it preserves the continuous information of moving objects [22, 23]. However, the line-based visualization methods suffer serious visual clutter due to the cross of the polylines in the large amount trajectories. To alleviate this problem, several techniques have been proposed, such as the clustering-based techniques [19, 33, 37] and advanced interaction techniques [20, 24]. In addition, the edge bundling techniques [36, 45] are designed to reduce visual clutter. Specifically, they wrap up the similar trajectories into bundles, then generate clear visualization results. Unlike existing line-based visualization techniques, we propose visual fidelity guaranteed sampling approaches for line-based trajectory visualization with large-scale input data. To the best of our knowledge, it is the first work which offers theoretical visual fidelity guarantee on the sampling result for large-scale line-based trajectory visualization.

2.2 Interactive Visualization for Large Dataset

(With the recent advanceof location-acquisition technology, the size of available trajectory dataset becomes extremely huge.) With the recent advancement of location-acquisition technology, the size of available trajectory dataset becomes extremely ubiquitous and huge. For example, the operating taxis in Shenzhen generate ~9.3GB trajectory data per day. Due to the limited rendering capability of modern commodity graphics device, generating visualizations for such scale of dataset takes considerable amount of time, or even impractical [29]. In the literature, many works have been proposed to achieve interactive visualization (in)for large dataset (not only trajectory dataset), we briefly elaborate the most representative pieces in (this subsection)the below.

Fig. 2 illustrates the architecture of interactive visualization systems for large datasets, e.g., Spotfire [8], Tableau [9], ATLAS [16], and Viate [42]. (It)The architecture consists of three layers: the user interface in front-end, the optimization techniques in middle-layer, and the (cloud-based) database management system in the back-end. Typically, the researchers in visualization community focused on improving the information visualization effectiveness at the front-end, e.g., designing novel visualization methods to assist data analysts to obtain data insights effectively (D3 [2]).Theypically, the researchers in visualization community typically focusesd on improving the effectiveness of data visualization ininformation visualization effectiveness at the front-end, e.g., to designing novel visualization methods (e.g., D3 [2]) to assist users into assist data analysts to obtaining data insights from data effectively. For the researchers in database community, they are working on the efficiency aspect for large data processing, e.g., devising big data processing systems and techniques for efficient query processing at back-end (Spark [1]). The database community, on the other hand, focuses on improving the efficiency of processing large datasets in the back-end (D3 [2]). For the researchers in database community, they are working on the efficiency aspect for large data processing, e.g., to deviseing big data processing systems and techniques for efficient query (e.g., processing at back-end (Spark [1]). In recent years, both visualization and database communities are dedicating to advance the techniques in interactive visual analysis for large-scale dataset, e.g., the optimizations in the middle-layer (see Fig. 2). We briefly elaborate these optimization techniques (in the middle-layer in subsequent)in subsequent.

Aggregating-based techniques: (Aggregating-based optimization techniques [37, 44] in the middle-layer support interactive visual analysis for large-scale data by reducing the number of rendering data.) Aggregating-based techniques pre-process raw data with aggregation techniques (e.g., clustering) in the middle-layer, yielding fewer rendering objects for interactive visual analytics. It achieves by preprocessing the raw data with aggregation techniques (e.g., clustering). Returning to the trajectory visual analysis, many works [21, 37, 39] partition the (spatial space)underlying territory into basic units, then visualize the information upon them by aggregation algorithms. For more details of other aggregating-based techniques, we refer the reader to surveys [12]. However, aggregating-based methods will cause information loss definitely. For instance, the continuous spatial traces of the moving objects are always missing and the rarely appeared trajectories are easily to be ignored. Our problem and solutions are different from these research works as we focus on visualizing the raw input data, instead of the aggregated results.

Sampling-based techniques: Sampling techniques are commonly used in the interactive visualization problems with large-scale input data. It is widely studied in both visualization and database communities [13,17,29]. In particular, [17] devised a sampling algorithm to preserve the meaningful data items according to the analyzing requirement such as the multi-class data analysis and hierarchical exploration. The most relevant work of ours in the literature is [29], which designed for the scatter plot and aim to not only solve the overdrawing of the points but also try to preserve the information distribution of the original dataset. Specifically, they formulated a loss function which evaluates the visual loss of the sampling result effectively, they validate the proposed method by three common visualization tasks, e.g., regression, density estimation and clustering. However, the techniques in [29] cannot be adapted to our large-scale trajectory visualization problem as (i) the complexity of the trajectories [31], and (ii) the loss function and its corresponding solutions are specified for scatter plot, not applicable for line-based trajectory visualization. For trajectory visual analysis, most of the existing trajectory sampling techniques (if not all) cluster the trajectories at first, then select the most representative trajectories from each cluster and visualize them. It is impractical to provide interactive visualizations for real-world applications as (i) the trajectory similarity computation and clustering algorithms are very expensive [30], and (ii) the trajectory clustering is still an open problem in both database and visualization communities [11,28]. Unlike the above research works, in this paper, we propose visual fidelity guaranteed sampling approaches for the large-scale trajectory visualization problem, we demonstrate the

superiority of our proposals by case-, user- and qualitative studies in real world datasets.

Caching-based and other techniques: Caching is commonly used to improve the performance of large data processing system, e.g., search engine [41]. Chan et al. present ATLAS [16] which utilizes caching techniques for the efficient data communication between server and client. In addition, it also exploits the powerful multi-core server to accelerate visual analysis task processing from the middle-layer to the back-end. Piringer et al. [32] propose a multi-threading architecture for the interactive visual exploration, which utilizes multi-core devices and avoids the multi-threading pitfalls to provide quick visual feedback. (However, our)Our proposed techniques in this work are orthogonal to the researches in this category.

3 Problem Statement

In this section, we first define our research problem in Section 3.1 formally. Section 3.2 analyzes its hardness.

3.1 Problem Definition

As we analyzed in Section 1, the large-scale (e.g., hundreds of millions GPS points) line-based trajectory visualization problem is very challenging due to the large data size and limited rendering capability of graphics devices. To make matters worse, the visualization result of large-scale trajectory dataset suffers visual clutter seriously. In this work, we focus on how to visualize large-scale trajectory dataset efficiently and effectively. In particular, our objective is to devise a visual fidelity guaranteed sampling method for large trajectory data visualization. The major challenges to achieve this goal are: (I) how to define visual fidelity theoretically? (II) how to guarantee the visual fidelity of the sampling-based visualization result? We commence our presentation by defining our research problem formally as follows.

Problem 1 (Large-scale trajectory visualization problem)**.** *Given a large-scale trajectory dataset* T *and a sampling rate* α*, the trajectory visualization problem is selecting a subset of trajectories* $\mathsf{R} \subseteq \mathsf{T}$ *with* $|\mathsf{R}| \leq \alpha|\mathsf{T}|$*, such that visual fidelity loss function* $loss(\mathsf{R},\mathsf{T})$ *is minimized.*

The key to solving the large-scale trajectory visualization problem (see Problem 1) is defining the visual fidelity loss function properly. Intuitively, the visual fidelity of the sampled visualization results R w.r.t. the original dataset T depends on the user specified visualization level of details (a.k.a., LOD). Given an empty canvas (e.g., displaying device) with a user specified level of details, the visualization process is rendering the trajectories into canvas with the given level of details (e.g., the number of pixels in each row and each column). Considering a trajectory data set T and a subset of trajectories $\mathsf{R} \subseteq \mathsf{T}$, the visual fidelity loss between R and T is defining as the different pixels of the visualization results about R and T in the canvas with specified LOD. We then define the visual fidelity loss function of sampling-based trajectory visualization problem as $loss(\mathsf{T},\mathsf{R}) = \frac{|\mathsf{V}(\mathsf{T})-\mathsf{V}(\mathsf{R})|}{|\mathsf{V}(\mathsf{T})|}$, where $\mathsf{V}()$ measures the rendered pixels in the canvas of the input trajectory dataset.

Thus, given a trajectory data set T and a sampling rate α, our research objective is finding subset R, such that the visualization fidelity loss function $loss(\mathsf{T},\mathsf{R})$ is minimized, i.e.,

$$\min_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|\leq\alpha|\mathsf{T}|} loss(\mathsf{T},\mathsf{R}) = \frac{|\mathsf{V}(\mathsf{T})-\mathsf{V}(\mathsf{R})|}{|\mathsf{V}(\mathsf{T})|}.$$

3.2 Hardness analysis

For the sake of presentation, we analyze the hardness of our research objective with a simple render manner of visualization result. We are aware there exists complex rendering scheme, e.g., different pixels has different colors, we will consider it shortly. In particular, for each pixel in the canvas with simple render manner, it will be rendered if there is a trajectory pass through it, otherwise it will not be rendered. Suppose each pixel in the canvas has an unique id, let U be the universal set of all pixels in the canvas. For each trajectory $t_i \in \mathsf{T}$, it consists of a set of pixels in the canvas. In other words, the pixel set of each trajectory $t_i \in \mathsf{T}$ is a subset of U. Consequently, the pixel set of the selected trajectory set R is also a subset of U as $\mathsf{R} = \cup_{t_i\in\mathsf{R}} t_i$.

Our research objective is minimizing loss function $loss(\mathsf{T},\mathsf{R}) = \frac{|\mathsf{V}(\mathsf{T})-\mathsf{V}(\mathsf{R})|}{|\mathsf{V}(\mathsf{T})|}$ subject to $\mathsf{R} \subseteq \mathsf{T}$ and $|\mathsf{R}| \leq \alpha|\mathsf{T}|$. Given an empty canvas, the visualized/rendered pixels of the input trajectory dataset T is a constant set, denotes as C.

Hence, our research objective of Problem 1 can be transformed as follows:

$$\begin{aligned}\text{Objective}: &\min_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=\alpha|\mathsf{T}|} \frac{|\mathsf{V}(\mathsf{T})-\mathsf{V}(\mathsf{R})|}{|\mathsf{V}(\mathsf{T})|} \Leftrightarrow \min_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=\alpha|\mathsf{T}|} \frac{|\mathsf{C}-\mathsf{V}(\mathsf{R})|}{|\mathsf{C}|} \\ &\Leftrightarrow \min_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=\alpha|\mathsf{T}|} -|\mathsf{V}(\mathsf{R})| \Leftrightarrow \max_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=\alpha|\mathsf{T}|} |\mathsf{V}(\mathsf{R})| \\ &\Leftrightarrow \max_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=\alpha|\mathsf{T}|} |\cup_{t_i\in\mathsf{R}} t_i|\end{aligned}$$

It is equivalent to the typical set cover maximization problem. Specifically, given an integer $k = \alpha|\mathsf{T}|$, and a collection trajectory pixel set $\mathsf{T} = \{t_1, t_2, \cdots, t_n\}$ with $\forall t_i \subset \mathsf{U}$, the objective of the set cover maximization problem is finding a subset $\mathsf{R} \subset \mathsf{T}$ such that $|\mathsf{R}| \leq k$ and the number of covered pixels in $|\cup_{t_i\in\mathsf{R}} t_i|$ is maximized. We omit the proof of its NP-hardness, as it has been shown in [6].

4 Visual Fidelity Guaranteed Sampling Approach

Due to the hardness of the Problem 1, we first introduce a straightforward solution (i.e., uniform random sampling) for it in Section 4.1. Then, we propose a visual fidelity guaranteed sampling approach in Section 4.2. Last, we devise several optimizations to improve the efficiency and effectiveness of our proposal in Section 4.3.

4.1 Uniform Random Sampling Algorithm

The straight forward solution for Problem 1 is uniform random sampling RAND. As the pseudocode in Algorithm 1 shown, it selects k trajectories from T randomly, which store in R, then render these selected trajectories in R as the visualization result.

Algorithm 1 $\mathsf{RAND}(\mathsf{T}, k = \alpha|\mathsf{T}|)$

```
1: Initialize result set R ← ∅
2: while |R| < k do
3:     tmp ← RandomSampling(T − R)
4:     R ← R ∪ {tmp}
5: Return R
```

Obviously, the uniform random sampling algorithm has good performance. However, it does not provide any guarantee on the visual fidelity of the sampled result set.

4.2 Visual Fidelity Guaranteed Sampling Algorithm

In this section, we present our visual fidelity guaranteed sampling algorithm for Problem 1. We start our presentation by elaborating the correlation between visual fidelity of sampled set R and user zoom level. For a given sampled set $\mathsf{R} \subseteq \mathsf{T}$, it has different visual fidelity loss values at different user zoom levels. The reason is the resolutions of R's visualized result are different at different zoom levels. For example, Google map [3] provides zoom levels range from 0 to 20, where level 0 is the lowest level (e.g., the whole world), level 20 is the highest level (e.g., individual building, if available). In order to devise a zoom level oblivious visualization for sampled dataset R, we use the highest zoom level to define the size of each pixel in the canvas in our problem. It means for each trajectory $t_i \in \mathsf{T}$, it is a set of pixels in the canvas at the highest zoom level.

The visual fidelity guaranteed sampling algorithm employs greedy paradigm. In particular, it finds the trajectory tmp in T which maximize the result set of $|\mathsf{R} \cup tmp|$ at each iteration, as Line 3 shown in Algorithm 2. It terminates after $k = \alpha|\mathsf{T}|$ iterations and returns R as result set for rendering.

Algorithm 2 VFGS($\mathsf{T}, k = \alpha|\mathsf{T}|$)

1: Initialize result set $\mathsf{R} \leftarrow \emptyset$
2: **while** $|\mathsf{R}| < k$ **do**
3: $\quad tmp \leftarrow argmax_{t_i \in \mathsf{T}} |\mathsf{R} \cup t_i|$
4: $\quad \mathsf{R} \leftarrow \mathsf{R} \cup \{tmp\}$
5: Return R

Interestingly, Algorithm 2 offers theoretical visual fidelity guarantee of the returning result R, as proved in Theorem 1.

Theorem 1. *Algorithm 2 provides* $1-(1-1/k)^k \geq (1-1/e) \approx 0.632$ *approximation result for large-scale trajectory visualization problem (i.e., Problem 1).*

Proof. The optimal solution of Problem 1 covers OPT pixels in k iterations. Let a_i be the number of newly covered pixels at the i-th iteration, b_i is the total number of covered pixels up to the i-th iteration (i.e., $b_i = \sum_{j=1}^{i} a_i$), and c_i be the uncovered pixels after i-th iteration (i.e., $c_i = OPT - b_i$). According to greedy paradigm, we can conclude the number of newly covered pixels at the $(i+1)$-th iteration is always greater than or equal to $\frac{1}{k}$ of the number of uncovered pixels after the i-th iteration, i.e., $a_{i+1} \geq \frac{c_i}{k}$. We prove Theorem 1 by proving $c_{i+1} \leq (1-1/k)^{i+1} \cdot OPT$. It holds $c_1 \leq (1-1/k) \cdot OPT$ as follows.

$$a_1 \geq c_0 \cdot 1/k = 1/k \cdot OPT \quad \text{as we concluded} \quad a_{i+1} \geq \frac{c_i}{k}$$
$$\Leftrightarrow b_1 \geq 1/k \cdot OPT \Leftrightarrow -b_1 \leq -1/k \cdot OPT \quad \text{as} \quad a_1 = b_1$$
$$\Leftrightarrow OPT - b_1 \leq OPT - 1/k \cdot OPT \Leftrightarrow c_1 \leq (1-1/k) \cdot OPT$$

For inductive hypothesis assume $c_i \leq (1-1/k)^i \cdot OPT$. Thus,

$$c_{i+1} = c_i - a_{i+1} \leq c_i - c_i/k = (1-1/k) \cdot c_i = (1-1/k)^{i+1} \cdot OPT$$

Hence, it holds $c_k \leq (1-1/k)^k \cdot OPT$. It is equivalent to $b_k \geq (1-(1-1/k)^k) \cdot OPT \geq (1-1/e) \cdot OPT \approx 0.632 \cdot OPT$. □

4.3 Optimization Techniques

With the above analysis, Algorithm 2 (VFGS) provides a visual fidelity guaranteed sampling algorithm for large-scale trajectory data visualization problem. However, it is inefficient for (very) large trajectory dataset (e.g., millions of trajectories) as the time complexity analyzed in the following Lemma 1.

Lemma 1 (Time Complexity)**.** *Given trajectory dataset* T *and an integer* $k = \alpha|\mathsf{T}|$*, the time complexity of Algorithm 2 is* $O(\alpha \cdot m \cdot |\mathsf{T}|^2)$*, where m is the maximum length of all trajectories in dataset* T.

Proof. At each iteration ($k = \alpha|\mathsf{T}|$ iterations in total), it computes the uncovered pixels of each trajectory in dataset T with $O(m)$ cost. The dataset T has $O(|D|)$ trajectories. Thus, the total cost is $O(k \cdot m \cdot |\mathsf{T}|) = O(\alpha \cdot m \cdot |\mathsf{T}|^2)$. □

Example: Given Porto trajectory dataset, it has 2.39 millions of taxi trajectories, the maximum length in it is 3,490. It takes 413.6 seconds to return a subset R with sampling rate 0.1%. Obviously, it is impractical for interactive trajectory explorations.

Due to the inefficient of our visual fidelity guaranteed sampling approach in Algorithm 2, we then devise performance optimizations to accelerate its running time. The core idea is utilizing the submodularity of the covered pixels of result set R, as shown in Lemma 2.

Lemma 2 (Submodularity)**.** *Suppose the contribution of trajectory t to the result set* R *is* $\Delta(\mathsf{R}, t) = |\mathsf{R} \cup t| - |\mathsf{R}|$*. Given a trajectory t and two result sets* R, R'*, where* $\mathsf{R} \subset \mathsf{R}'$ *and* $t \notin \mathsf{R}$*, it holds* $\Delta(\mathsf{R}, t) \geq \Delta(\mathsf{R}', t)$.

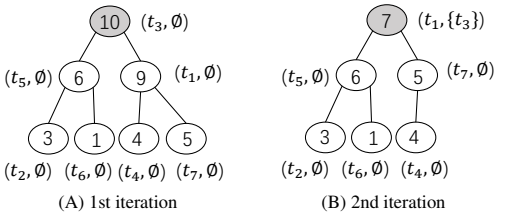


Fig. 3. An illustration of lazy computing manner via Lemma 2

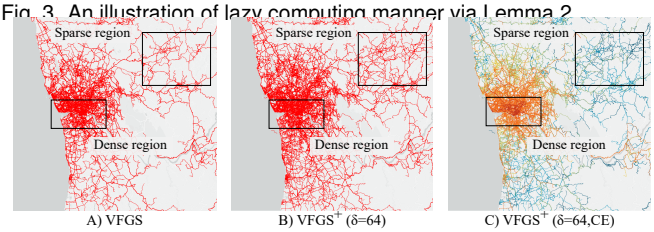


Fig. 4. Advance Approach VFGS$^+$ with Porto trajectory dataset, sampling rate is 0.5%: (A) VFGS, (B) VFGS$^+$, and (C) VFGS$^+$CE

Proof. The contribution value of trajectory t to a given result set R (e.g., $\Delta(\mathsf{R}, t) = |\mathsf{R} \cup t| - |\mathsf{R}|$) is the new covered pixels of t w.r.t. result set R, i.e., $|t| - |\mathsf{R} \cap t|$. It holds $t \cap \mathsf{R} \subseteq t \cap \mathsf{R}'$ as R' is a superset of R. Thus, we have $|t| - |t \cap \mathsf{R}| \geq |t| - |t \cap \mathsf{R}'|$. Hence, it holds $\Delta(\mathsf{R}, t) = |\mathsf{R} \cup t| - |\mathsf{R}| \geq |\mathsf{R}' \cup t| - |\mathsf{R}'| = \Delta(\mathsf{R}', t)$. □

With the help of submodularity in Lemma 2, it reduces many unnecessary trajectory contribution value computations. In particular, we maintain a max-heap for the number of uncovered pixels of each trajectory, we employ a lazy computing manner, i.e., only compute the contributions of a given trajectory when it is necessary. Fig. 3(a) shows a tiny max-heap example about the numbers of uncovered pixels of each trajectory from t_1 to t_7 with result set $\mathsf{R} = \emptyset$. At the 1st iteration, the root node of the max-heap will be selected, i.e., t_3 in Fig. 3(A). At the 2nd iteration, the number of uncovered pixels of the root node t_1 is updated to 7 w.r.t. result set $\mathsf{R} = \{t_3\}$ (see gray node at Fig. 3(B)). Then t_1 will be selected at the 2nd iteration without computing the number of uncovered pixels in other trajectories, i.e., all white nodes at Fig. 3(B). The reason is their contributions will be less than 7 via the submodularity shown in Lemma 2.

The performance of Algorithm 2 is improved significantly as we only compute its contribution values when it is necessary. To exemplify, Algorithm 2 costs 413.6 seconds to return the results with sampling rate 0.1% Porto taxi trajectory dataset. However, it only needs 1.2 seconds in our performance optimized VFGS.

5 Advance Approach: VFGS$^+$

Until now, VFGS in Algorithm 2 offers a visual fidelity guaranteed sampling approach for large-scale trajectory visualization problem (see Problem 1), which returns the visual fidelity guaranteed result efficiently via the optimization techniques in Section 4.3. It means that the challenges (i) large trajectory dataset and (ii) limited rendering capability of graphics device (see Section 1) have been addressed. In this section, we focus on the third challenge of it, i.e., visual clutter. In particular, we devise an advance approach VFGS$^+$ to alleviate it by considering (i) trajectory data distribution, and (ii) human perception capability. We elaborate (i) and (ii) by the examples in Fig. 4 shortly.

Trajectory data distribution: Considering Porto trajectory dataset, Fig. 4(A) is the visualization result of VFGS with sampling ratio 0.5%. Obviously, the real-world trajectory dataset is non-uniform distributed. For example, the trajectories in the dense region are much more than these in the sparse region, as illustrated by the rectangles in Fig. 4(A).

Human perception capability: Intuitively, it is much easier for humans to distinguish the difference between sparse regions rather than dense regions in Fig. 4(A) and (B). The core reason is the perception capability of human beings is limited. In particular, the visual difference of human beings will be diminished when the visualized trajectories is

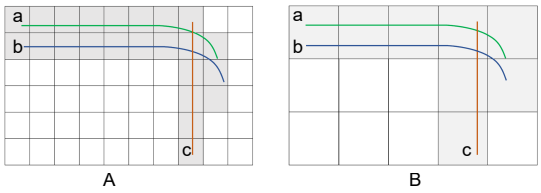


Fig. 5. An illustration of VFGS$^+$ with different zoom levels

large enough with a given level of details, i.e., the difference between two dense regions in Fig. 4(A) and (B).

Taking the above two observations into consideration, the returning result of visual fidelity guaranteed sampling approach VFGS could be further improved by delivering richer information at sparse regions and reducing visual clutter in dense regions. In this section, we devise an advance approach VFGS$^+$ (see Algorithm 3) to achieve the above two objectives. Specifically, we introduce perception tolerance parameter δ in VFGS$^+$, which models human's perception capability at the highest level of details. In other words, suppose the pixel (x,y) in canvas is covered by result set R at the highest level, the pixels around (x,y), i.e., from $(x-\delta, y-\delta)$ to $(x+\delta, y+\delta)$, are not necessary to cover as they are in the perception tolerance of human beings.

Fortunately, we can slightly revise VFGS in Algorithm 2 to incorporate the perception tolerance parameter δ in advance approach VFGS$^+$, as shown in Algorithm 3. It measures the contribution of each trajectory t_i w.r.t the selected trajectory set R's augmented set R$^+$ (in Line 4). The augmented set R$^+$ will be updated by the selected trajectory tmp and its tolerance pixels set (in Line 6).

Algorithm 3 VFGS$^+(\mathsf{T}, k = \alpha|\mathsf{T}|, \delta)$

1: Initialize result set $\mathsf{R} \leftarrow \emptyset$
2: Initialize augmented result set $\mathsf{R}^+ \leftarrow \emptyset$
3: **while** $|\mathsf{R}| < k$ **do**
4: $\quad tmp \leftarrow argmax_{t_i \in \mathsf{T}} \mathsf{R}^+ \cup t_i$
5: $\quad \mathsf{R} \leftarrow \mathsf{R} \cup \{tmp\}$
6: $\quad \mathsf{R}^+ \leftarrow \mathsf{R}^+ \cup \mathsf{augment}(tmp, \delta)$
7: **for** each t in T **do** ▷ Representative encoding
8: $\quad tr \leftarrow argmin_{t_i \in \mathsf{R}} \mathsf{augment}(t_i, \delta) - t$
9: $\quad tr.\mathsf{cnt}++$
10: Return R

Interestingly, the visual clutter large trajectory visualization problem can be further reduced by encoding representative trajectories in R (the returning result of the advance approach VFGS$^+$) with colors. In particular, VFGS$^+$ selects the trajectory which has the largest uncovered pixels by taking human's perception tolerance capability into account at each iteration, instead of only choosing the trajectory with the largest uncovered pixels in VFGS (see Algorithm 2). During its selection process, some of trajectories will not be included into the result set R even they have more uncovered pixels w.r.t. R. The reason is their uncovered pixels are too close to the pixels in the selected trajectories, i.e., within the tolerance area of selected pixels. Taken Fig. 5(A) as an example, suppose $\delta = 1$ and trajectory a was selected at the first iteration, the selected trajectory in the second iteration is c instead of b as almost all pixels in b is in the tolerance area of a's.

Inherently, the VFGS$^+$ trajectory selection process embeds the representativeness of each trajectory in the result set R. We define the representativeness of a trajectory as the number of influenced trajectories in the dataset T. We compute the representativeness of each trajectory in R from Line 7 to Line 9 in Algorithm 3, then visualize them by encoding with different colors. Fig. 4(C) shows the visualized result of the advance approach VFGS$^+$ by encoding the trajectory representativeness with colors. Obviously, the trajectories in dense region have darker color than these in sparse regions as there many trajectories in dense region, thus the selected trajectories in the dense region are more representative.

Last but not least, it is worth to point out that our advance approach VFGS$^+$ provides excellent visual fidelity over VFGS at arbitrary zooming resolutions naturally. The key technique to achieve that is it considers the zooming resolutions inherently when introducing the perception tolerance δ. Take Fig. 5 as an example, the zoom level in Fig. 5(A) is higher than it in Fig. 5(B). As our above elaboration, our advance approach VFGS$^+$ selects trajectory a and c at Fig. 5(A). When it zoomed out, as shown in Fig. 5(b), it still captures the main sketch of the underlying dataset (as gray cells shown).

6 Experimental Evaluation

We evaluate our techniques on two real-world datasets, i.e., Porto and Shenzhen in this section. The trajectory dataset of Porto [5] has 2.39 million of taxi trajectories, 75.67 million of GPS points in total, its maximum trajectory length is 3,490 GPS points. The trajectories in Porto cover several cities around the Porto and has been cleaned for further analysis. Shenzhen [7] includes 3.07 million of taxi trajectories with 53.53 GPS points, the maximum trajectory in it has 2,268 GPS points.

In Section 6.1, we evaluate the effectiveness of our proposal by the case studies on Porto and Shenzhen trajectory dataset, respectively. We then conduct a user study to demonstrate the superiority of our proposal in three real world applications in Section 6.2. We perform a qualitative evaluation in Section 6.3 at last. We conducted all experiments on a machine with Intel i7-8700, 3.2 GHertz CPU, 24 GBytes memory and NVIDIA GeForce GTX1080, 8 GHertz VRAM GPU, running on Windows 10. We implemented all methods in Java 1.8. The methods call on the Processing 3 [4] for rendering.

6.1 Case Study

We demonstrate the effectiveness of our proposal by the case studies in Porto (in Section 6.1.1) and Shenzhen (in Section 6.1.2).

6.1.1 Case Studies on Porto

We first refer the cases in Fig. 1 to elaborate the effectiveness of our proposal from the following three aspects.

Effect of approaches with different zoom-levels: Considering zoom level 11, Fig. 1(A) is the visualization result of full Porto trajectory dataset. Given the sampling rate $\alpha = 1\%$, Fig. 1(C) and (E) visualized the returning results of uniform random sampling algorithm (RAND, see Algorithm 1) and our advanced visual fidelity guaranteed sampling approach (VFGS$^+$, see Algorithm 3), respectively. Obviously, Fig. 1(E) looks much more closer to Fig. 1(A) when comparing with Fig. 1(C). In particular, our proposed VFGS$^+$ not only preserves the visual structure of the full dataset, but also shows the details of these cities which are far away from the center, as the dashed cycles shown in Fig. 1(E). However, the details of these dashed cycles in Fig. 1(C), i.e., the returning result from RAND, are lost. This issue turns more serious when we zoom in to the details of the visualization results. Fig. 1(H) and (I) are the visualization result of RAND and VFGS$^+$ (with color encoding) at zoom level 15 with sampling rate $\alpha = 1\%$. Comparing with the visualization result of full dataset at such level, as shown in Fig. 1(G), the RAND in Fig. 1(H) only returns few trajectories and many information in raw data are lost. Surprisingly, our VFGS$^+$ in Fig. 1(I) captures the main sketch of the full dataset, even with more clear details with the help of color encoding.

Effect of sampling rate: We then evaluate the effect of sampling rate in different approaches. Fig. 1(B) and (C) are the visualization results of RAND with sampling rate 0.1% and 1%, respectively. Fig. 1(D) and (E) visualized the returning trajectory sets of our VFGS$^+$ with sampling rate 0.1% and 1%, respectively. We then have the following observations. (I) the larger sampling rate, the better visual fidelity of the visualization results, e.g., Fig. 1(C) and (E) are more closer to Fig. 1(A) when comparing with Fig. 1(B) and (D), respectively. (II) the visualization result of VFGS$^+$ with sampling rate 0.1% in Fig. 1(D) performs even more better than the result of RAND with sampling rate 1% in Fig. 1(C) to capture the visual structure of the full dataset in Fig. 1(A).

Effect of color encoding: Next, we present the superiority of color encoding scheme. Given zoom level 11 and sampling rate 1%, Fig. 1(E) and (F) are the visualization results of our VFGS$^+$ without and with

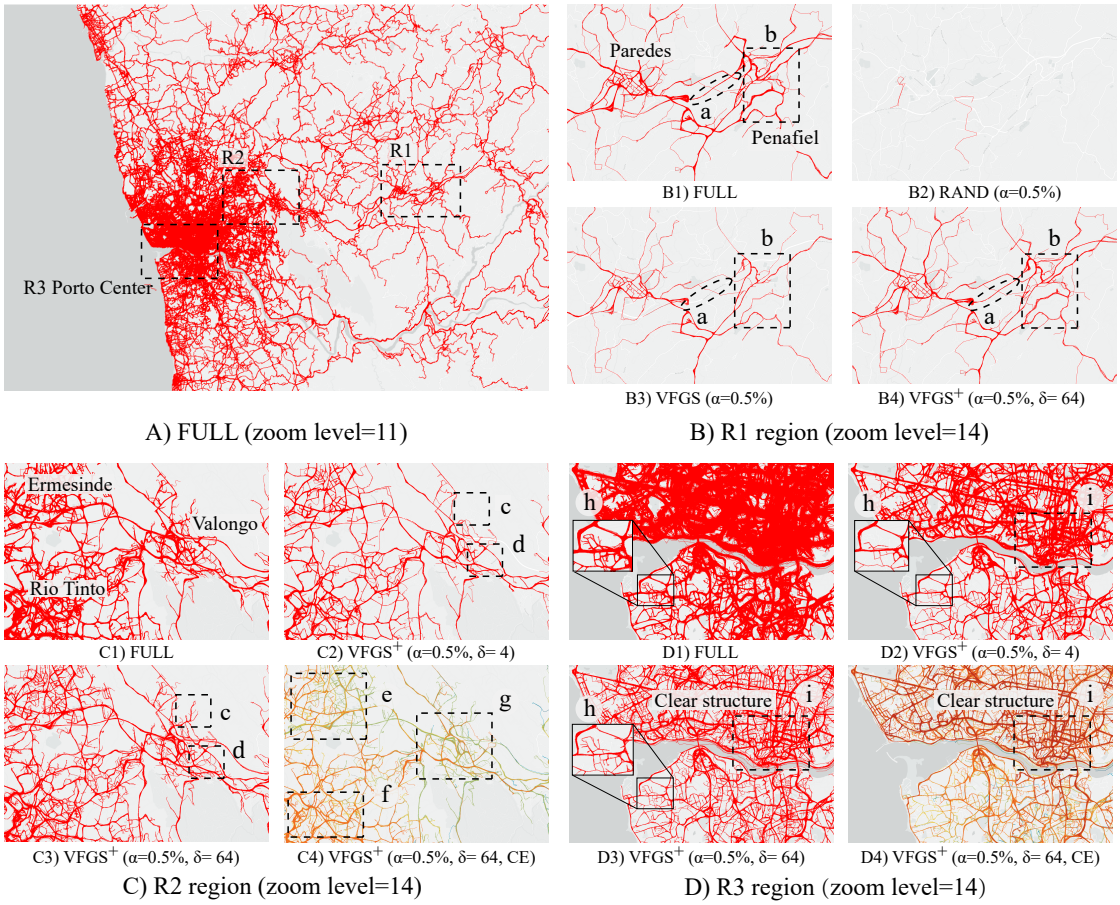


Fig. 6. Effectiveness studies of VFGS$^+$ at dense and sparse regions with detail visualizations in Porto.

color encoding scheme, respectively. It is worth to point out the visual clutter in the visualization result of full dataset is serious, as the embedded rectangle shown in Fig. 1(A). It also exists in the result of VFGS$^+$, see the embedded rectangle in Fig. 1(E). However, the color encoding scheme in Fig. 1(F) reduces the visual clutter in Fig. 1(A) and (E) successfully. Obviously, it provides clear visual structure of the input dataset. In addition, the visualized results in Fig. 1(G) and (I) confirm the effectiveness of our proposed color encoding scheme for visual clutter in large dataset again.

We next present the effectiveness of our proposals with different detail views by investigating three regions of interest in Porto, as R1, R2, and R3 shown in Fig. 6(A).

Sparse region R1: R1 is a sparse region and has few trajectories, as the visualization result of full Porto dataset shown in Fig. 6(B1). The reason is the two cities Paredes and Penafiel in R1 are far away from the center of Porto. Given sampling rate $\alpha = 0.5\%$, Fig. 6(B2), (B3) and (B4) are the visualization results of the returning trajectory set from RAND, VFGS and VFGS$^+$ with $\delta = 64$, respectively. As our above statement, the result of RAND almost misses all information in sparse region. While VFGS performs much better than RAND as it provides theoretical visual fidelity guarantee, it still lost detail information. Taking Fig. 6(B1) as reference, the trajectory bundle and trajectory structure are lost in Fig. 6(B3a) and (B3b). As expected, our advanced approach VFGS$^+$ in Fig. 6(B4) with perception tolerance value $\delta = 64$ did an excellent job to capture the details in the full dataset when comparing with VFGS in Fig. 6(B3). As shown in Fig. 6(B4b), the trajectory sketch of Penafiel is almost the same as it in Fig. 6(B1b), the visualized result of full dataset.

Median region R2: It is near to the center of Porto, which has more taxi trajectories than R1, see Fig. 6(A). As noted in Fig. 6(C1), R2 includes three cities: Ermesinde, Rio Tinto and Valongo. Fig. 6(C2) and (C3) visualized the returning result of VFGS$^+$ with perception tolerance value $\delta = 4$ and 64, respectively. Obviously, Fig. 6(C3) has more trajectory branch details than (C2), as the rectangles c and d shown in them. In other words, the larger perception tolerance value, the more details in this region reserved at zoom level 14. Fig. 6(C4) is the result of VFGS$^+$ which enable color encoding scheme. Intuitively, Fig. 6(C4) shows its superiority over (C3) to capture the trajectory distributions. For example, the color of the region f in Fig. 6(C4) is darker than the rest two regions *e* and *g*. Thus, we can conclude Rio Tinto (region f) has more taxi trajectories than other two cities, which is hard to be concluded via Fig. 6(C3) even (C1), the visualization result of full dataset. It also confirms the color encoding scheme could enrich the visual information in large trajectory visualization.

Dense region R3: It is the center of Porto, which has the highest concentration of the trajectories and causes serious visual clutter, as visualized in Fig. 6(D1). For example, the structure of trajectories in Fig. 6(D1i) is unclear. VFGS$^+$ with $\delta = 4$ alleviates the visual clutter and preserves the trajectory distribution, see Fig. 6(D2). Fig. 6(D3) visualized the result of VFGS$^+$ with $\delta = 64$, which enhances the visual fidelity of Fig. 6(D2). Specifically, it preserves more details(see rectangle h) and has a more clear structure in the densest region (see rectangle i). Visually, Fig. 6(D4) is the best among these four visualization results. It confirms the advantages of VFGS$^+$ with color encoding scheme.

6.1.2 Case Studies on Shenzhen

We further evaluate the effectiveness of our proposal by using the taxi trajectories in Shenzhen, China. The Shenzhen trajectory dataset has many different characteristics with Porto, e.g., trajectory distribution, city centers, taxi move patterns. We set sampling rate $\alpha = 1\%$ and perception tolerance value $\delta = 64$ in this section.

Overview of Shenzhen: Fig. 7(A) is the visualization result of full Shenzhen dataset at zoom level 11. The dense regions in southern

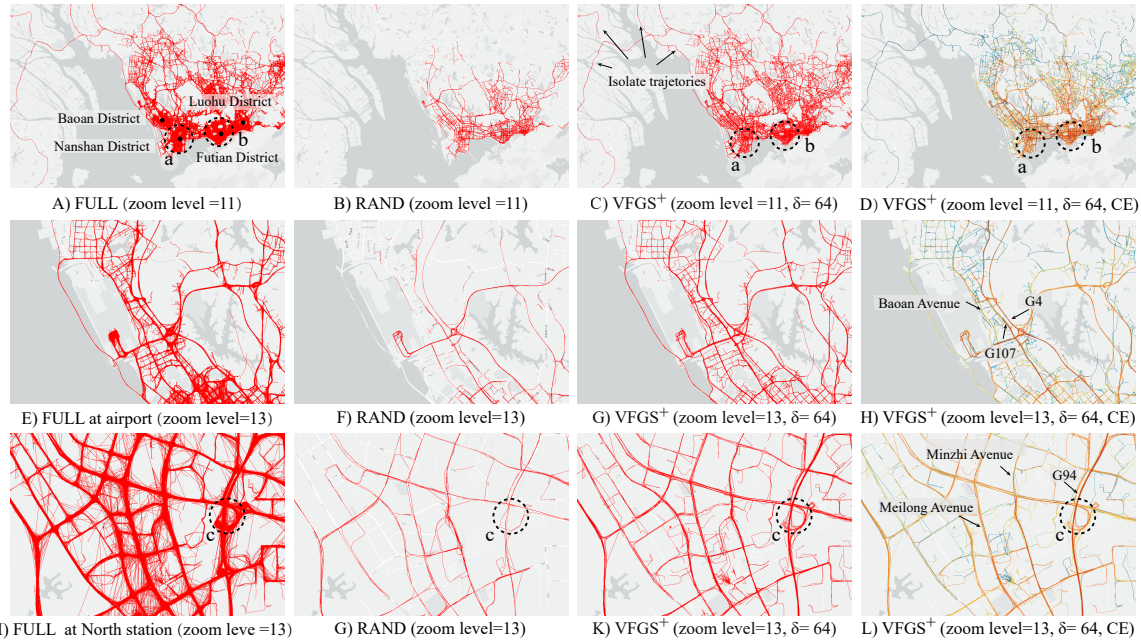


Fig. 7. Case studies on Shenzhen taxi trajectory dataset, sampling rate $\alpha = 1\%$

of Shenzhen, as the dashed circles shown in Fig. 7(A), are *Baoan, Nanshan, Futian* and *Luohu* districts, which are the most prosperous commercial regions in this city. The returning results of RAND, VFGS$^+$ and VFGS$^+$ with color encoding scheme are visualized in Fig. 7(B), (C) and (D), respectively. Not surprisingly, the visualized result of RAND in Fig. 7(B) is quite different from the full dataset in Fig. 7(A). VFGS$^+$ in Fig. 7(C) shows it superiority by capturing the overview of Shenzhen dataset and even preserves the isolated trajectories, as left-upper corner highlighted in Fig. 7(C). It owes to VFGS$^+$ provides theoretical visual fidelity guarantees on the returning result set. VFGS$^+$ with color encoding further improved the visual fidelity of VFGS$^+$. Specifically, both Fig. 7(A) and (C) are suffering from visual clutter seriously, e.g., it is unable to recognize the main roads in the circles a and b as both are full with visualized trajectories. However, the result of VFGS$^+$ with color encoding, as shown in Fig. 7(D), reduce the visual clutter perfectly. For example, we can conclude the main roads of circle a and b are these roads with darker colors.

We then present the advantages of our proposal in two representative areas, i.e., airport and North railway station, in Shenzhen dataset.

Airport in Shenzhen: Compare with visualization result of full dataset in Fig. 7(E), the visualized result of RAND in Fig. 7(F) only includes very few trajectories. VFGS$^+$ without and with color encoding in Fig. 7(G) and (H) reserve the major structure of the airport area excellent. Moreover, VFGS$^+$ with color encoding provides richer information by computing the representativeness of trajectories. For example, the taxi trajectories which pass through G4 and G104 is more than that in Baoan Avenue. The reason is that the colors of G4 and G104 is darker than Baoan Avenue, as notated in Fig. 7(H).

North railway station in Shenzhen: We next investigate the visualizations of the full dataset, RAND result set, and VFGS$^+$ result set around North railway station of Shenzhen, which are shown from Fig. 7(I) to (L). Interestingly, VFGS$^+$ without and with color encoding scheme visualized the overpass near North railway state clearly, as circle c shown in both Fig. 7(K) and (L). It is not clear in full dataset visualization shown as Fig. 7(Ic) due to visual clutter. It even disappeared in the visualized result of RAND in Fig. 7(G). Moreover, it is easy to estimate the traffic flow in VFGS$^+$ visualization result. For example, the road G94 has a higher road traffic flow than the Minzhi Avenue and Meilong Avenue, as different colors shown in Fig. 7(L).

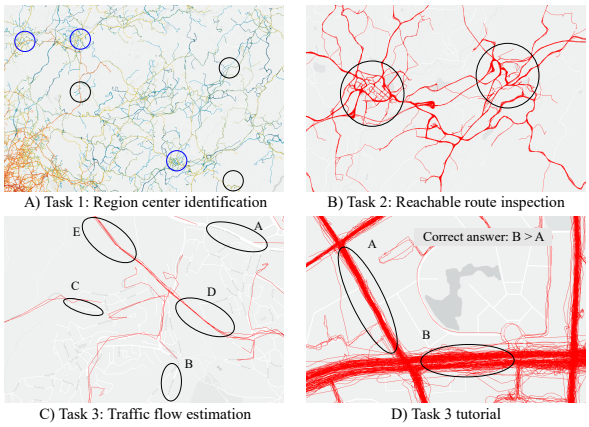


Fig. 8. Three tasks in user study

6.2 User Study

In this section, we conduct an extensive user study on three real-world applications, i.e., region center identification, reachable route inspection, and traffic flow comparison, to demonstrate the superiority of our proposal. We present our user study setting in Section 6.2.1, and analyze the user study results in Section 6.2.2.

6.2.1 User study setting

Participants and apparatus: We recruit 186 participants (24 females, 162 males, aged 18 to 29 with mean=21.16, standard derivation =1.48) with normal vision or normal corrected vision. All participants have the background of computer science. The user study system is a web-based platform, all visualized images displayed are with size 450*300. All participants perform the user study on their own computers.

Studied visualization results: We use the taxi trajectory dataset of Porto and Shenzhen for the user study. We study the visualization results which generated by different approaches in three real-world tasks. We first introduce the studied data generation methods, then elaborate the tasks shortly.

The visualization results we investigated in user study are: (i) full dataset FULL, (ii) the result sets of RAND (see Algorithm 1), (iii) the result sets of VFGS with performance optimizations (see Algorithm 2), (iv) the result sets of VFGS$^+$ without color encoding, and (iv) the result sets of VFGS$^+$ with color encoding (see Algorithm 3), denotes

as VFGS$^+$CE. The sampling rate is $\alpha = 0.5\%$ and perception tolerance value $\delta = 64$ in all the visualization results in the user study section. In each task, we use the identical regions with different visualized trajectories, which are returned from the above approaches.

User study tasks: All participants perform three tasks: (T1) region center identification, (T2) reachable route inspection, and (T3) traffic flow estimation.

(T1) region center identification. The center of the city or commercial region plays an important role in traffic management. Consequently, the passing taxi trajectories of these centers are more than its surrounding regions, and results in a star-shape cluster of trajectories in the visualization. In this task, we randomly select 6 different regions which include city or commercial centers from Porto and Shenzhen. For each region, we ask the participants to identify the correct city/region center(s) in it. As shown in Fig. 8(A), it asks the participants to identify 3 correct centers among these 6 cycles by clicking the corresponding cycles. Specifically, T1 has 30 visualization views in total. For each region of the visualization views, we label the locations of the centers in it as the correct centers at first. We then randomly select other areas which are not centers and far away from the correct centers, and label them as incorrect ones. The number of correct centers in each visualization view is given to all participates.

(T2) reachable route inspection. Intuitively, the visualized trajectories indicate the reachable routes of different regions. In this task, we give a visualization view with two circular areas, as illustrated in Fig. 8(B), then ask the participates to inspect the representative reachable routes between these two areas. We assume the more trajectories in that route, the more representative it is. For each visualization view, the number of reachable routes is given. In T2, we randomly select 7 visualizations of 7 different regions which include two or more cities/commercial districts. In each region, we choose the identical two circular areas randomly for its 5 different visualization results.

(T3) traffic flow comparison. In practice, a road with large traffic flow has many passing trajectories, thus results in a denser and broader trajectory brunch in the visualization. In the trajectory visualization with color encoding, such kind of pattern can also be highlighted by a concentration of trajectories with dark color. In this task, we ask the participates to compare the traffic flows in two roads by the given visualization results, as shown in Fig. 8(C). In particular, the participants are asked to choose the road with larger traffic flow by clicking the radio box. They also can choose "I am not sure" if they cannot decide the answer. T3 includes 5 randomly selected regions and each of them has clear road structure. It has 25 visualization views and 60 road pair comparison in total. We count the number of passing through trajectories in each road as the exact traffic flow in it.

User study procedure: In the user study system, we first provide a brief introduction about the motivation, tasks and visual encoding scheme, then followed by three tasks. In each task, we include a tutorial (with correct result) to help the participants familiarizing themselves with the interface, interaction and tasks. For example, Fig. 8(D) shows a tutorial of T3 in Fig. 8(C), where the traffic flow in road A is smaller than it in road B, as the correct answer shown. We then randomly choose different views with different questions in each task for different participates. After they completed all the questions, their answers are collected and saved in the database for the result analysis. At last, a post-interview was conducted to collect the feedback of the participants. To evaluate the answers given by the participants, we refer the reviewers to our supplementary video for details of our user study procedure.

6.2.2 User study result analysis

Fig. 9 depicts the average accuracy of all five approaches applied in the three specified tasks.

For (T1) region center identification, the average accuracy of our proposed approaches (i.e., VFGS, VFGS$^+$, and VFGS$^+$CE) is higher than its of RAND algorithm. Moreover, our proposed approaches have a very close similar performance with visualizing the full dataset FULL. It means the visualized returning results of our proposed approaches

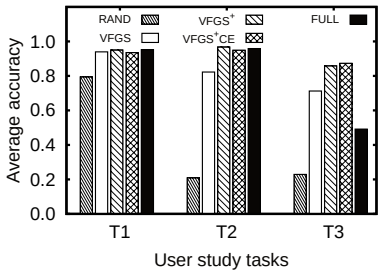
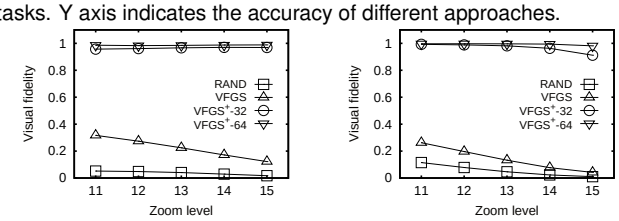


Fig. 9. Average accuracy of three user studied tasks. X axis shows three tasks. Y axis indicates the accuracy of different approaches.



(A) Porto (B) Shenzhen

Fig. 10. Visual fidelity of proposed approaches

worked as excellent as the full data visualization for the exploration of human activity center application. We observed that the performance of VFGS$^+$CE is slightly worse than performances of VFGS$^+$ and FULL. In the post-interview, some of the participants say that the color of trajectories may distract user's attentions and make the cluster characteristics not obvious.

For the reachable route inspection study in T2, it is no doubt the RAND has the worst performance among these 5 visualization approaches as it lost many (if not all) detail information. Unlike center identification, the reachable route inspection are always performed at a fine-grained level of visualization, which requires good preservation of the detail information, especially, for the sparse regions with few trajectories. Thus, the advantages of our advanced approaches VFGS$^+$ and VFGS$^+$CE over VFGS become obvious and clear. It owes to our advance approaches taken the data distribution and perception tolerance into consideration explicitly.

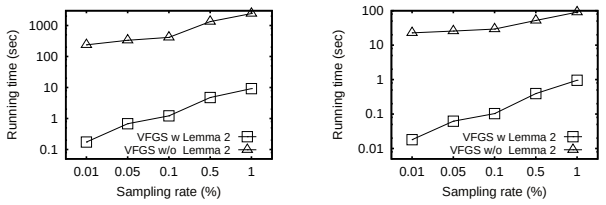
Visually, the task of traffic flow comparison in T3 are more difficult than T1 and T2, which results in relative lower average accuracy for all approaches. As expected, RAND is the worst. Interestingly, the average accuracy of visualization views of FULL is lower than our proposed approaches, i.e., VFGS, VFGS$^+$ and VFGS$^+$CE. In the post-interview, the participants pointed out that many visualization views of FULL dataset had serious visual clutter, which made it is impossible to compare the traffic flows in the two road segments. The average accuracy of our proposed VFGS$^+$ shows VFGS$^+$ alleviated the visual clutter problem and preserved the clear structure. VFGS$^+$CE further highlighted the crowded road segments from the surroundings by color, which resulted it has the highest average accuracy in the task T3.

In summary, the qualitative user study of our proposal demonstrates the effectiveness of VFGS$^+$ for large trajectory visualization by three real-world tasks. All of our proposals (VFGS, VFGS$^+$, and VFGS$^+$CE) outperform the RAND approach significantly. In addition, the participates achieved equivalent or higher accuracy score in VFGS$^+$ when comparing with the visualization of full dataset FULL.

6.3 Qualitative Evaluation

In this section, we conduct qualitative evaluation of our proposals on Porto and Shenzhen trajectory datasets from two aspects: (i) the visual fidelity in different zoom levels, and (ii) the running time with different sampling rates.

Visual fidelity evaluation: We first evaluate the visual fidelity of our proposed methods. We measure the visual fidelity of different approaches over the full dataset FULL by using the *loss*() function we defined in Section 3.1. Fig. 10 (A) and (B) shows the visual fidelity of RAND, VFGS, VFGS$^+$ with $\delta = 32$ and VFGS$^+$ with $\delta = 64$ from



(A) Porto (B) Shenzhen

Fig. 11. The running time of VFGS with or without optimization techniques

zoom level 11 to 15(overview to detail view) in Porto and Shenzhen, respectively. We can conclude that: (i) RAND approach did not guarantee the visual fidelity of the result; (ii) even VFGS offers theoretical visual fidelity guarantee w.r.t. the optimal sampled result set with given sampling rate, but it still has room for improving over the FULL dataset; (iii) VFGS$^+$ with $\delta = 32$ and $\delta = 64$ has excellent visual fidelity w.r.t. the FULL dataset. The minimum visual fidelity value is 0.95 and 0.91 in Porto and Shenzhen, respectively. It also confirms the superiority of our proposal; and (iv) the visual fidelity of VFGS$^+$ falls with the rising of zoom levels, e.g., from zoom level 11 to 15. The reason is the higher zoom level, the more detail information are required.

Running time evaluation: Last, we report the running time of our VFGS on two datasets: Porto and Shenzhen by varying the sampling rate from 0.01% to 1%. It is no doubt our visual fidelity guaranteed sampling approach VFGS is quite slow without the performance optimizations in Section 4.3. Our optimized VFGS (e.g., VFGS with Lemma 2) outperform VFGS by one to three orders of magnitudes in both Porto and Shenzhen, as shown in Fig. 11(A) and (B). Finally, with excellent performance of our VFGS, we conclude that our proposals support interactive visualization for large trajectory data exploration, i.e., generate visualization results within seconds.

7 Conclusion and Future Work

In this paper, we present a novel sampling technique VFGS$^+$ which guarantees the visual fidelity at overview and reduce the visual clutter at the detail views. We evaluate the effectiveness of the proposed method by applying our approaches to different dataset and conducting extensive user studies on three trajectory exploration tasks. There are several promising future directions, e.g., i) improving the visual fidelity by considering the trajectory segments sampling instead of the whole trajectories; and ii) developing advance color encoding schemas to present the spatial distribution of trajectories more precisely.

References

[1] Apache spark. https://spark.apache.org/.

[2] D3. https://d3js.org/.

[3] Google map. https://www.google.com/maps/preview.

[4] The open-source graphical library. https://processing.org.

[5] Porto taxi trajectory dataset. http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html.

[6] Set cover maximization. https://en.wikipedia.org/wiki/Maximum_coverage_problem.

[7] Shenzhen taxi trajectory dataset. http://jtys.sz.gov.cn/.

[8] Spotfire. https://www.tibco.com/products/tibco-spotfire.

[9] Tableau. https://www.tableau.com/.

[10] Visual clutter. https://infovis-wiki.net/wiki/Visual_Clutter.

[11] P. K. Agarwal, K. Fox, K. Munagala, A. Nath, J. Pan, and E. Taylor. Subtrajectory clustering: Models and algorithms. In *PODS*, pp. 75–87, 2018.

[12] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *2008 IEEE symposium on visual analytics science and technology*, pp. 51–58. IEEE, 2008.

[13] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualizaton. In *2013 IEEE International Conference on Big Data*, pp. 1–8. IEEE, 2013.

[14] G. Borruso. Network density estimation: a gis approach for analysing point patterns in a network space. *Transactions in GIS*, 12(3):377–402, 2008.

[15] J. Chae, D. Thom, Y. Jang, S. Kim, T. Ertl, and D. S. Ebert. Public behavior response analysis in disaster events utilizing visual analytics of microblog data. *Computers & Graphics*, 38:51–60, 2014.

[16] S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan. Maintaining interactivity while exploring massive time series. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pp. 59–66. IEEE, 2008.

[17] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma. Visual abstraction and exploration of multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1683–1692, 2014.

[18] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015.

[19] 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, pp. 201–210. Wiley Online Library, 2013.

[20] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE transactions on visualization and computer graphics*, 19(12):2149–2158, 2013.

[21] D. Guo. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1041–1048, 2009.

[22] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *2011 IEEE Pacific Visualization Symposium*, pp. 163–170. IEEE, 2011.

[23] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE transactions on visualization and computer graphics*, 15(6):1017–1024, 2009.

[24] R. Krüger, D. Thom, M. Wörner, H. Bosch, and T. Ertl. Trajectorylenses–a set-based filtering and exploration technique for long-term trajectory data. In *Computer Graphics Forum*, vol. 32, pp. 451–460. Wiley Online Library, 2013.

[25] O. D. Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *2011 IEEE Pacific visualization symposium*, pp. 171–178. IEEE, 2011.

[26] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE transactions on visualization and computer graphics*, 23(1):1–10, 2016.

[27] S. Liu, J. Pu, Q. Luo, H. Qu, L. M. Ni, and R. Krishnan. Vait: A visual analytics system for metropolitan transportation. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1586–1596, 2013.

[28] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1328–1343, 2011.

[29] Y. Park, M. Cafarella, and B. Mozafari. Visualization-aware sampling for very large databases. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 755–766. IEEE, 2016.

[30] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis. Similarity search in trajectory databases. In *14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, pp. 129–140. IEEE, 2007.

[31] N. Pelekis, I. Kopanakis, C. Panagiotakis, and Y. Theodoridis. Unsupervised trajectory sampling. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 17–33. Springer, 2010.

[32] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.

[33] S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. Visually driven analysis of movement data by progressive clustering. *Information Visualization*, 7(3-4):225–239, 2008.

[34] R. Scheepens, N. Willems, H. van de Wetering, and J. J. van Wijk. Interactive visualization of multivariate trajectory data with density maps. In *2011 IEEE Pacific Visualization Symposium*, pp. 147–154. IEEE, 2011.

[35] B. Tang, M. L. Yiu, K. Mouratidis, and K. Wang. Efficient motif discovery in spatial trajectories using discrete fréchet distance. EDBT, 2017.

[36] M. Thöny and R. Pajarola. Vector map constrained path bundling in 3d environments. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on GeoStreaming*, pp. 33–42, 2015.

[37] 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*, 22(1):11–20, 2015.

[38] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, and Q. Wu. Visual exploration of sparse traffic trajectory data. *IEEE transactions on visualization and computer graphics*, 20(12):1813–1822, 2014.

[39] J. Wood, J. Dykes, and A. Slingsby. Visualisation of origins, destinations and flows with od maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[40] Z. Xie and J. Yan. Kernel density estimation of traffic accidents in a network space. *Computers, environment and urban systems*, 32(5):396–406, 2008.

[41] C. Xu, B. Tang, and M. L. Yiu. Diversified caching for replicated web search engines. In *ICDE*, pp. 207–218. IEEE, 2015.

[42] C. Yang, Y. Zhang, B. Tang, and M. Zhu. Vaite: A visualization-assisted interactive big urban trajectory data exploration system. In *ICDE*, pp. 2036–2039, 2019.

[43] X. Yang, Z. Zhao, and S. Lu. Exploring spatial-temporal patterns of urban human mobility hotspots. *Sustainability*, 8(7):674, 2016.

[44] W. Zeng, C.-W. Fu, S. M. Arisona, and H. Qu. Visualizing interchange patterns in massive movement data. In *Computer Graphics Forum*, vol. 32, pp. 271–280. Wiley Online Library, 2013.

[45] W. Zeng, Q. Shen, Y. Jiang, and A. Telea. Route-aware edge bundling for visualizing origin-destination trails in urban traffic. In *Computer Graphics Forum*, vol. 38, pp. 581–593. Wiley Online Library, 2019.

[46] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pp. 1029–1038, 2010.

[47] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):1–29, 2011.