

# Visual Fidelity Guaranteed Sampling for Large Trajectory Data Visualization

Qiaomu Shen<sup>†</sup>, Chuan Yang<sup>†</sup>, Chaozu Zhang<sup>†</sup>, Dan Zeng<sup>‡</sup>, Wei Zeng<sup>\$</sup>, Bo Tang<sup>†</sup>

<sup>†</sup> Department of Computer Science and Engineering, Southern University of Science and Technology

qshen@connect.ust.hk, {11612732@mail,11712021@mail,tangb3@}{sustech.edu.cn}

<sup>‡</sup> Electrical Engineering, Mathematics and Computer Science, University of Twente

<sup>\$</sup> Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

d.zeng@utwente.nl, zengwei81@gmail.com

## ABSTRACT

Visualizing large-scale trajectory data is the core subroutine in many smart city applications, e.g., traffic management, route recommendation. However, it suffers from limited rendering capability and visual clutter issues. Sampling can effectively mitigate the issues, yet existing methods have an attenuating effect on visual fidelity. In this work, we propose visual fidelity guaranteed sampling techniques for line-based visualization of large trajectories. We first define a pixel-based 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 propose VFGS<sup>+</sup> which improves the effectiveness of VFGS by taking data distribution and human perception into consideration. We conduct extensive experimental studies to demonstrate the effectiveness and efficiency of our methods on real-world trajectory datasets. In addition, comprehensive user studies further illustrate the superiority of our proposals in various applications, e.g., traffic flow comparison, and reachable route inspection.

### PVLDB Reference Format:

Qiaomu Shen, et al.. Visual Fidelity Guaranteed Sampling for Large

Trajectory Data Visualization. PVLDB, 14(1): XXX-XXX, 2020.

doi:XX.XX/XXX.XX

## 1 INTRODUCTION

Nowadays, the ubiquitous location-acquisition devices lead to an explosive increase of movement data (a.k.a. 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 [33], urban planning [31], route recommendation [39], and location-based services [26, 38]. Line-based trajectory visualization, i.e., connecting the locations of a moving object by polylines, is a popular and conventional visualization method [19]. However, large-scale line-based trajectory visualization is challenging. The reasons are (i) large trajectory data size and limited rendering capability of graphics device, and

(ii) visual clutter in large data visualization. We elaborate them as follows:

**Large trajectory data size and limited rendering capability of graphics device:** 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 [21]. However, due to the hardware limitation, the rendering capability of modern commodity GPUs is limited. We did a benchmark experiment to evaluate the rendering capability of NVIDIA GeForce GTX 1080 with 8GB video memory. It needs 13.95 seconds to render 1 million trajectories in Porto. Obviously, it cannot support interactive visual exploration in large-scale trajectory dataset.

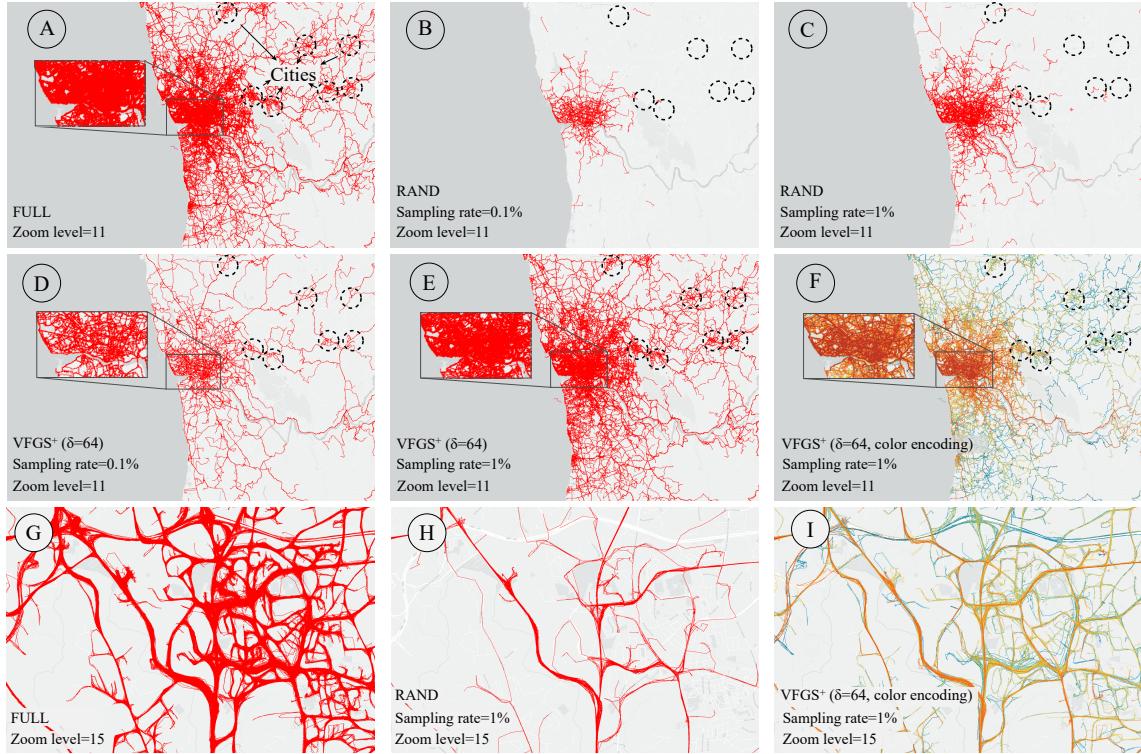
**Visual clutter in large data visualization:** Visual clutter is a common issue in data visualization [11]. Figure 1(A) is the visualization result of the full Porto taxi trajectory dataset. Intuitively, the region shown in the embedded figure of A suffers visual clutter issue seriously, i.e., the road network almost cannot be recognized in it, which hinders the abilities of human-users to explore the dataset and identify the underlying data insights.

Sampling techniques are de-facto standards for large-scale data analysis in both database and visualization communities. 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 [14] employs a reduction layer between the visualization layer and the data management layer. The reduction layer embeds 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 the large trajectory data visualization problem as it does not have any guarantees about the sampling results. Take Figure 1(B) and (C) as examples, they are the visualization results of uniform random sampling method (RAND) on Porto taxi trajectory dataset with sampling rate 0.1% and 1%, respectively. Visually, both visualized results cannot capture the overview of the Porto trajectory dataset, as shown in Figure 1(A).

In database community, Park et al. devised a visualization-aware sampling algorithm (VAS) for large-scale scatter points visualization in [28], 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 [28] is

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.  
doi:XX.XX/XXX.XX



**Figure 1: Visualization results comparison. (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<sup>+</sup> with different parameters. (II) G is visualization of the full Porto dataset at zoom level 15, where H and I are the corresponding visualized results of random sampling and VFGS<sup>+</sup> (with sampling rate 1%), respectively.

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. 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 propose a novel pixel-based 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 by taking data distribution and human perception capability into consideration in the advance approach (VFGS<sup>+</sup>). Figure 1(D) and (E) show the visualization results of our proposal VFGS<sup>+</sup> on Porto taxi trajectory dataset with the sampling rate 0.1% and 1%, respectively. Obviously, the visualization fidelity of them are much better than the uniform random sampling visualization results with the same sampling rates, see Figure 1(B) and (C). Figure 1(F) is the returning result of our proposal which colors the trajectories according to the trajectory representativeness. It has the same parameters of

Figure 1(E). Visually, the visual clutter in Figure 1(A) and (E) are alleviated in Figure 1(F). In addition, our proposals are robustness with different zoom levels. Figure 1(G), (H), and (I) depict the visualization results of the Porto dataset, the returning result of uniform random sampling RAND and the returning result of VFGS<sup>+</sup> with color encoding at zoom-level 15, for example, we can obtain them by zooming in the visualization result in Figure 1(A), (C), and (F), respectively. Intuitively, the visualization result of our proposal VFGS<sup>+</sup> in Figure 1(F) outperforms the uniform random sampling method RAND in Figure 1(H) significantly. It even performs better than Figure 1(G), the visualized result of the Porto dataset, as it reduces visual clutter in Figure 1(G) by using color encoding scheme to capture the representativeness of different trajectories.

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 in Section 3.
- We devise an approximation algorithm VFGS with a suite of optimization techniques, e.g., submodularity, lazy computing, for it in Section 4.
- We propose an advance approach VFGS<sup>+</sup> to further enhance the effectiveness of our approximate algorithm, which addresses the visual clutter by introducing perception tolerance parameter, and encodes the representativeness of each 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 comprehensive user studies to show the effectiveness on three real-world applications.

## 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). As stated in [19], 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 them and refer the interested readers to a recent survey [19] 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 [15, 16, 27, 35, 37]. The region-based visualization approaches divide the whole region into sub-regions in advance, then visualize the aggregated information in each sub-region [22, 32, 34]. 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 [23, 24]. 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 [20, 30, 32] and advanced interaction techniques [21, 25]. 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 advancement of location-acquisition technology, the size of available trajectory dataset becomes extremely huge. For example, the operating taxis in Shenzhen generate ~9.3GB trajectory data per day. Figure 2 illustrates the architecture of interactive visualization systems for large datasets, e.g., Spotfire [8], Tableau [9], ATLAS [17], and Viate [36]. It 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 focus on improving the effectiveness of data visualization at the front-end, e.g., designing novel visualization methods to assist data analysts to obtain data insights effectively (D3 [2]). For

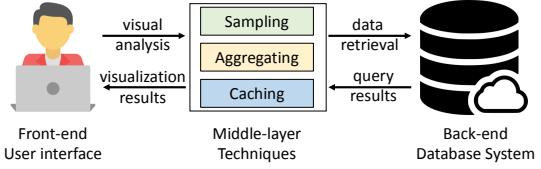


Figure 2: Interactive visualization system architecture.

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]). 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 Figure 2). We briefly elaborate these optimization techniques in this section.

**Aggregating-based techniques:** many works [22, 32, 34] divide the spatial space into basic units, then visualize the information upon them by aggregation algorithms for trajectory visual analysis tasks. For more details of other aggregating-based techniques, we refer the reader to the related works [12, 13]. 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 widely used in both visualization and database communities [14, 18, 28]. The most relevant work of ours in the literature is [28], 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. However, the techniques in [28] cannot be adapted to our large-scale trajectory visualization problem as trajectory data is more complex than scatter points.

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:** Chan et al. present ATLAS [17] 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. [29] 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. 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 the hardness of our research problem.

### 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  $\alpha$ , the trajectory visualization problem is selecting a subset of trajectories  $R \subseteq T$  with  $|R| \leq \alpha|T|$ , such that visual fidelity loss function  $loss(R, T)$  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  $R \subseteq T$ , 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  $loss(T, R) = \frac{|V(T) - V(R)|}{|V(T)|}$ , where  $V()$  measures the rendered pixels in the canvas of the input trajectory dataset.

Therefore, given a trajectory data set  $T$  and a sampling rate  $\alpha$ , our research objective is finding subset  $R$ , such that the visualization fidelity loss function  $loss(T, R)$  is minimized:

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

### 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 exist complex rendering schemes, e.g., different pixels has different colors, we will consider them 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  $t_i \in T$ , it consists of a set of pixels in the canvas. In other words, the pixel set of each trajectory  $t_i \in T$  is a subset of  $U$ . Consequently, the pixel set of the selected trajectory set  $R$  is also a subset of  $U$  as  $R = \bigcup_{t_i \in R} t_i$ .

Our research objective is minimizing loss function  $loss(T, R) = \frac{|V(T) - V(R)|}{|V(T)|}$  subject to  $R \subseteq T$  and  $|R| \leq \alpha|T|$ . 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:

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

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

## 4 OUR SOLUTION: VFGS

Due to the hardness of the Problem 1, we first introduce a straightforward solution, i.e., uniform random sampling RAND. Specifically, it randomly selects  $k$  trajectories from dataset  $T$  and stores them in result set  $R$ . Last, we render these selected trajectories in  $R$  as the visualization result. Obviously, the uniform random sampling RAND algorithm does not provide any guarantee on the visual fidelity of the sampled result set.

In this section, we next propose a visual fidelity guaranteed sampling method VFGS in Section 4.1. Last, we devise several optimizations to improve the efficiency of our proposal VFGS in Section 4.2.

### 4.1 Visual Fidelity Guaranteed Sampling

In this section, we present our visual fidelity guaranteed sampling algorithm for Problem 1. We first elaborate the correlation between visual fidelity of sampled set  $R$  and user zoom level. For a given result sampled set  $R \subseteq T$ , 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  $t_i \in T$ , it is a set of pixels in the canvas at the highest zoom level.

With the above setting, we next describe our visual fidelity guaranteed sampling algorithm in Algorithm 1, which employs greedy paradigm. In particular, it finds the trajectory  $tmp$  in  $T$  which maximize the result set of  $|R \cup tmp|$  at each iteration, as Line 3 shown in Algorithm 1. It terminates after  $k = \alpha|T|$  iterations and returns  $R$  as result set for rendering.

---

#### Algorithm 1 VFGS( $T, k = \alpha|T|$ )

---

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

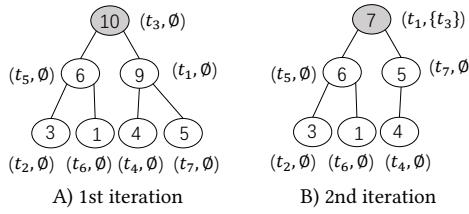


Figure 3: Lazy computing manner.

Interestingly, Algorithm 1 offers theoretical visual fidelity guarantee of the returning result  $R$ , as proved in Theorem 1<sup>1</sup>.

**THEOREM 1.** *Algorithm 1 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).*

## 4.2 Optimization Techniques

With the above analysis, Algorithm 1 (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  $k = \alpha|T|$ , the time complexity of Algorithm 1 is  $O(\alpha \cdot m \cdot |T|^2)$ , where  $m$  is the maximum length of all trajectories in dataset  $T$ .*

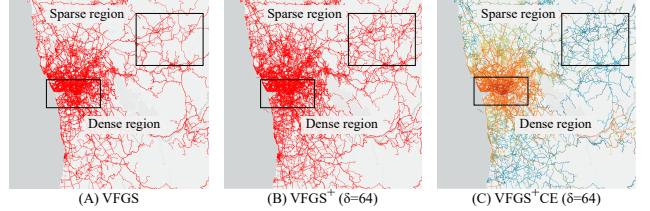
**Example:** Given Porto trajectory dataset, it has 2.39 millions of taxi trajectories, the maximum length in it is 3,490. It takes 413.6 seconds to return a subset  $R$  with sampling rate 0.1%. Obviously, it is impractical for interactive trajectory explorations.

Due to the inefficient of our visual fidelity guaranteed sampling approach in Algorithm 1, 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  $\Delta(R, t) = |R \cup t| - |R|$ . Given a trajectory  $t$  and two result sets  $R, R'$ , where  $R \subset R'$  and  $t \notin R$ , it holds  $\Delta(R, t) \geq \Delta(R', t)$ .*

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. Figure 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  $R = \emptyset$ . At the 1st iteration, the root node of the max-heap will be selected, i.e.,  $t_3$  in Figure 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  $R = \{t_3\}$  (see gray node at Figure 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 Figure 3(B). The reason is the contribution of the trajectories in all white nodes will be less than 7 according to the submodularity in Lemma 2.

<sup>1</sup>We omitted all the proofs of the theorems and lemmata in this work due to space reasons, and refer the interested readers to our technical report[10].

Figure 4: Advance approach VFGS<sup>+</sup> with Porto ( $\alpha = 0.5\%$ ).

The performance of Algorithm 1 is improved significantly as we only compute its contribution values when it is necessary. To exemplify, Algorithm 1 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<sup>+</sup>

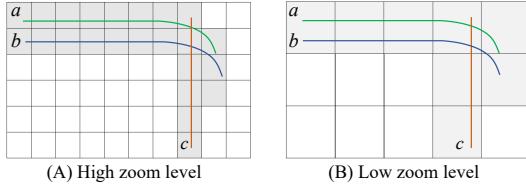
Until now, VFGS in Algorithm 1 offers a visual fidelity guaranteed sampling approach for large-scale trajectory visualization problem (see Problem 1), which returns the visual fidelity guaranteed result efficiently via the optimization techniques in Section 4.2. It means that the challenges (i) large 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<sup>+</sup> to alleviate it by considering (i) trajectory data distribution, and (ii) human perception capability. We elaborate (i) and (ii) by the examples in Figure 4 shortly.

**Trajectory data distribution:** Considering Porto trajectory dataset, Figure 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 Figure 4(A).

**Human perception capability:** Intuitively, it is much easier for humans to distinguish the difference between sparse regions rather than dense regions in Figure 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 large enough with a given level of details, i.e., the difference between two dense regions in Figure 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<sup>+</sup> to achieve the above two objectives. Specifically, we introduce perception tolerance parameter  $\delta$  in VFGS<sup>+</sup>, 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 1 to incorporate the perception tolerance parameter  $\delta$  in advance approach VFGS<sup>+</sup>. It measures the contribution of each trajectory  $t_i$  w.r.t the



**Figure 5: VFGS<sup>+</sup> with different zoom levels.**

selected trajectory set R's augmented set R<sup>+</sup>, i.e., the selected trajectories and their tolerance pixels.

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<sup>+</sup>) with colors. In particular, VFGS<sup>+</sup> 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 1). 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 Figure 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<sup>+</sup> 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, and visualize them by encoding with different colors. Figure 4(C) shows the visualized result of the advance approach VFGS<sup>+</sup> 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<sup>+</sup> 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  $\delta$ . Take Figure 5 as an example, the zoom level in Figure 5(A) is higher than it in Figure 5(B). As our above elaboration, our advance approach VFGS<sup>+</sup> selects trajectory  $a$  and  $c$  at Figure 5(A). When it zoomed out, as shown in Figure 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. 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. Last, we perform a qualitative evaluation in Section 6.3 at last. We conducted all experiments on a machine with Intel i7-8700, 3.2 GHz CPU, 24 GBytes memory and NVIDIA GeForce GTX1080, 8 GHz 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

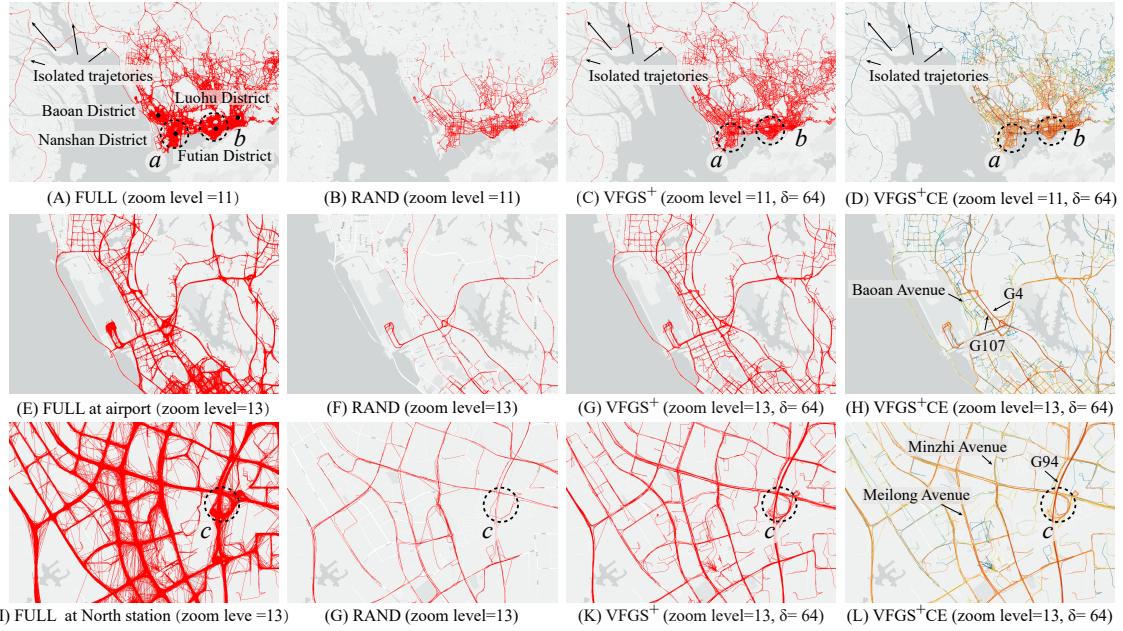
We demonstrate the effectiveness of our proposal by the case studies in Porto in Section 6.1.1 and Shenzhen in Section 6.1.2.

**6.1.1 Case Studies on Porto.** We elaborate the cases in Figure 1 to illustrate the effectiveness of our proposal from the following three aspects. For the sake of page limitation, we refer the interested reader to our technical report [10] for other cases in Porto dataset.

**Effect of approaches with different zoom-levels:** Considering zoom level 11, Figure 1(A) is the visualization result of full Porto trajectory dataset. Given the sampling rate  $\alpha = 1\%$ , Figure 1(C) and (E) visualized the returning results of uniform random sampling algorithm (RAND, see Algorithm ??) and our advanced visual fidelity guaranteed sampling approach (VFGS<sup>+</sup>, see Algorithm ??), respectively. Obviously, Figure 1(E) looks much more closer to Figure 1(A) when comparing with Figure 1(C). In particular, our proposed VFGS<sup>+</sup> 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 Figure 1(E). However, the details of these dashed cycles in Figure 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. Figure 1(H) and (I) are the visualization result of RAND and VFGS<sup>+</sup> (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 Figure 1(G), the RAND in Figure 1(H) only returns few trajectories and many information in raw data are lost. Surprisingly, our VFGS<sup>+</sup> in Figure 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. Figure 1(B) and (C) are the visualization results of RAND with sampling rate 0.1% and 1%, respectively. Figure 1(D) and (E) visualized the returning trajectory sets of our VFGS<sup>+</sup> 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., Figure 1(C) and (E) are more closer to Figure 1(A) when comparing with Figure 1(B) and (D), respectively; (ii) the visualization result of VFGS<sup>+</sup> with sampling rate 0.1% in Figure 1(D) performs even more better than the result of RAND with sampling rate 1% in Figure 1(C) to capture the visual structure of the full dataset in Figure 1(A).

**Effect of color encoding:** Next, we present the superiority of color encoding scheme in VFGS<sup>+</sup>, which denotes as VFGS<sup>+</sup>CE in subsequent sections. Given zoom level 11 and sampling rate 1%, Figure 1(E) and (F) are the visualization results of our VFGS<sup>+</sup> and VFGS<sup>+</sup>CE (i.e., VFGS<sup>+</sup> with color encoding), respectively. It is



**Figure 6: Case studies on Shenzhen taxi trajectory dataset, sampling rate  $\alpha = 1\%$ .**

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

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

**Overview of Shenzhen:** Figure 6(A) is the visualization result of full Shenzhen dataset at zoom level 11. The dense regions in southern of Shenzhen, as the dashed circles shown in Figure 6(A), are *Baoan*, *Nanshan*, *Futian* and *Luohu* districts, which are the most prosperous commercial regions in this city. The returning results of RAND, VFGS<sup>+</sup> and VFGS<sup>+CE</sup> are visualized in Figure 6(B), (C) and (D), respectively. Not surprisingly, the visualized result of RAND in Figure 6(B) is quite different from the full dataset in Figure 6(A). VFGS<sup>+</sup> in Figure 6(C) shows its superiority by capturing the overview of Shenzhen dataset and even preserves the isolated trajectories, as highlighted in left-upper corner of Figure 6(C). It owes to VFGS<sup>+</sup> provides theoretical visual fidelity guarantees on the returning result set. VFGS<sup>+</sup> with color encoding VFGS<sup>+CE</sup> further improved the visual fidelity of VFGS<sup>+</sup>. Specifically, both Figure 6(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 trajectories. However, the result of VFGS<sup>+</sup> with

color encoding, as shown in Figure 6(D), reduce the visual clutter perfectly. For example, it is clear that the main roads of circle *a* and *b* are these roads with darker colors in Figure 6(D).

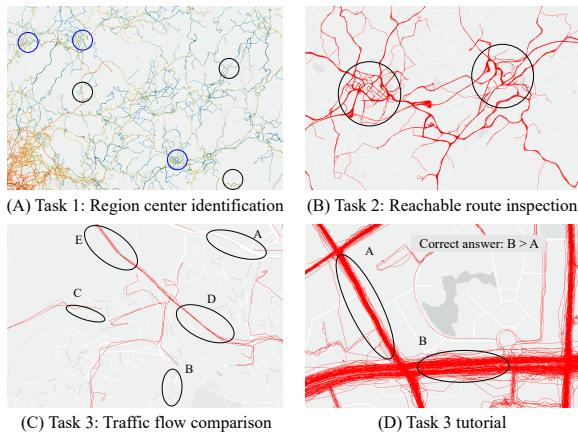
We then present the advantages of our VFGS<sup>+</sup> in two representative areas, i.e., airport and North railway station, in Shenzhen dataset.

**Airport in Shenzhen:** Comparing with visualization result of full dataset in Figure 6(E), the visualized result of RAND in Figure 6(F) only includes very few trajectories. Both VFGS<sup>+</sup> and VFGS<sup>+CE</sup> (see Figure 6(G) and (H)) reserve the major structure of the airport area excellent. Moreover, VFGS<sup>+CE</sup> 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 highlighted in Figure 6(H).

**North railway station in Shenzhen:** We next investigate the visualizations of the full dataset, uniform random sampling result set, and visual fidelity guaranteed sampling result set around North railway station of Shenzhen, which are shown from Figure 6(I) to (L). Interestingly, VFGS<sup>+</sup> and VFGS<sup>+CE</sup> visualized the overpass near North railway state clearly, as circle *c* shown in both Figure 6(K) and (L). Due to visual clutter, the overpass is not clear in Figure 6(I), which visualized the full dataset. It even disappeared in the visualized result of RAND in Figure 6(G). Moreover, it is easy to compare the traffic flows in different roads via VFGS<sup>+CE</sup> 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 Figure 6(L).

## 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



**Figure 7: Three tasks in user study.**

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 recruited 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 displayed visualization views are with size 450\*300. All participants performed 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 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 ??), (iii) the result sets of VFGS with performance optimizations (see Algorithm 1), (iv) the result sets of VFGS<sup>+</sup> without color encoding, and (iv) the result sets of VFGS<sup>+</sup> with color encoding (see Algorithm ??), i.e., VFGS<sup>+</sup>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 performed three tasks: (T1) region center identification, (T2) reachable route inspection, and (T3) traffic flow comparison.

**(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 result in a star-shape cluster of trajectories in the visualization. In this task, we randomly selected 6 different regions which include city or commercial centers from Porto and Shenzhen. For each region, we asked the participants to identify the correct city/region center(s) in it. As shown in Figure 7(A), it asked the participants to identify 3 correct centers among these 6 cycles by clicking the corresponding cycles. Specifically, T1 had 30 visualization views in total. For each region

of the visualization views, we labelled the locations of the city or commercial region centers in it as the correct centers at first. We then randomly selected other locations far away from the correct centers, and labelled them as incorrect ones. We fixed the number of correct centers in each visualization view as 3.

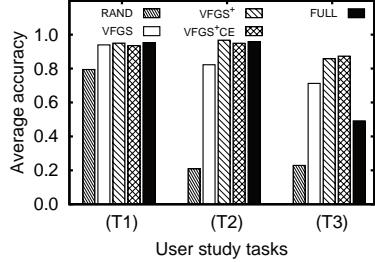
**(T2) reachable route inspection.** Intuitively, the visualized trajectories indicate the reachable routes which connect different regions. In this task, we gave a visualization view with two circular areas, as illustrated in Figure 7(B), then asked the participants to inspect the representative reachable routes between these two areas. We assumed the more trajectories in that route, the more representative it is. For each visualization view, the number of reachable routes was given. In T2, we randomly selected 7 different regions, each of them includes two or more cities/commercial districts. For the 5 different visualization views of a selected region, we chose two identical circular areas in each of them randomly.

**(T3) traffic flow comparison.** In practical, a road with large traffic flow has lots of passing trajectories, thus it 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 asked the participants to compare the traffic flows in two roads by the visualization results, as shown in Figure 7(C). In particular, the participants were asked to choose the road with larger traffic flow by clicking the radio box. They also could choose “I am not sure” if they could not decide the answer. T3 included 5 randomly selected regions and each of them had a clear road structure. It had 25 visualization views and 60 road pair comparisons in total. We counted the number of passing trajectories in each road as the exact traffic flow in it.

**User study procedure:** In the user study system, we first provided a brief introduction about the motivation, tasks and visual encoding scheme, then followed by three tasks. In each task, we included a tutorial (with the correct result) to help the participants familiarizing themselves with the interface, interaction and tasks. For example, Figure 7(D) shows a tutorial of T3 in Figure 7(C), where the traffic flow in road A is smaller than it in road B, as the correct answer shown in it. We then randomly chose different views with different questions in each task for different participants. After they completed all the questions, their answers were collected and saved in the database for the result analysis. At last, post-interviews were conducted to collect the feedback of the participants. We refer the reviewers to our supplementary video for the details of our user study tasks and procedure.

**6.2.2 User study result analysis.** Figure 8 depicts the average accuracy of all five approaches applied in the three user study tasks.

For (T1) region center identification task, the average accuracy of our proposed approaches (i.e., VFGS, VFGS<sup>+</sup>, and VFGS<sup>+</sup>CE) are higher than that of RAND. 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 worked as excellent as the full data visualization for the exploration of human activity center application. We observed that the performance of VFGS<sup>+</sup>CE is slightly worse than the performances of VFGS<sup>+</sup> and FULL. In the post-interviews, some



**Figure 8: Average accuracy of three user studied tasks. X axis shows three tasks. Y axis indicates the accuracy of different approaches.**

of the participants said that the color of trajectories may distract user’s attention and make the cluster characteristics not obvious.

For (T2) reachable route inspection task, 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 region center identification in (T1), the reachable route inspection are always performed at a fine-grained level of visualization, which requires good preservation of the details, especially for the sparse regions with few trajectories. Thus, the advantages of our advance approaches VFGS<sup>+</sup> and VFGS<sup>+</sup>CE over VFGS become obvious and clear. The reason is that our advance approaches considered the data distribution and perception tolerance explicitly.

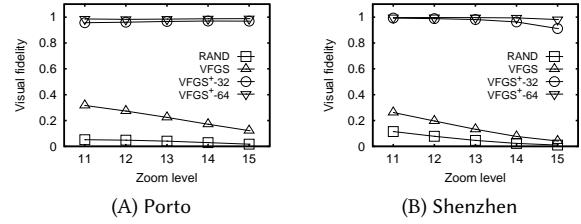
Visually, the task of traffic flow comparison in (T3) is more difficult than (T1) and (T2), which results in relatively lower average accuracy for all approaches. As expected, RAND is the worst. Interestingly, the average accuracy of visualization views of FULL is lower than our proposed approaches, i.e., VFGS, VFGS<sup>+</sup> and VFGS<sup>+</sup>CE. In the post-interviews, the participants pointed out that many visualization views of FULL 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<sup>+</sup> shows VFGS<sup>+</sup> alleviated the visual clutter problem and preserved the clear structure. VFGS<sup>+</sup>CE further highlighted the crowded road segments from the surroundings by color, which resulted in the highest average accuracy in the task (T3).

In summary, the qualitative user study of our proposal demonstrates the effectiveness of VFGS<sup>+</sup> for large trajectory visualization by three real-world tasks. All of our proposals (VFGS, VFGS<sup>+</sup>, and VFGS<sup>+</sup>CE) outperform the RAND approach significantly. In addition, the participants achieved equivalent or higher accuracy scores in VFGS<sup>+</sup> and VFGS<sup>+</sup>CE when comparing with the visualization of FULL.

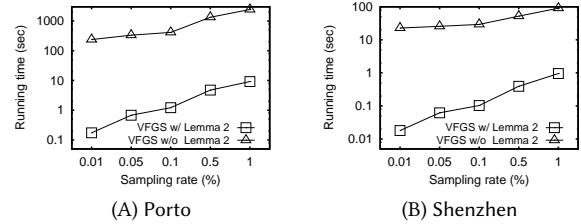
### 6.3 Qualitative Evaluation

In this section, we conduct a 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 by using the *loss()* function we defined in Section 3.1. Figure 9(A) and (B) show the visual fidelity of RAND, VFGS, VFGS<sup>+</sup> with  $\delta = 32$  and VFGS<sup>+</sup> with  $\delta = 64$  from zoom



**Figure 9: Visual fidelity of proposed approaches.**



**Figure 10: The running time of VFGS with or without Lemma 2.**

level 11 to 15 (i.e., 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 a given sampling rate, but it still has room for improving over the FULL; (iii) VFGS<sup>+</sup> 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<sup>+</sup> falls with the rising of zoom levels, e.g., from zoom level 11 to 15. The reason is the higher zoom level, the more details are expected.

**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 submodularity of contribution value, see Lemma 2 in Section 4.2. Our optimized VFGS (e.g., VFGS with Lemma 2) outperforms VFGS by one to three orders of magnitudes in both Porto and Shenzhen, as shown in Figure 10(A) and (B). Finally, with the excellent performance of our VFGS, we conclude that our proposals support interactive visualization for large trajectory data exploration, i.e., they generate visualization results within seconds.

## 7 CONCLUSION AND FUTURE WORK

In this paper, we present a novel sampling technique VFGS<sup>+</sup> 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] 2020. Apache Spark. <https://spark.apache.org/>.
- [2] 2020. D3. <https://d3js.org/>.
- [3] 2020. Google map. <https://www.google.com/maps/preview>.
- [4] 2020. The open-source graphical library. <https://processing.org>.
- [5] 2020. Porto taxi trajectory dataset. <http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html>.
- [6] 2020. Set cover maximization. [https://en.wikipedia.org/wiki/Maximum\\_coverage\\_problem](https://en.wikipedia.org/wiki/Maximum_coverage_problem).
- [7] 2020. Shenzhen taxi trajectory dataset. <http://jtys.sz.gov.cn/>.
- [8] 2020. Spotfire. <https://www.tibco.com/products/tibco-spotfire>.
- [9] 2020. Tableau. <https://www.tableau.com/>.
- [10] 2020. Technical Report. [https://acm.sustech.edu.cn/btang/vfgs\\_tr.pdf](https://acm.sustech.edu.cn/btang/vfgs_tr.pdf).
- [11] 2020. Visual clutter. [https://infovis-wiki.net/wiki/Visual\\_Clutter](https://infovis-wiki.net/wiki/Visual_Clutter).
- [12] Natalia Adrienko and Gennady Adrienko. 2010. Spatial generalization and aggregation of massive movement data. *TVCG* 17, 2 (2010), 205–219.
- [13] Gennady Adrienko and Natalia Adrienko. 2008. Spatio-temporal aggregation for visual analysis of movements. In *IEEE symposium on visual analytics science and technology*. IEEE, 51–58.
- [14] Leilani Battle, Michael Stonebraker, and Remco Chang. 2013. Dynamic reduction of query result sets for interactive visualization. In *IEEE International Conference on Big Data*. IEEE, 1–8.
- [15] Giuseppe Borruso. 2008. Network density estimation: a GIS approach for analysing point patterns in a network space. *Transactions in GIS* 12, 3 (2008), 377–402.
- [16] Junghoon Chae, Dennis Thom, Yun Jang, SungYe Kim, Thomas Ertl, and David S Ebert. 2014. Public behavior response analysis in disaster events utilizing visual analytics of microblog data. *Computers & Graphics* 38 (2014), 51–60.
- [17] Sye-Min Chan, Ling Xiao, John Gerth, and Pat Hanrahan. 2008. Maintaining interactivity while exploring massive time series. In *IEEE Symposium on Visual Analytics Science and Technology*. IEEE, 59–66.
- [18] Haidong Chen, Wei Chen, Honghui Mei, Zhiqi Liu, Kun Zhou, Weifeng Chen, Wentao Gu, and Kwan-Liu Ma. 2014. Visual abstraction and exploration of multi-class scatterplots. *TVCG* 20, 12 (2014), 1683–1692.
- [19] Wei Chen, Fangzhou Guo, and Fei-Yue Wang. 2015. A survey of traffic data visualization. *TITS* 16, 6 (2015), 2970–2984.
- [20] Nivan Ferreira, James T Klosowski, Carlos E Scheidegger, and Cláudio T Silva. 2013. Vector field k-means: Clustering trajectories by fitting multiple vector fields. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 201–210.
- [21] Nivan Ferreira, Jorge Poco, Huy T Vo, Juliana Freire, and Cláudio T Silva. 2013. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *TVCG* 19, 12 (2013), 2149–2158.
- [22] Diansheng Guo. 2009. Flow mapping and multivariate visualization of large spatial interaction data. *TVCG* 15, 6 (2009), 1041–1048.
- [23] Hanqi Guo, Zuchao Wang, Bowen Yu, Huijing Zhao, and Xiaoru Yuan. 2011. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *IEEE Pacific Visualization Symposium*. IEEE, 163–170.
- [24] Christophe Hurter, Benjamin Tissières, and Stéphane Conversy. 2009. Fromday: Spreading aircraft trajectories across views to support iterative queries. *TVCG* 15, 6 (2009), 1017–1024.
- [25] Robert Krüger, Dennis Thom, Michael Wörner, Harald Bosch, and Thomas Ertl. 2013. TrajectoryLenses—a set-based filtering and exploration technique for long-term trajectory data. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 451–460.
- [26] Dongyu Liu, Di Weng, Yuhong Li, Jie Bao, Yu Zheng, Huamin Qu, and Yingcai Wu. 2016. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *TVCG* 23, 1 (2016), 1–10.
- [27] Siyuan Liu, Jiansi Pu, Qiong Luo, Huamin Qu, Lionel M Ni, and Ramayya Krishnan. 2013. VAIT: A visual analytics system for metropolitan transportation. *TITS* 14, 4 (2013), 1586–1596.
- [28] Yongjoo Park, Michael Cafarella, and Barzan Mozafari. 2016. Visualization-aware sampling for very large databases. In *ICDE*. IEEE, 755–766.
- [29] Harald Piringer, Christian Tominski, Philipp Muigg, and Wolfgang Berger. 2009. A multi-threading architecture to support interactive visual exploration. *TVCG* 15, 6 (2009), 1113–1120.
- [30] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Adrienko, and Gennady Adrienko. 2008. Visually driven analysis of movement data by progressive clustering. *Information Visualization* 7, 3-4 (2008), 225–239.
- [31] Bo Tang, Man Lung Yiu, Kyriakos Mouratidis, and Kai Wang. 2017. Efficient motif discovery in spatial trajectories using discrete fréchet distance. In *EDBT*.
- [32] Tatiana Von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia Adrienko, Gennady Adrienko, and Andreas Kerren. 2015. Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. *TVCG* 22, 1 (2015), 11–20.
- [33] Zuchao Wang, Tangzhi Ye, Min Lu, Xiaoru Yuan, Huamin Qu, Jacky Yuan, and Qianliang Wu. 2014. Visual exploration of sparse traffic trajectory data. *TVCG* 20, 12 (2014), 1813–1822.
- [34] Jo Wood, Jason Dykes, and Aidan Slingsby. 2010. Visualisation of origins, destinations and flows with OD maps. *The Cartographic Journal* 47, 2 (2010), 117–129.
- [35] Zhixiao Xie and Jun Yan. 2008. Kernel density estimation of traffic accidents in a network space. *Computers, environment and urban systems* 32, 5 (2008), 396–406.
- [36] Chuang Yang, Yilan Zhang, Bo Tang, and Min Zhu. 2019. Vaite: A Visualization-Assisted Interactive Big Urban Trajectory Data Exploration System. In *ICDE*. IEEE, 2036–2039.
- [37] Xiping Yang, Zhiyuan Zhao, and Shiwei Lu. 2016. Exploring spatial-temporal patterns of urban human mobility hotspots. *Sustainability* 8, 7 (2016), 674.
- [38] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *WWW*. 1029–1038.
- [39] Yu Zheng and Xing Xie. 2011. Learning travel recommendations from user-generated GPS traces. *TIST* 2, 1 (2011), 1–29.