Visual Fidelity Guaranteed Sampling for Large Trajectory Data Visualization

Category: Research

Paper Type: please specify

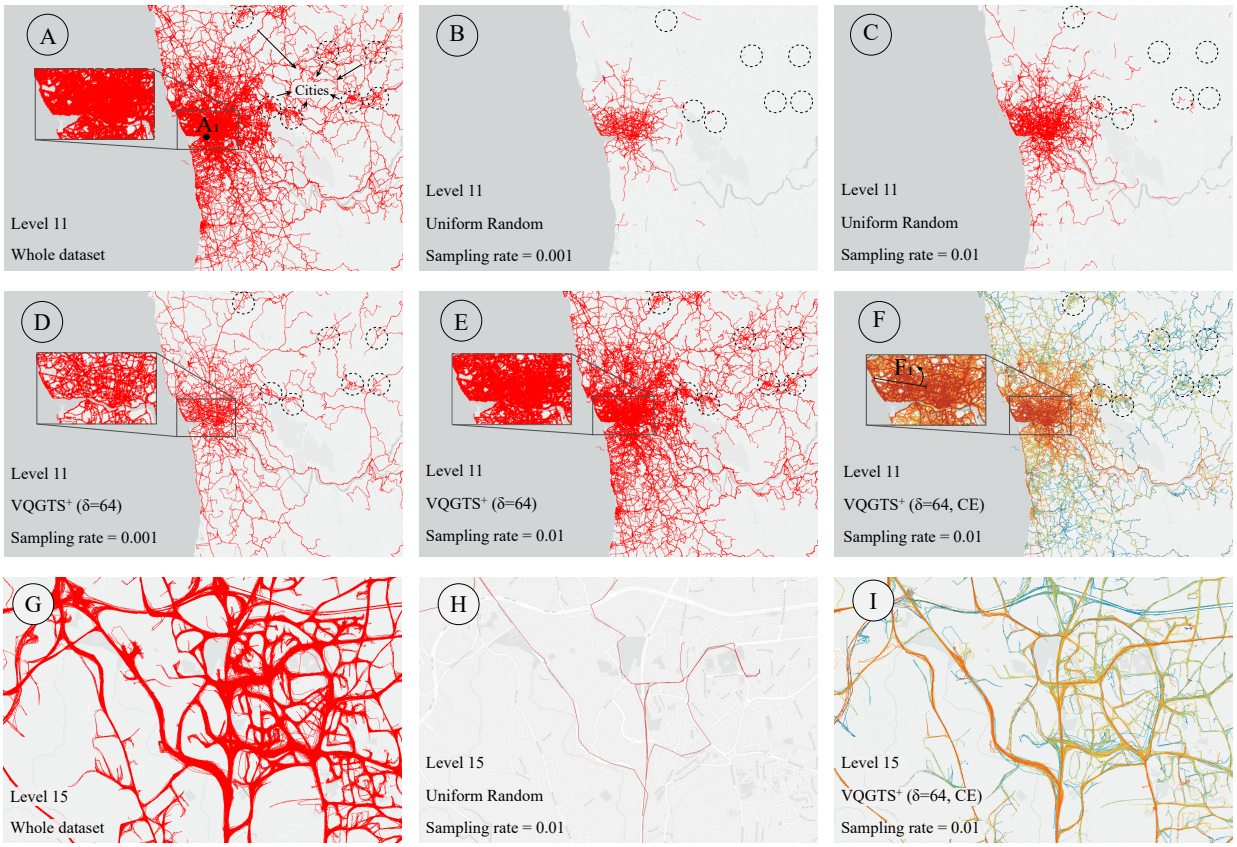


Fig. 1. Visualization results. (I) at zoom level 11, A is the visualization result of the whole Proto taxi trajectory dataset. B and C are visualization results of uniform random sampling approach with sampling rate 0.001 and 0.01, respectively. D, E, F are the visualization results of our proposal VQGTS$^+$ with different parameters. (II) G, H, and I are the visualization results of whole dataset, uniform sampling, and our proposal 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 challenge as (i) the large trajectory data size, (ii) the limited rendering ability of graphics device, and (iii) visual clutter issue in big data visualization. In this work, we propose a visual fidelity guaranteed sampling approach for the large-scale line-based trajectory visualization problem. 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-complete to select a sized-k subset trajectories with minimal visual fidelity loss. Next, we devise an approximate algorithm VQGTS with a suite of optimization techniques, which returns visual fidelity quality guaranteed visualization result efficiently. Moreover, we improve the visual effectiveness of VQGTS by taking human interpretation ability and visual clutter issue into consideration. 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., traffic flow detection, reachable route discovery.

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

1 Introduction

Nowadays, the widely used location-acquisition devices lead to an explosive increase of the movement data (i.e., GPS trajectories) from urban moving objects, e.g., cars, sharing bikes, and pedestrians. Trajectory visual analysis have been employed in many smart city applications, e.g., traffic management [27], urban planning, route recommendation [36] and location-based services [16, 35]. The most popular and conventional trajectory visualization method is the line-based visualization [8], i.e., connecting the GPS points of movement objects by polylines. However, efficient and effective large-scale line-based trajectory visualization is very challenge. The reasons are (i) large data size of trajectories, (ii) limited rendering ability of graphics device and (iii) visual clutter issue. We elaborate them as follows:

Large data size of trajectories: The trajectory data size is extremely

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

huge. For example, Shenzhen has 24,237 taxis and generates more than 82.8 millions GPS locations (e.g., taxi trajectories) in each day[1]. In New York City, it has 13,000 taxis, averagely, they carry over 1.0 million passengers and make 500,000 trips per day [10].

Limited rendering ability of graphics device: Rendering refers to the use of hardware device (e.g., Graphics Processing Unit) in the automatic generation of visualization result. However, due to the hardware limitation, the rendering ability of current Graphics Processing Unit is limited. We did a benchmark experiment to evaluate the rendering ability of NVIDIA GeForce GTX 1080 with 8GB video memory. We vary the number of trajectories from 1,000 to 1 millions, which are randomly selected from the Porto taxi trajectory dataset[2]. 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 32.67 millions GPS points (almost 40% of Shenzhen taxi trajectory GPS points in one day). Moreover, the render cost is linear with the input data trajectories. Thus, it is impractical to visualize billion-level GPS points via the commodity GPUs.

Visual clutter issue: Visual clutter is a common issue in big data visualization. Figure 1A is the visualization result of the whole 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 literature. Unfortunately, none of them could address these three challenges simultaneously and perfectly. In particular, the spatial aggregation based approaches [26, 33] preprocess the massive movement data, and visualize the results after preprocessing. These approaches alleviate the large data size and limited rendering ability issues in large-scale spatial data visualization. Nevertheless, they ignored the visual clutter issue in raw spatial data as they only visualize the aggregated/preprocessed results. In other words, the visualization result may lose many information in raw data. In recent years, many the visualization research proposed to address the visual clutter issue, e.g., edge bundling [25, 34] and density map [15, 24]. However, these works neither focus on line-based trajectory visualization nor designed for large-scale trajectory dataset.

Sampling techniques is a de-facto standard 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 [3] employs a reduction layer between visualization layer and data management layer. The reduction layer embedded an uniform random sampling algorithm to sample data randomly if 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 Figure 1B,C as an example, they are the visualization results of uniform random sampling method on Porto taxi trajectory dataset with sampling rate 0.001 and 0.01, respectively. Visually, both visualized figures cannot capture the overview of the input data, as illustrated in Figure 1A.

In database community, Park et al. devised a visualization-aware sampling algorithm (VAS) for large-scale scatter points visualization problem in [19], 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 [19] is only for scatter points, it cannot be used to measure the quality of trajectory visualization results.

In this work, we propose a visual fidelity guaranteed sampling approach (VQGTS) 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 proposal VQGTS 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 issue. 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-complete to select a sized-k subset of trajectories which has the minimal visual fidelity loss. Next, we devise an approximate algorithm (VQGTS) which returns a sized-k subset of trajectories and provides the visual fidelity guarantees of it. Last but not least, we settle the visual clutter issue explicitly and overcome it by taking the level of details and human interpretation ability into consideration in the improved algorithm (i.e., VQGTS$^+$). Figure 1(D,E) show the visualization result of our proposal VQGTS$^+$ on Porto taxi trajectory dataset with sampling rate 0.001 and 0.01, respectively. Obviously, the visualization fidelity of them (comparing with the visualization result of whole data set in Figure 1(A)) is much better than the uniform sampling visualization results with the same sampling rate (see B and C in Figure 1). Figure 1(F) is the returning result of our proposal which colors the trajectories with different popularity. It has the same input parameters of Figure 1(E). Intuitively, the visual clutter issue in Figure 1(E) is almost removed in Figure 1(F). In addition, our proposal scales the multi zoom resolution very well. Figure 1(G,H,I) depict the visualization result of the whole dataset, uniform random sampling and our VQGTS$^+$ at zoom-level 15, e.g., we can obtain them by zooming in the visualization result in Figure 1(A,C,F). Intuitively, the visualization result of our proposal (Figure 1(F)) outperforms the uniform random sampling method significantly (Figure 1(H)). It even performs better than Figure 1(G), the visualized result of the whole dataset, as it reduces visual clutter in G by encoding the popularity of different roads by different colors.

The contributions of this paper are:

- We formulate the visual fidelity guaranteed sampling problem for large trajectory data visualization, and prove it is NP-complete (see Section 3).
- We devise an approximation algorithm for it with a suite of optimization techniques, e.g., lazy updating, result heap maintaining (see Section 4).
- We propose an advanced approach to further enhance the effectiveness of our approximate algorithm, which address the visual clutter issue by introducing human interpretation ability parameter, and encodes the popularity of each road by different colors (see Section 5).
- We conduct extensive experiments on real-world trajectory dataset to demonstrate the effectiveness of our proposal, and conduct qualitative user studies to show their superiorities in various applications (see Section 6).

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 formulate 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.

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.

[1] http://jtys.sz.gov.cn/

[2] http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html

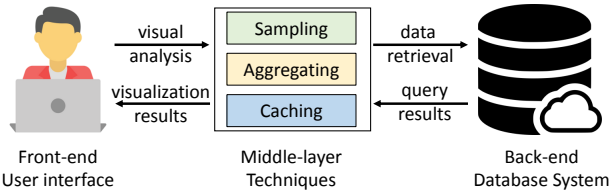


Fig. 2. Interactive visualization system architecture for large dataset

2.1 Trajectory visual analysis

Trajectory is the most common representation of the object movement. 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 [8], 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 [8] 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 [4, 5, 17, 30, 32]. 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, e.g., direction and path [8]. The region based visualization approaches divide the whole region into sub-regions in advance, then visualize the aggregated information in each sub-region [11, 26, 28]. 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 [12, 13]. 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, many clustering techniques are proposed for the visual analysis with various trajectory datasets, e.g., flight [9], taxi trips [23] and hurricane trajectories [2]. In addition, advanced interaction techniques [10, 14] and edge bundling techniques [34] are devised to detect the movement patterns. Unlike existing line-based visualization techniques, we propose a visual fidelity guaranteed sampling approach 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 problem.

2.2 Interactive visualization for large dataset

With the recent advance 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. Due to the limited rendering ability of modern commodity graphics device, generating visualizations for such scale of dataset always take considerable amount of time, or even impractical in practice [19]. In literature, many works are proposed to achieve interactive visualization in large dataset (not only for trajectory dataset), we briefly elaborate the most representative pieces in this subsection.

Figure 2 illustrates the architecture of interactive visualization systems (e.g., Spotfire [3], Tableau [4]) for large-scale datasets. 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 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 [5]). 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 [6]). In recent year, 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 middle-layer in subsequent.

Aggregating-based techniques: Aggregating-based optimization techniques [26, 33] in the middle-layer provide interactive visual analysis for large-scale data by reducing the number of rendering data. It achieves by preprocessing the raw data with aggregation techniques (e.g., clustering). Returning to the trajectory visual analysis, the aggregating techniques can be further distinguished by how to generate the spatial partitions. For example, OD Map [28] divides the whole map into nested uniform grid, and uses the color of a grid to present the flow magnitude. [11] employs the hierarchical administrative regions as basic units and visualize the flow by linking different units. Instead of using static predefined spatial partitions in [11,28], MobilityGraph [26] visualizes the aggregated tweet post locations in these regions which are returned by a spatial graph clustering algorithm dynamically. 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 is a de-facto standard solution for the interactive visualization problems with large-scale input data. It is widely studied in both visualization and database communities [3, 7, 19], In particular, [7] 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 [19], 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 [19] cannot be adapted to our large-scale trajectory visualization problem as (i) the complexity of the trajectories [21], 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 [20], and (ii) the trajectory clustering is still an open problem in both communities [1, 18]. Unlike the above research works, in this paper, we propose a visual fidelity guaranteed sampling approach (VQGTS) for the large-scale trajectory visualization problem, we demonstrate the superiority of our proposal by both case- and user- studies in real world datasets.

Caching-based and other techniques: Caching is commonly used to improve the performance of large-scale data processing system, e.g., search engine [31]. Chan et al. present ATLAS [6] 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. [22] 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 proposed techniques in this work are orthogonal to the researches in this category.

[3] https://www.tibco.com/products/tibco-spotfire
[4] https://www.tableau.com/
[5] https://d3js.org/
[6] https://spark.apache.org/

3 PROBLEM FORMULATION

Visualizing a large collection of trajectories are used frequently in map service or smart city applications. When exploring a large collection of trajectories, efficient and effective large-scale trajectory visualization is challenging in both academia and industry. The reasons are (i) the size of trajectory data is very large (e.g., several GB in an hour), and (ii) the limited rendering ability of existing commercial graphics device (e.g., XXX).

A naive solution to employ sampling idea for large-scale trajectory visualization problem is randomly selecting several trajectories from the data set then visualize it by graphics device. However, the visualization result may be not acceptable by the user. In this work, we study the large-scale trajectory visualization problem. In particular, we focus on visual quality guaranteed sampling method for large trajectory data visualization. The major challenges to design visual quality guaranteed sampling method are: (I) how to define visual quality theoretically? (II) how to guarantee the quality of the sampling-based visualization result? In this work, we first formate visual quality by defining the loss function between the visualization results of the whole dataset and sampled dataset. With the loss function, we analyze the hardness of the problem, and devise a visual quality guaranteed sampling algorithm for it.

3.1 Problem description

Problem 1 (Sampling-based trajectory visualization problem)**.** *Given a large-scale trajectory dataset* T *and an integer k, the trajectory visualization problem is selecting a subset of trajectories* $\mathsf{R} \subseteq \mathsf{T}$*, such that loss function loss*(R,T) *is minimized.*

From the user's perspective, there are many ways to define the loss function *loss* between the visualization result qualities of the sampled subset R and the whole dataset T. For example, [19] defined point-based loss function for very large scatter points visualization. However, it is not applicable for trajectory data visualization. In order to address that, we propose an novel loss function for trajectory visualization problem.

Intuitively, the visual quality difference between the visualization results of two trajectory datasets 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 quality 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 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 number of rendered pixels in the canvas of a given trajectory dataset.

Given a trajectory data set T and an integer k, our research objective is finding subset R, such that the visualization quality loss function $loss(\mathsf{T},\mathsf{R})$ is minimized, i.e.,

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

3.2 Hardness analysis

In real-world applications, the pixels in canvas will be rendered by different colors according to the specified visualization scheme. For the sake of presentation, we analyze the hardness of our research objective with a simple render manner of visualization result. In particular, for each pixel in the canvas, 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 $\mathscr{U}$ be the universal set of all pixels in the canvas For each trajectory $\mathsf{T}_i \in \mathsf{T}$, it consists of a set of pixels in the canvas, e.g., it is a subset of $\mathscr{U}$. Thus, the subset R also is a subset of $\mathscr{U}$ as $\mathsf{R} = \cup_{\mathsf{R}_i\in\mathsf{R}}\mathsf{R}_i$.

Our research objective is minimizing loss function $loss(\mathsf{T},\mathsf{R}) = \frac{\mathsf{V}(\mathsf{T})-\mathsf{V}(\mathsf{R})}{\mathsf{V}(\mathsf{T})}$. Obviously, the visualization result of T is a constant value, denotes as C. Our research objective of Problem 1 can be transformed as follows:

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

It is equivalent to select sized-k trajectory set R from T which $\cup_{\mathsf{R}_i\in\mathsf{R}}\mathsf{R}_i$ is maximized.

It is a typical set cover maximization problem[7], which is a well-known NP-hard problem in literature.

4 VISUAL QUALITY GUARANTEED SAMPLING APPROACH

Due to the hardness of the Problem 1, we first introduce a straightforward solution for it in Section 4.1. Then, we propose a visual quality 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. As the pseudocode in Algorithm 1 shown, it randomly selecting k trajectories from T, then render these selected k trajectories into the canvas.

Algorithm 1 RandSampling(T,k)

```
1: Initialize result set R ← ∅
2: while |R| < k do
3:     R_tmp ← RAND(T − R)
4:     R ← 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 quality of the visualization result.

4.2 Visual Quality Guaranteed Sampling Algorithm

In this section, we present our visual quality guaranteed sampling algorithm for Problem 1. We start our presentation by elaborating the relationship between visual quality of sampled set R and user zoom level. Obviously, for a given sampled set $\mathsf{R} \subseteq \mathsf{T}$, it has different loss values at different user zoom level. The reason is the map are of the same canvas region will be updated at different zoom level. For example, Google map[8] 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. For each trajectory $\mathsf{T}_i in \mathsf{T}$ is a set of pixels at highest zoom level in the canvas.

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

Interestingly, Algorithm 2 has a nice theoretical property, i.e., it guarantees the visual quality of the returning set 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 trajectory visualization problem (i.e., Problem 1).*

[7] https://en.wikipedia.org/wiki/Maximum_coverage_problem
[8] https://www.google.com/maps/preview

Algorithm 2 VQGTS(T,k)

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

Proof. The optimal solution of Problem 1 covers OPT elements in k iterations. Let a_i be the number of newly covered elements at the i-th iteration, b_i is the total number of covered elements up to the i-th iteration (i.e., $b_i = \sum_{j=1}^{i} a_i$), and c_i be the uncovered elements after i-th iteration (i.e., $c_i = OPT - b_i$). According to greedy paradigm, we can conclude the number of newly covered elements at the $(i+1)$-th iteration is always greater than or equal to $\frac{1}{k}$ of the number of uncovered elements 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 provides a visual quality guaranteed sampling algorithm for large trajectory data visualization problem. However, it is impractical for (very) large trajectory dataset (i.e., billions of trajectories) as the time complexity analyzed in the following Lemma 1.

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

Proof. The complexity analysis is straight forward as all trajectories (i.e., $|\mathsf{T}|$) will be updated (i.e., $O(m)$ for each one) and the trajectory with maximum number of uncovered pixels will be selected at each iteration (k iterations in total). Thus, it is $O(k \cdot m \cdot |\mathsf{T}|)$. □

Motivated by this, we devise an lazy updating manner to accelerate the running time performance of our proposed visual quality guaranteed sampling algorithm. The core idea is the submodularity of the covered pixels of result set R as shown in Lemma 2.

Lemma 2 (Submodularity)**.** *Given a trajectory* T_i *and two result sets* S, S'*, where* $\mathsf{S} \subset \mathsf{S}'$ *and* $\mathsf{T}_i \notin \mathsf{S}'$*, it holds* $|\mathsf{S} \cup \mathsf{T}_i| - |\mathsf{S}| \geq |\mathsf{S}' \cup \mathsf{T}_i| - |\mathsf{S}'|$*.*

Proof. Let the contribution value of trajectory T_i to a given result set S as $|\mathsf{S} \cup \mathsf{T}_i| - |\mathsf{S}|$. It is the new covered pixels of $|\mathsf{T}_i|$, i.e., $|\mathsf{T}_i| - |\mathsf{T}_i \cap \mathsf{S}|$. It holds $\mathsf{T}_i \cap \mathsf{S} \subseteq \mathsf{T}_i \cap \mathsf{S}'$ as S' is a superset of S. Thus, we have $|\mathsf{T}_i| - |\mathsf{T}_i \cap \mathsf{S}| \geq |\mathsf{T}_i| - |\mathsf{T}_i \cap \mathsf{S}'|$, it is equivalent to $|\mathsf{S} \cup \mathsf{T}_i| - |\mathsf{S}| \geq |\mathsf{S}' \cup \mathsf{T}_i| - |\mathsf{S}'|$. □

With the help of submodularity property, it reduces a lot of unnecessary trajectory updating computations. In particular, we maintains a max-heap for the number of uncovered pixels of each trajectory w.r.t., result set R. Figure 4(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 Figure 4(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 $\mathsf{R} = \{\mathsf{T}_3\}$ (see gray node at Figure 4(b)). T_1 will be selected at the 2nd iteration without updating the number of uncovered pixels in other trajectories, i.e., all white nodes at Figure 4(b) as the submodularity property holds.

In summary, in the lazy updating manner, the number of uncovered pixels in each trajectory will only be computed with the latest result set R when it is necessary, e.g., only T_1 will be updated at the 2nd iteration in Figure 4. It reduces many unnecessary computations through the lazy updating manner, e.g., all white nodes did not update at the 2nd iteration in the above example. We analyze the time complexity of Algorithm 2 with lazy updating manner in Theorem 3.

Lemma 3 (Optimized Time Complexity)**.** *Given trajectory dataset* T *and an integer k, the time complexity of Algorithm 2 with lazy updating manner is* $O(|\mathsf{T}| + k \cdot m \cdot t \log |\mathsf{T}|)$*, where t is the maximum number of updating among all k iterations and* $t \ll |\mathsf{T}|$*.*

Proof. It first takes $O(|\mathsf{T}|)$ time to construct the max-heap. Then, it incurs $O(m \cdot t \log |\mathsf{T}|)$ cost to select the trajectory with maximum uncovered pixels at each iteration (k iterations in total). Hence, the overall cost is $O(|\mathsf{T}| + k \cdot m \cdot t \log |\mathsf{T}|)$. □

Bo: To exemplify, Algorithm 2 costs 2.3 hours to return the results on Lisbon taxi trajectory dataset. However it only takes 3.2 seconds with lazy updating manner.

5 Advance Approach: VQGTS$^+$

Even though our proposed visual quality guaranteed sampling VQGTS produces good quality visualization result for large trajectory dataset when comparing with random sampling method. In this section, we devise an advanced VQGTS (i.e., VQGTS$^+$) to further enhance the visual quality of VQGTS by exploiting (i) the inherent characteristic in the large trajectory dataset, and (ii) the interpretation ability of human beings.

We then elaborate (i) and (ii) by the examples in Figure 3. Considering Lisbon trajectory dataset, Figure 3(a) shows its visualization result of VQGTS with sampling ratio 1%. The distribution of real-world trajectory dataset is non-uniform inherently. For example, the dense region (Lisbon downtown) has much more trajectories than the sparse region in Figure 3(a). Obviously, it is much easier for us to distinguish the difference between sparse regions rather than dense regions in Figure 3(a) and (b). The major reason is that end users will treat the two dense regions in Figure 3(a) and (b) as identical since the interpretation ability of human beings is limited at such level of details. However, the visual difference between the two sparse regions in Figure 3(a) and (b) at that level of details is in the range of ours interpretation ability.

Hence, the returning result of visual quality guaranteed sampling method VQGTS could be further improved by delivering richer information at sparse regions, i.e., enhancing the visual details in the region where is in the range of end user's interpretation ability. Motivated by this, we devise an advance approach VQGTS$^+$ (see Algorithm 3) by incorporating a parameter δ during trajectory selection process in VQGTS to achieve the above goal. In particular, we employ the parameter δ to model the end user's interpretation ability at the most high level of details. Surprisingly, our advance approach VQGTS$^+$ not only provides better visualization result when comparing with VQGTS with the same sampling rate (e.g., Figure 3(a) and (b) are the returning result of VQGTS and VQGTS$^+$ respectively), but also embeds the popularity of selected trajectories by encoding the rest trajectories in the dataset in them, e.g., Figure 3(c) is the visual result of VQGTS$^+$ with color encoded popularity.

Instead of measuring the contribution of each trajectory w.r.t the selected trajectories in R directly, we introduce interpretation tolerance parameter δ to capture end user's interpretation ability at the highest zoom level. Specifically, suppose the pixel with location (x,y) in canvas is covered by result set R at the highest level, the pixels around (x,y) are not necessary to cover as they are beyond the interpretation ability of end users. It means the end user cannot distinguish the

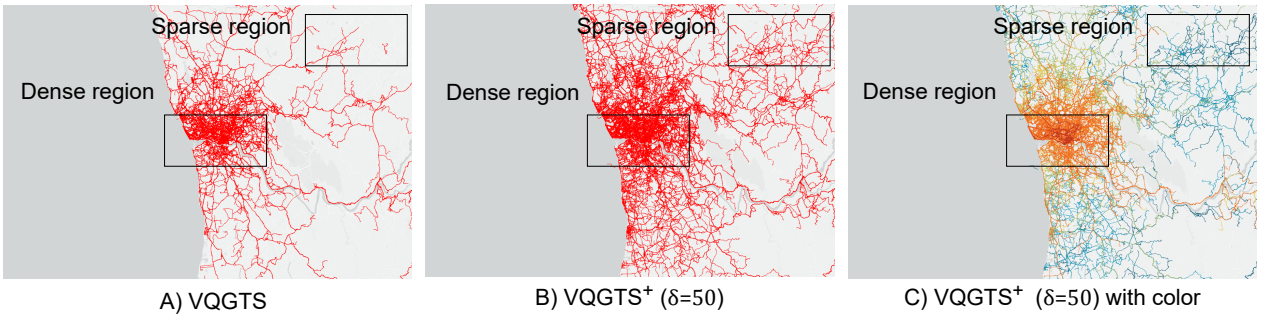
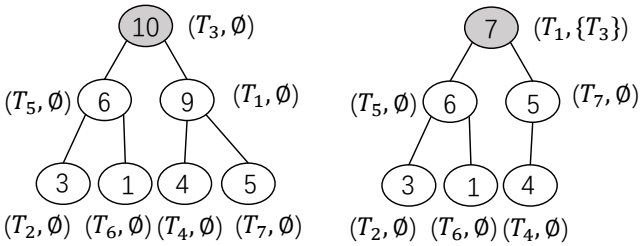


Fig. 3. Detail-aware visual quality guaranteed trajectory sampling, A: VQGTS, B: advanced VQGTS (VQGTS$^+$), C: VQGTS$^+$ and popularity encoded by color.



(a) 1st iteration (b) 2nd iteration

Fig. 4. Lazy updating manner illustration

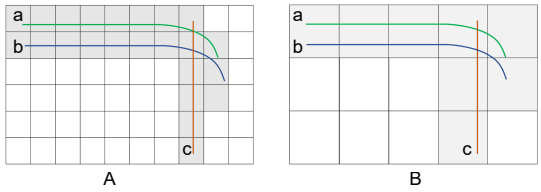


Fig. 5. Zoom resolution2

visual quality difference between rendering (x,y) and rendering all pixels in from $(x-\delta,y-\delta)$ to $(x+\delta,y+\delta)$) even at the highest zoom resolution. Fortunately, we can incorporating the interpretation tolerance parameter δ into our proposed visual quality guaranteed trajectory sampling algorithm in Algorithm 2 by slightly revising it. As illustrated in Algorithm 3, it measures the contribution of each trajectory 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 R_tmp and its tolerance pixels set (in Line 6).

Algorithm 3 VQGTS$^+(\mathsf{T},k,\delta)$

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

Moreover, we further exploit the result set of interpretation tolerance considered VQGTS to address the visual clutter problem in large trajectory visualization application. In particular, it selects the trajectory which has largest uncovered pixels by taking end user's interpretation tolerance on visual quality into account at each iteration, instead of only choose the trajectory with largest uncovered pixels in VQGTS. During the above selection process, some trajectories will not be included into the result set R even they have larger number of uncovered pixels. The reason is their uncovered pixels are too close to the pixels in the selected trajectory, i.e., within the tolerance area of selected pixels. Taken Figure 5(a) as example with $\delta=1$, suppose trajectory a was selected at the first iteration, the trajectory selected in the second iteration is c, instead of b as almost all pixels in b is in the tolerance area of a's. It means it embeds the popularity of each selected trajectory during the result set R selection process naturally. We define the popularity of each selected trajectory as the number of close trajectories in the rest dataset, i.e., $\mathsf{T}-\mathsf{R}$. Thus, we compute the popularity of trajectories in R from Line 7 to Line 9 in Algorithm 3. We then visualize the popularity of each selected trajectory by encoding its popularity by different colors. Figure 3(c) shows the returning result of advance approach VQGTS$^+$ by encoding the popularities by colors. Obviously, the trajectories in dense region are more popular than these in sparse regions.

Last but not least, it is worth to point out that our advance approach VQGTS$^+$ provides more better visual quality over VQGTS at arbitrary zooming resolutions. The key technique to achieve that is it considers the zooming resolutions inherently when introducing the interpretation tolerance δ. Take Figure 5 as an example, Figure 5(a) and (b) show two different zoom-levels. The zoom level in Figure 5(a) is higher than it in Figure 5(b). As our above elaboration, our advance approach VQGTS$^+$ selects trajectory a and c at Figure 5(a). When it zoomed-out, as shown in Figure 5(b), it still capture the main sketch of the underlying dataset.

6 Evaluation

We first apply our approaches to several real-world dataset and compare our method with uniform random sampling. Then we conduct several user studies on specific analysis tasks.

6.1 Case Study

In this section, we evaluate our method by presenting the applications on two taxi trajectory datasets: Porto and Shenzhen. We compare the visualizations generated by the full dataset, random sampling and the proposed method from multiple levels of details.

6.1.1 Case of Porto Distinct

Our first example uses taxi trajectories collected from 442 active taxis in Porto distinct, Portugal [9]. The trajectories cover several cities in and around the Porto distinct and has been cleaned for further analysis.

Overview of the Porto Distinct: Figure 1(A) presents the visualization generated by the whole dataset, from which we can get an overview of the movement patterns in Porto district. For example, large number of trajectories are concentrated in the center of the figure(shown around

[9] http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html

Figure 1(A_1)), indicating that the most taxi activities are around the Porto City. In addition, many trajectory clusters distributed across the land, which indicates the locations of other cities in Porto district(shown as the dashed circle in Figure 1(A)).

We compare the VQGTS$^+$ with *uniform random sampling* at the overview. Figure 1(C,E) show the visualization generated by uniform random sampling and VQGTS$^+$ respectively. Both of these two sampling methods take 0.01 as the sampling rate. The uniform random sampling almost only preserves the visual structure around the Porto City and the trajectories at the marginal regions are lost. The visualization generated by VQGTS$^+$ looks close to the whole dataset. Not only the Porto City but also the marginal city structures are well preserved, which are shown as the dash circles in Figure 1(E).

Furthermore, the color encoding further enhances the visualization by revealing the representativeness of the trajectories. As shown by the rectangle region in Figure 1(E), there is no clear pattern that can be discovered due to the dense concentration of massive trajectories. While in the same region of Figure 1(F), some trajectories with high representative scores are highlighted by dark orange color such as F_1, which indicates Avenida da Boavista, a main avenue in Porto City.

An important factor affecting the visual fidelity of sampling results is the sampling rate. Figure 1(C,B) show the sampling results of random method with sampling rate set as 0.01 and 0.001. With the decreasing of the sampling rate, the shape of the trajectory visualizations clearly shrink to the Porto City and result to the loss of visual fidelity at the marginal region. Figure 1(E,D) demonstrate the visualization of VQGTS$^+$ with the sampling rate of 0.01 and 0.001. We observe that when the sampling rate is decreased, the overview framework of the trajectories remains the same but the trajectories around Porto City are significantly removed since more blank space/gap can be found around the Porto City as shown in the rectangle of Figure 1(D).

Detail view of Porto Distinct: To demonstrate the effectiveness of VQGTS$^+$ at detail view, we select three regions of interest(as shown in the Figure6(A)), which have different trajectory density and generate the visualization by setting the map level of 14)

Region R3 is far away from Porto City and contains two other cities: Paredes and Penafiel. Region R3 has very few trajectories as shown in Figure 6(D_1). Compare with the visualization of full dataset, the random sampling almost misses all trajectories in this region(as shown in Figure 6(D_2)). While VQGTS samples much more trajectories than random sampling as shown in Figure 6(D_3). However, some meaningful structure are still missing such as the trajectory bundle shown in Figure 6(h), which is laid in on the road road name connecting the two cities of Paredes and Penafiel. Further more, the trajectory structure of city Penafiel is not precisely preserved(shown as region g in Figure 6(h)) because some trajectory branches are missed. By setting the interpretation tolerance parameter as 64, VQGTS$^+$ generate a more confidential visualization than VQGTS shown as Figure 6(D_4).

Region R2 is near to the center of Porto have are more taxi trajectories. There are three cities located in the region R2 including Ermesinde, Rio Tinto and Valongo. Figure 6(C_2,C_3) present the visualization generated by VQGTS$^+$ with the interpretation tolerance parameters of 4 and 64. We observe that the visualization shown in Figure 6(C_3) have more details trajectory branches than Figure 6(C_3)(as shown in regions c,d of Figure 6(C_2,C_3)). In this case, a larger interpretation tolerance parameter is more beneficial in preserving the details at this level. Furthermore, Figure 6(C_4) shows the visualization of VQGTS$^+$($delta = 64$) with color encoding. In the comparison with Figure 6(C_3), the visualization in Figure 6 further highlights the movement distribution. For instance, we observe that region f in Figure 6(C_4) has more deep colored trajectories than region e and g, thus we inspect that city of Rio Tinto has more taxi trajectories than other two cities.

The region R1 is the center of Porto city, which has the highest concentration of the trajectories and cause serious visual clutter if all trajectories are visualized(as shown in Figure6(B_1)). VQGTS$^+$($delta = 4$) greatly alleviates the visual clutter and preserves the framework which basically follows the road network as shown in Figure 6(B_2). Furthermore, when setting the interpretation tolerance parameter *delta* as 64, the structure is more clear as shown in region b of Figure 6(B_3,B_4), and have more trajectory details than VQGTS$^+$($delta = 4$) as shown in region a of Figure 6(B_2,B_3). The trajectories with color encoding further enhance the visualization thus the audiences can compare the traffic flow of two route more easily.

6.1.2 Shenzhen Trajectories

We further evaluate the proposed approach using the taxi trajectories of Shenzhen, a booming city in the southern China and has very different urban form from Porto District. The dataset we used includes 428K taxi trajectories collected from ** taxis in one day. All the visualization generated by the sampling methods which sets the sampling rates as 0.01.

Overview of Shenzhen: Figure 8(A-D) present the overview visualization generated by whole dateset, random sampling, VQGTS$^+$ and VQGTS$^+$ with color encoding at the top level. The visualization of raw dataset(Figure 8(A)) shows the there are several dense trajectory clusters in southern districts of Shenzhen, including *Baoan*, *Nanshan*, *Futian* and *Luohu* districts, which are the most prosperous commercial zones of Shenzhen. Figure 8(B) presents the trajectories generated by random sampling. We observe that the most of the trajectories sampled by random sampling are located at the commercial zones. On the other hand, the trajectories at the north Shenzhen are missing, thus making the visualization visually different from the whole dataset. VQGTS$^+$ outperforms random method by guaranteeing the spatial coverage of the whole trajectories thus result in a higher visual fidelity. Further more, some isolate trajectories are still preserved shown in Figure 8(C). VQGTS$^+$ with color is able to reveal the spatial distribution of the trajectories. For example, for the regions a, b in Figure 8(A or C), the visualization is unable to explain which region has more taxi activities because both of these two regions are fully covered by trajectories. In Figure 8(D), we find the more trajectories encoded by deep color are found in region a than that of region b, which indicates more trajectories can be found in region a than region b.

Detail view of Shenzhen: Then we narrow down to the region of airport. Compare with visualization of whole dataset(shown as Figure 8(E)), the random sampling only preserves the trajectories pass through several routes with very high traffic flow(shown as Figure 8(F)). Both VQGTS$^+$ and VQGTS$^+$ with color encoding can visualize the trajectory structure very well. The VQGTS$^+$ with color encoding further enriches the information by encoding the trajectory with color. For example, we can observe that there are more trajectories passing through G4 and G104 than Baoan Avenue, which is hard to be discovered from the Figure 8(E,F,G).

Similarly, in the region near to the Shenzhen North Railway Station, the visualization generated by VQGTS$^+$ can reveal some road structure such as the round entrance to the motorway shown as region c in Figure 8(K). With the color encoding, we can also easily discover that the road G94 have a higher road traffic flow than the Minzhi avenue and Mellon avenue as shown in Figure 8(L).

6.2 User Study

To further evaluate the effectiveness of VQGTS$^+$ from the the audience perspective, we conducted formal user studies to compare how users perform the urban exploration tasks with visualizations generated by whole dataset, *random sampling* and VQGTS$^+$.

6.2.1 Experiment setting

Participants and apparatus: We recruit 100 participants (** females, ** males, aged 20 to **(mean=**, SD=**)) with normal vision or normal corrected vision. All of these participants have the background of computer science. The study system is a web-based platform which has the size-fixed interface and the participants perform the user study on their own computers. The system interface is shown as Figure 9. Considering the unfairness caused by the screen size, we recommend all participants to set the resolution of the screen as 1980 * 1080 before the experiment. All images displayed on the interface have the same

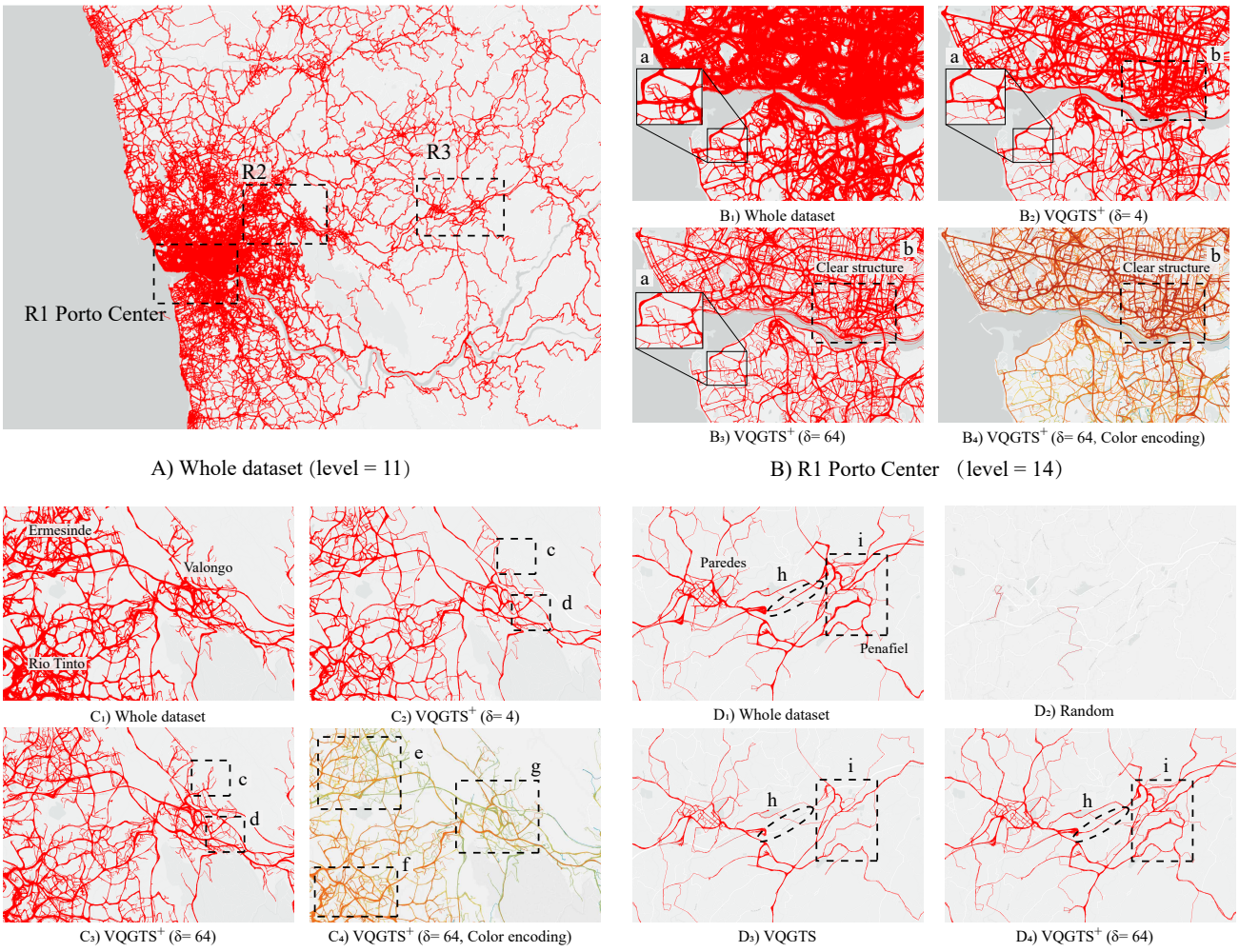


Fig. 6. Visualization at dense and sparse region respectively.

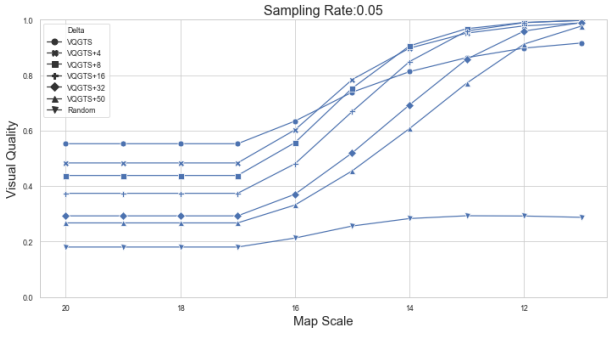


Fig. 7. Visual quality chart. X axis indicates map scale from detail view to overview; y axis indicate the visual quality.

size of(AA*BB).

Tasks: All participants needed to perform three types of tasks(shown as Figure 9(B,C,D)):

- **T1. Traffic flow estimation.** The participants are asked to select the road segment with higher traffic flow from two candidates. A road with larger traffic flow has more trajectories in the dataset passing through the road segment, thus lead to 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 warm color.
- **T2. City/commercial region center identification.** The participants need to identify the centers of the cities or commercial regions through the trajectory visualization. The city or commercial region centers always have more passing trajectories from different directions than the surrounding regions which results in the start-shape cluster of trajectories in the visualization.
- **T3. Reachable route inspection.** The reachable routes indicate the routes connecting two regions, these routes must have the passing trajectories. The participants were asked to inspect the reachable routes between the two regions by drawing lines.

Data generation: We used the taxi trajectory dataset of Porto and Shenzhen for the user study. The testing data for each type of tasks were generated as follows:

Tasks of T1. We selected several visualization views which contain clear road structures. Then the number of trajectories passing through each road segment was counted as the traffic flow of this road segment. In the tasks of T1, two road segments in a view are given randomly, and the users needed to select the one with higher traffic flow.

Tasks of T2. We randomly chose several visualization views which contain city/commercial regions and labeled the locations of each

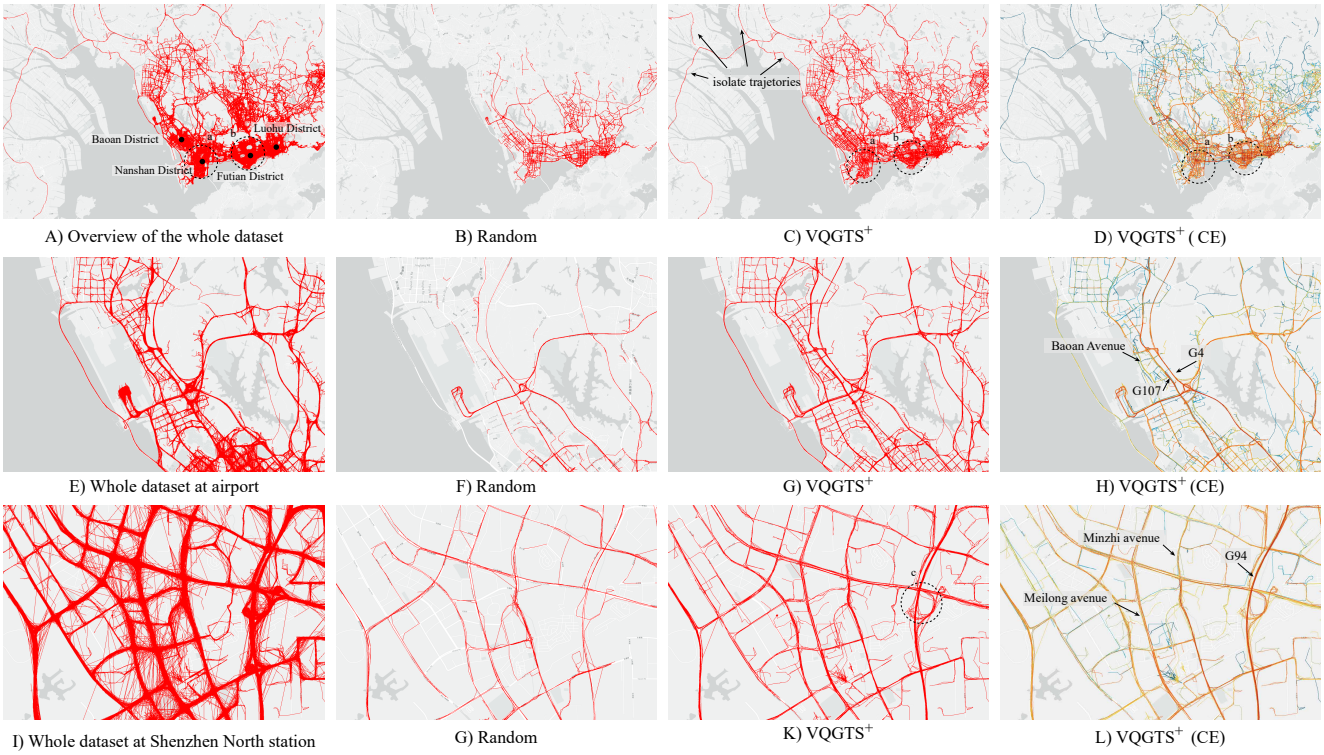


Fig. 8. Case study of Shenzhen.

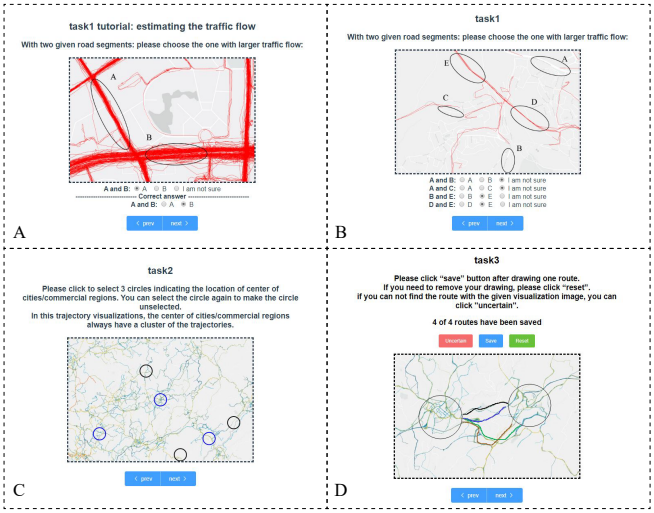


Fig. 9. The interface of user study platform.

city/commercial region center on the visualization as the correct locations first. Then we randomly generated locations and remove the locations close to the correct locations, the remaining locations are the error locations. In each task of T2, with a given visualization, the same number of correct and wrong locations will be randomly selected and the participants were asked to select the locations indicating the city/commercial region centers.

Tasks of T3. We randomly chose several visualization views which contain two or more city/commercial regions, which will be marked by circles. In each task of T3, an visualization view and two regions will be selected randomly.

Procedure: The user study began with the introduction which introduces the motivation, tasks and visual encoding. Then the following sessions are divided into three blocks according to the task types. Each block starts with a task tutorial, in which the participants could perform several demo tasks, thus familiarizing themselves with the interface, interaction and tasks. For example, Figure 9(A) shows the demo task of T1, in which the users can check the correct answer after clicking the "check" button. In each task of T1, with a given visualization, multiple road segment will be identified by ellipses(shown as 9(B)). Several road segment pairs are listed under the visualizations. The participants were asked to choose the one with larger traffic flow by clicking the radio box. They can also choose "I am not sure" if they cannot decided the answer. In each task of T2, a visualization view is given and several regions are marked by circles(as shown by Figure 9(C)). The users needed select the regions which could be city/commercial centers by click the corresponding circles. In each task, the number of correct regions are given. In each task of T3, with a visualization view and two circles(shown as Figure 9(D)), the users needed to draw several most representative reachable routes to connect the two circular regions. The number of the reachable routes is given.

6.2.2 Results

Accuracy:

Time usage:

User feedback:

7 Conclusion and Future Work

Visualizing large trajectory dataset is challenge due to two reasons: visual clutter and long rendering time. Data sampling technique, an effective method in reducing the rendering time by shrinking the data size, has been applied in a variety of data. However, very few work target at the trajectory sampling especially from the perspective of visualization. The most commonly used sampling method, uniform random sampling technique, always generate results with very poor visual quality because very few trajectories located at margin regions can be preserved. We fill the gap by proposing a novel sampling techniques VQGTS$^+$ which guarantees the visual quality at overview and reduce

the visual clutter at the detail view. The technique characteristics and a series of parameters setting are discussed. We compare VQGTS$^+$ with uniform random sampling in regarding to visual quality preservation and time-usage. We evaluate the effectiveness of proposed method by applying our method to different dataset and conducting users studies on specific interactive trajectory exploration tasks.

Even though it is recommended to use our method with caching techniques, our experience in the experiment shows that a faster algorithm will be more user friendly for the real world ad-hoc exploration tasks. For future work, we first plan to reduce the time usage by leveraging the advanced database techniques such as the indexing technique or use GPU acceleration. In addition, there are several directions can be further explored to enrich the information presented by the visualization. First we will develop different color encoding schema to present the spatial distribution of trajectory more precisely. In current schema, the color of one trajectory is the same, thus the color of the long trajectories may mislead the users because they pass through many regions with different level of the traffic crowdedness. One solution is to use gradient color schema to encode the trajectories. Another interesting direction is to extend the approach to support the mulit-class characteristics which is a commonly existed in variety of trajectory dataset.

REFERENCES

[1] 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.

[2] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia. Clustering trajectories by relevant parts for air traffic analysis. *IEEE transactions on visualization and computer graphics*, 24(1):34–44, 2017.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] A. Woodruff, J. Landay, and M. Stonebraker. Constant density visualizations of non-uniform distributions of data. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 19–28, 1998.

[30] 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.

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

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.