Visual Fidelity Guaranteed Sampling for Large Trajectory Data Visualization

Category: Research

Paper Type: please specify

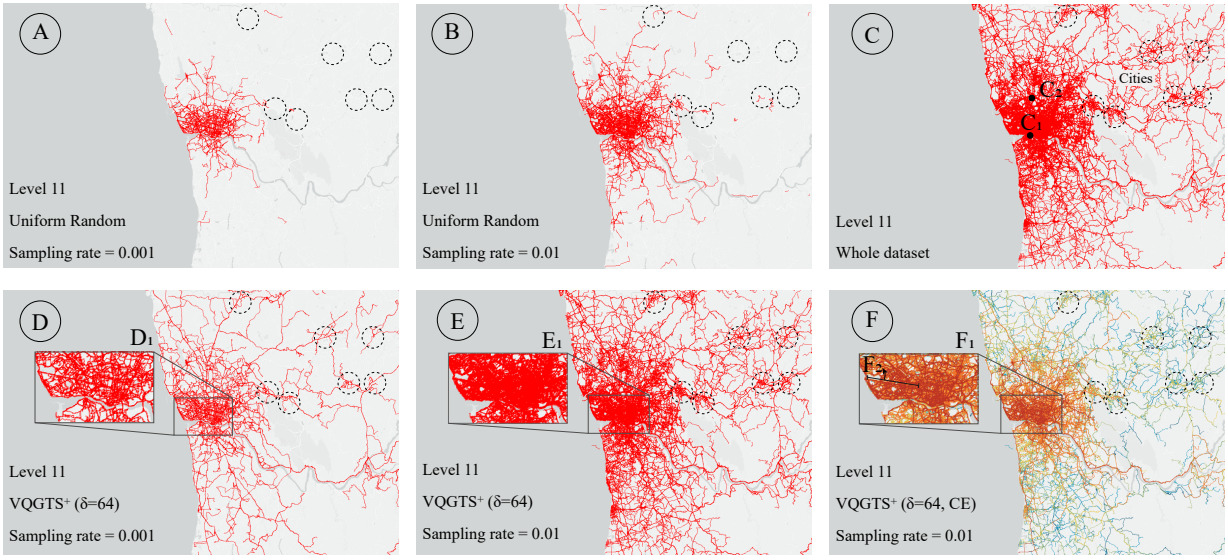


Fig. 1. Visualization results. (I) at zoom level 11, C is the visualization result of the whole Proto taxi trajectory dataset. A and B are visualization results of uniform random sampling approach with sampling rate 0.001 and 0.01, respectively. D, E, F are the visualization results of our proposal VQGTS$^+$ 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 line-based trajectories is the core subroutine in many smart city applications, e.g., traffic management, route recommendation. However, it is challenge as (i) the large trajectory data size and (ii) the limited rendering ability of graphics device. 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 of 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 level of details and visual clutter issues into consideration. We conduct extensive experimental studies to demonstrate the effectiveness of our proposal over state-of-the-art sampling methods on real-world trajectory datasets. In addition, qualitative user studies further demonstrate the superiority of our approach in various applications, e.g., traffic flow detection, city commercial region center identification.

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 which is recorded in the form of trajectories. For example, the taxis trajectory is one of the common studied movement data which is always considered as the representative of human movement trace in a city. Using the taxi dataset in Shenzhen as an example, more than 1 million trajectory data can be collected every day, which are recorded by the sampling locations. The analysis over these databases can be applied in many fields such as traffic management [27], urban planning, route recommendation [35] and location-based services [15, 34].

Visualizing a large collection of trajectories are used frequently in map service or smart city applications. The most popular and conventional method is the line-based visualization [7]: connecting the passing points of movement objects by polylines. To handle the big dataset, many visualization products such as Spotfire [1] and Tableau [2] support advanced database management systems as a "backend" for the efficient data processing the query. The current visualization tools always don't scale well for the presentation of very large trajectory dataset due to the two challenges, visual clutter and limited rendering speed, which hinders the abilities of human-users for interactively exploring the dataset and identifying the movement patterns. In recent years, most of the visualization research works mainly try to address the visual clutter issue by proposing new techniques such as the spatial aggregation [26, 32], edge bundling [25, 33] and density map [14, 23]. Instead, in this paper, we focus on the challenge of inefficient rendering in the large trajectory dataset by involving data sampling techniques.

It is time consuming to generate very simple visualization when the

[1] https://www.tibco.com/products/tibco-spotfire
[2] https://www.tableau.com/

data size become very large. Using Porto taxi data [3] as an example, Table 1 demonstrates the rendering time at each dataset size. shall also mention which rendering toolkit is used here. It shows that normal method takes more than 14 minutes (seconds?) to generate the graphics for 1 million trajectories, which is far beyond the human-acceptable response time for the interactive exploration [24]. One work closely related to ours is ScalaR [2], which adds a reduction layer between visualization layer and data management layer. The reduction layer uses an uniform random sampling method to sample data once the query results are large enough, thus to reduce the amount of data to be visualized. Further more, Park et al. propose VAS [18] which implements new sampling techniques to guarantee the visual quality. However, these sampling techniques are designed for the simple dataset, and have been approved effective in scatter plot or map plot. However, the trajectory sampling is more challenge due to the complexity of data form(e.g. varying lengths, lack of compact representation, difficulty in measuring the similarity) that makes traditional density-biased sampling techniques inappropriate. 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 because of the visual information loss in the sparse distributed regions.

Table 1. The time used to generate the visualization with different trajectory amount. shall add a column of number of line segments or number of nodes.

Data size	Time (ms)
100	2
1,000	16
10,000	143
100,000	1,416
1,000,000	13,950

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 study how to reduce the rendering time and preserve the visual quality for the large-scale trajectory visualization. We extend the motivation of visualization-aware sampling to trajectory dataset and propose a novel sampling strategy, **v**isualization **a**ware **t**rajectory **s**ampling(VATS), that produces high-visual-quality line-based trajectory visualization at different zooming resolutions. We first format 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. Figure 2 depicts an comparison among the ground truth, uniform random sampling and our proposed method. With the same sampling set size(1%), the proposed method generates a higher-fidelity visualization and support the multi-resolution very well. At last, color encoding are applied to enhance the distribution of trajectories.

We summarize our contribution as follows:

- We formulate VATS as an optimization problem.
- We prove VAST problem is NP-hard and offer an efficient approximation algorithms.
- We conduct several experiments using real-world data to demonstrate the effectiveness of the proposed method in comparison with random uniform sampling.

The remaining parts are constructed as follows: section 2 discusses the related work. In section 3, we identify the specific problem and provide an overview of our solution. We define the problem and propose the solution in the section 4 and 5. The implementation and experiment setting are introduced in section 6. In section 7, we conduct case studies and user studies to evaluate our approach. Finally, we conclude this paper and propose the possible future directions in section 8.

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

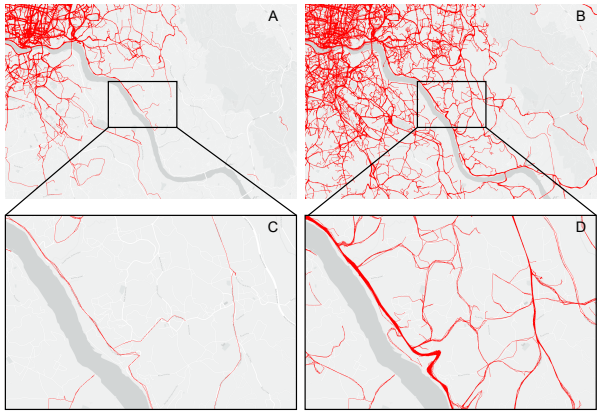


Fig. 2. Trajectory sampling generated by uniform random sampling(A,C) and VQGTS(B,D) at same sampling rate. In both high-level(A,B) and low level(C,D) view, our approach preserved more detail information about the trajectories especially for the sparse regions.

2 Related Work

The most related techniques to our work include the visual analysis of trajectory dataset, the methodology of large data visualization and data sampling.

2.1 Trajectory analysis

Trajectory, consisting of a sequence of spatial locations, is the most common form of the object movement. To support the understanding and analysis of the trajectory dataset, many visualization and visual analytics system are developed. The detailed summary of these work is presented in [7]. These techniques can be classified into three categories according to visualization form: point-based visualization, line-based visualization and region-based visualization.

The point-based visualization capture the basic spatial distribution of the passing points of the moving object. Furthermore, many density-based methods such as the kernel density estimation(KDE) are applied based on the point-based visualization [4, 16, 31], by the sacrifice of the detail the information of trajectories, these methods alleviate the visual clutter caused by large amount of data. Furthermore, to be better applied in the city environment, advanced KDE techniques are developed to capture the moving patterns along the road networks [3, 30]. In the study of urban traffic, the point-based visualization can capture the hot regions, but unable to identify the movement of the individual case and reveal the moving information such as the direction and origin-destination [7]. Line-based techniques are the most commonly used visualization methods which present the trace of the movement as polylines, thus to preserve the continuous moving information [11, 12]. However, due the large amount of the trajectories, the line-based methods always cause serious visual clutter due to the cross of the polylines. To alleviate this problem, the clustering techniques are applied in the visual analytics for various dataset such as flight [8], taxi trips [22] and hurricane trajectories [1]. Moreover, advanced interaction techniques [9,13], sampling techniques [] and edge bundling techniques [33] are also developed to better present the movement patterns. The region based techniques divide the whole region into sub-regions in advance and then visualize the traffic situation before the sub-regions. These methods visualize the macro-pattern very well by leveraging different aggregation techniques such as the administrative regions [10], uniform grid [28] and spatial clustering results [26].

2.2 Interactive visualization for large dataset

The movement dataset, such as the urban traffic, always contains millions of trajectories. Limited by the rendering capability of graphic devices, generating visualizations for such scale of dataset always need to take considerable amount of time.

Advanced computing techniques have proposed in the visualization of large dataset. Chan et al. present ATLAS [5] which leverages the powerful mulit-core server and advanced caching techniques for the

efficient data communication between server and client. Piringer et al. [21] propose a multi-threading architecture for the interactive visual exploration. This method takes advantage of multi-core devices and avoids the pitfalls related to the multi-threading thus to provides quick visual feedback.

Aggregation approaches leverage the aggregation operation implemented before visualization to reduce the items will be rendered. Specifically for the spatial temporal data, these method can be further categories according to 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. Some work directly use the hierarchical administrative regions [10] as basic units and use visualization the flow by linkage between these units. All the uniform grid- and administrative region-based method are static because they are predefined. On the other hand, the region can be divided dynamically according the movement patterns. For example, MobilityGraph [26] leverages a spatial graph clustering algorithm to aggregate the tweet posts.

2.3 Data sampling techniques

Another widely applied technique to support the large data analysis is sampling technique which has been studied in both database and visualization communities. A good sampling method will reduce the data size as much as possible and still preserve the specific important feature.

Current advancing sampling techniques in the visualization domain are mostly 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. Some works design advanced sampling algorithms to preserve the meaningful data items according to the analyzing requirement such as the multi-class data analysis and hierarchical exploration [6]. Furthermore, to the usage of more visual channels of the points other than location such as color [6], size [29] and opacity are discussed. Closely related to our work, Park et al. [18] proposed the visualization-aware techniques for the scatter plot. They proposed visualization-inspired loss which effectively evaluates the visual loss of the sampling result and validates the proposed method based on three common visualization goals: regression, density estimation and clustering.

In comparison with the sampling techniques for scatter plot, the trajectory sampling is more challenging because of the complexity of the trajectories [20]. Most of the existing trajectory sampling techniques cluster the trajectories firs and then select the most representative trajectories from each cluster, which highly depend on the distance calculation and clustering algorithms [19]. Some techniques further focus on the clustering and sampling of trajectory segments instead of the whole trajectories [17].

3 Problem Formulation

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).

Many exiting visual analytics systems leverage powerful database manage system as the backend to facilitate the fast data processing. Based on the solution proposed in ScalaR [2], a common visualization framework involving sampling technique is illustrated as Figure 3, where a sampling layer is set between the backend and frontend. Since the sampling methods are always designed for complicated task, the algorithms may not be efficient enough to support the interactive data exploration. Thus the cache model is always implemented to save the sampling results. In our scenario, the users query large amount of data(e.g. all Shenzhen trajectories in one week) once and then conduct interactive multi-resolution exploration based on the sampled data, thus the method need to guarantee the visual quality well across different resolutions.

Sampling is a delta-facto solution for the problems with big data. Target at the sampling requirement, the naive solutions such as uniform random sampling cannot generate acceptable because the serious visual information loss. In this section, we first define a loss function to evaluate the visual quality between the visualization results between whole dataset and sampled subset. Then we analyze the hardness of the problem and design algorithms for it.

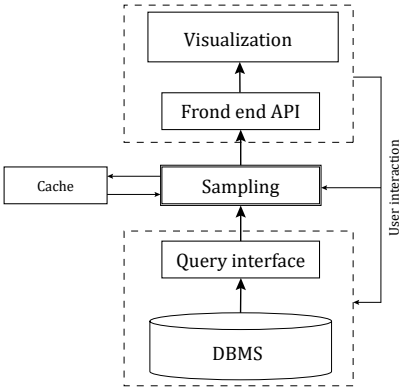


Fig. 3. A visualization framework involving sampling layer between the front-end and database management system.

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, [18] 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:

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

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[4], 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[5] 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).*

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

Algorithm 2 VQGTS(T,k)

```
1: Initialize result set R ← ∅
2: while |R| < k do
3:     R_tmp ← argmax_{T_i∈T} R ∪ T_i
4:     R ← R ∪ {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 5(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 5(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 5(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 5(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 5. 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 4. Considering Lisbon trajectory dataset, Figure 4(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 4(a). Obviously, it is much easier for us to distinguish the difference between sparse regions rather than dense regions in Figure 4(a) and (b). The major reason is that end users will treat the two dense regions in Figure 4(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 4(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 4(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 4(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 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 $\mathsf{R}_t mp$ 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 6(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 4(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.

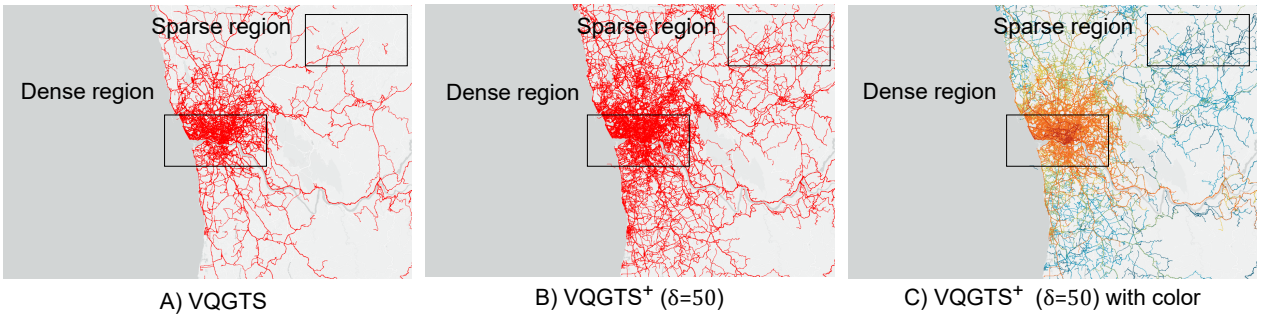
Last but not least, it is worth to point out that our advance approach VQGTS$^+$ provides more better visual quality over VQGTS at arbitrary zooming resolutions. The key technique to achieve that is it considers the zooming resolutions inherently when introducing the interpretation tolerance δ. Take Figure 6 as an example, Figure 6(a) and (b) show two different zoom-levels. The zoom level in Figure 6(a) is higher than it in Figure 6(b). As our above elaboration, our advance approach VQGTS$^+$ selects trajectory a and c at Figure 6(a). When it zoomed-out, as shown in Figure 6(b), it still capture the main sketch of the underlying dataset.

6 Evaluation

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

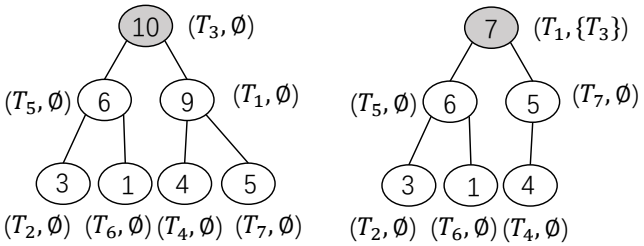
6.1 Case Study

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



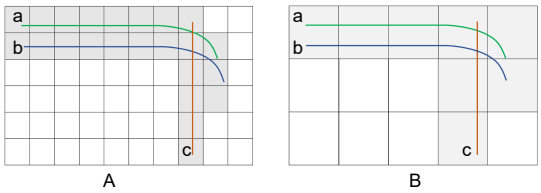
A) VQGTS B) VQGTS$^+$ (δ=50) C) VQGTS$^+$ (δ=50) with color

Fig. 4. 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. 5. Lazy updating manner illustration



A B

Fig. 6. Zoom resolution2

6.1.1 Case of Porto Distinct

Our first example uses taxi trajectories collected from 442 active taxis in Porto distinct, Portugal [6]. 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 C 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 Figure 1 C_1) indicating that the most taxi activities are around the Porto City. In addition, many trajectory clusters distributed across the land indicate the locations of other cities in Porto district(shown as the dash circle in Figure 1 C).

We compare the VQGTS$^+$ with *uniform random sampling* at the overview. Figure 1 B and 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 margin city structure are well preserved, which are shown as the dash circles in 1 E.

Furthermore, the color encoding further enhances the visualization by revealing the representativeness of the trajectories. As shown by Figure 1 E_1, there is no clear pattern can be discovered due to the dense concentration of massive trajectories. While in Figure 1 F_1, some trajectories with high representative score are highlighted by dark orange color such as F_2, 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 B and C 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. Figure1 E and 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 framework of the trajectories remains the same but the trajectories around Porto City are significantly removed because more blank space/gap can be found in the center of figure as shown in Figure 1 D_1.

Detail view of Porto Distinct: To demonstrate the effectiveness of VQGTS$^+$ at detail view, we select three regions of interest(as shown in the Figure7 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 7 D_1. Compare with the visualization of full dataset, the random sampling almost misses all trajectories in this region(as shown in Figure 7 D_2). While VQGTS samples much more trajectories than random sampling as shown in Figure 7 D_3. However, some meaningful structure are still missing such as the trajectory bundle shown in Figure 7 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 7 D_3 i) because some trajectory branches are missed. By setting the representative parameter as 64, VQGTS$^+$ generate a more confidential visualization than VQGTS shown as Figure 7 D_4.

Region 2 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 7 C_2 and C_3 present the visualization generated by VQGTS$^+$ with the representative parameters of 4 and 64. We observe that the visualization shown in Figure 7 C_3 have more details trajectory branches than Figure 7 C_3(as shown in regions c,d of Figure 7 C_2 and C_3). In this case, a larger representative parameter is more beneficial in preserving the details at this level. Furthermore, Figure 7 C_4 shows the visualization of VQGTS$^+$($delta = 64$) with color encoding. In the comparison with Figure 7 C_3, the visualization in Figure 7 further highlights the movement distribution. For instance, we observe that region f in Figure 7 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 Figure7 B_1). VQGTS$^+$($delta = 4$) greatly alleviates the visual clutter and preserves the framework which basically follows the road network as shown in Figure 7 B_2.

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

Furthermore, when setting the representative parameter *delta* as 64, the structure is more clear as shown in region b of Figure 7 B_3 and B_4, and have more trajectory details than VQGTS$^+$($delta = 4$) as shown in region *a* of Figure 7 B_2 and 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 9 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 9 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 9 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 9 C. VQGTS$^+$ with color is able to reveal the spatial distribution of the trajectories. For example, for the regions a, b in Figure 9 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 9 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 9 E), the random sampling only preserves the trajectories pass through several routes with very high traffic flow(shown as Figure 9 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 pass through G4 and G104 than Baoan Avenue, which is hard to be discovered from the Figure 9 E, F and 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 9 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 9 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 fixed size of interface and the participants perform the user study on their own computers. The system interface is shown as Figure 10. 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 size of(AA*BB).

Tasks and data generation.: All participants needed to perform four types of tasks:

- **T1. Traffic flow estimation.** The participants are asked to select the route with higher traffic flow from two candidate road segments
- **T2. Trajectory cluster identification.** The participants need to identify the centers of the cities or commercial regions from the trajectory visualization.
- **T3. Route inspection.** The participants were asked to inspect the reachable routes between the two regions by draw lines or push the button indicating uncertain.
- **T4. Visual similarity comparison.** With a given visualization generated by full dataset as ground truth, the participants were asked to rank a set of visualizations according to the similarity to the ground truth.

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 generate as follows:

Tasks of T1. We selected several visualization views which contain enough clear road structure. Then the number of trajectories passing through each road segment were counted as the traffic flow of this road segment. In the tasks of T1, two road segments 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 city/commercial region centers 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 regions centers.

Tasks of T3. We randomly chose several visualization views which contain two or more city/commercial regions. In the tasks of T3, the users were asked to draw the reachable routes between the two selected regions.

Tasks of T4. With a given center location and zoom scale, we generated the visualizations by using whole dataset, uniform random sampling and VQGTS$^+$ with different parameters. Using visualization generated by whole dataset as ground truth, users were required to score the other visualizations according to the similarity to the ground truth.

Procedure: The user study began with the introduction, in which the motivation, tasks and visual encoding were introduced. The following sessions are divided into four blocks according to the task types. Each block starts with a task tutorial, in which the participants could perform several demo tasks and were free to ask questions, thus familiarizing themselves with the interface, interaction and tasks. Then the users were asked to finish five normal tasks. During this procedure, both the answer and the time usage are recorded. After all tasks were finished, the participants would fill a questionnaire about their comments.

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

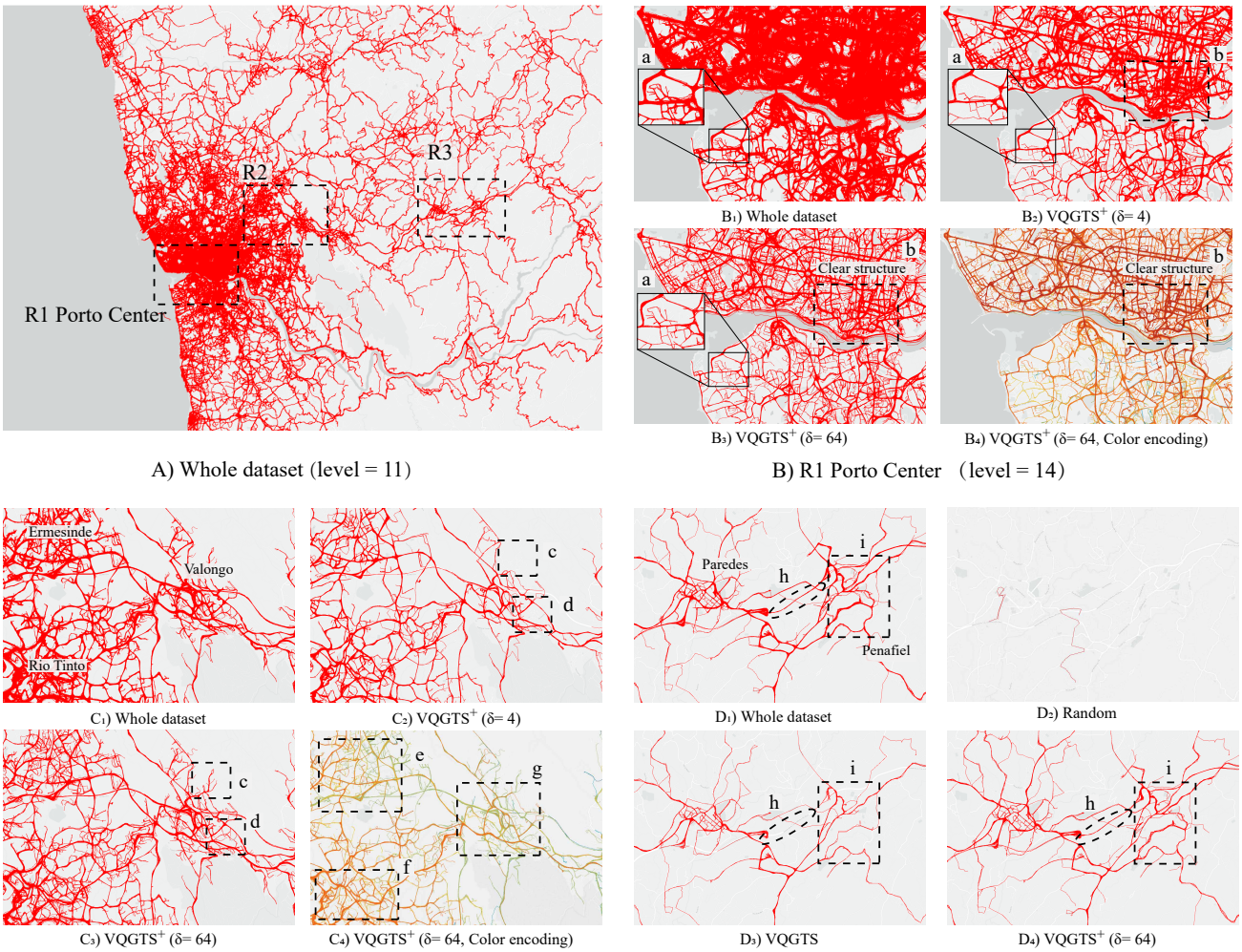


Fig. 7. Visualization at dense and sparse region respectively.

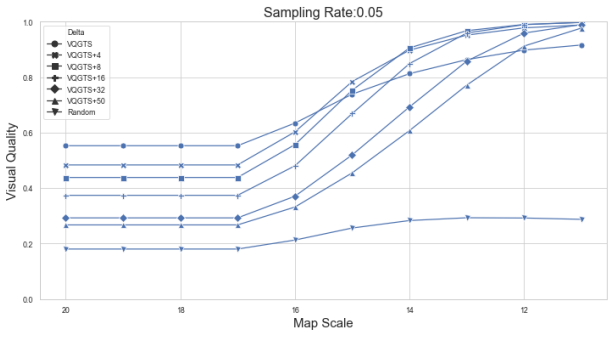


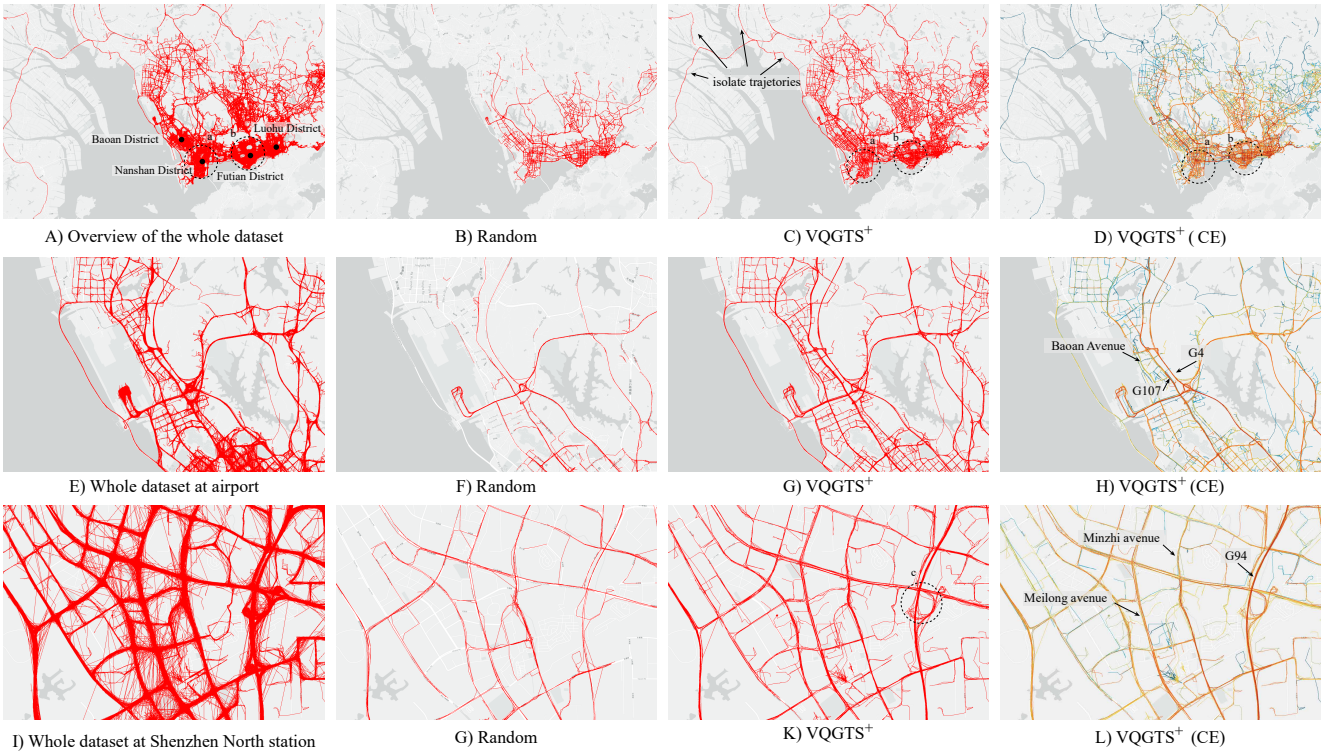
Fig. 8. Visual quality chart. X axis indicates map scale from detail view to overview; y axis indicate the visual quality.

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] 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.



A) Overview of the whole dataset B) Random C) VQGTS$^+$ D) VQGTS$^+$ (CE)

E) Whole dataset at airport F) Random G) VQGTS$^+$ H) VQGTS$^+$ (CE)

I) Whole dataset at Shenzhen North station G) Random K) VQGTS$^+$ L) VQGTS$^+$ (CE)

Fig. 9. Case study of Shenzhen.

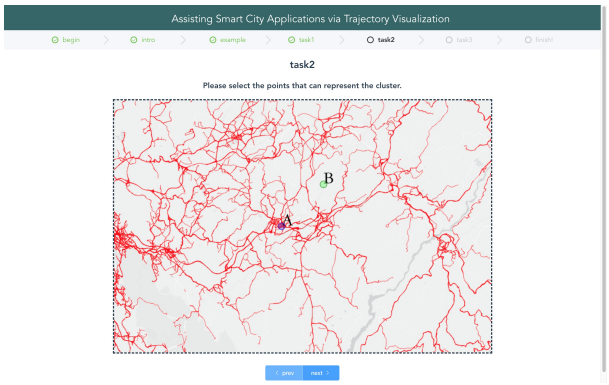


Fig. 10. The interface of user study platform.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] B. Shneiderman. Response time and display rate in human performance with computers. *ACM Computing Surveys (CSUR)*, 16(3):265–285, 1984.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.