

# Visual Fidelity Maximized Sampling for Large Trajectory Data Visualization

Category: Research

Paper Type: please specify

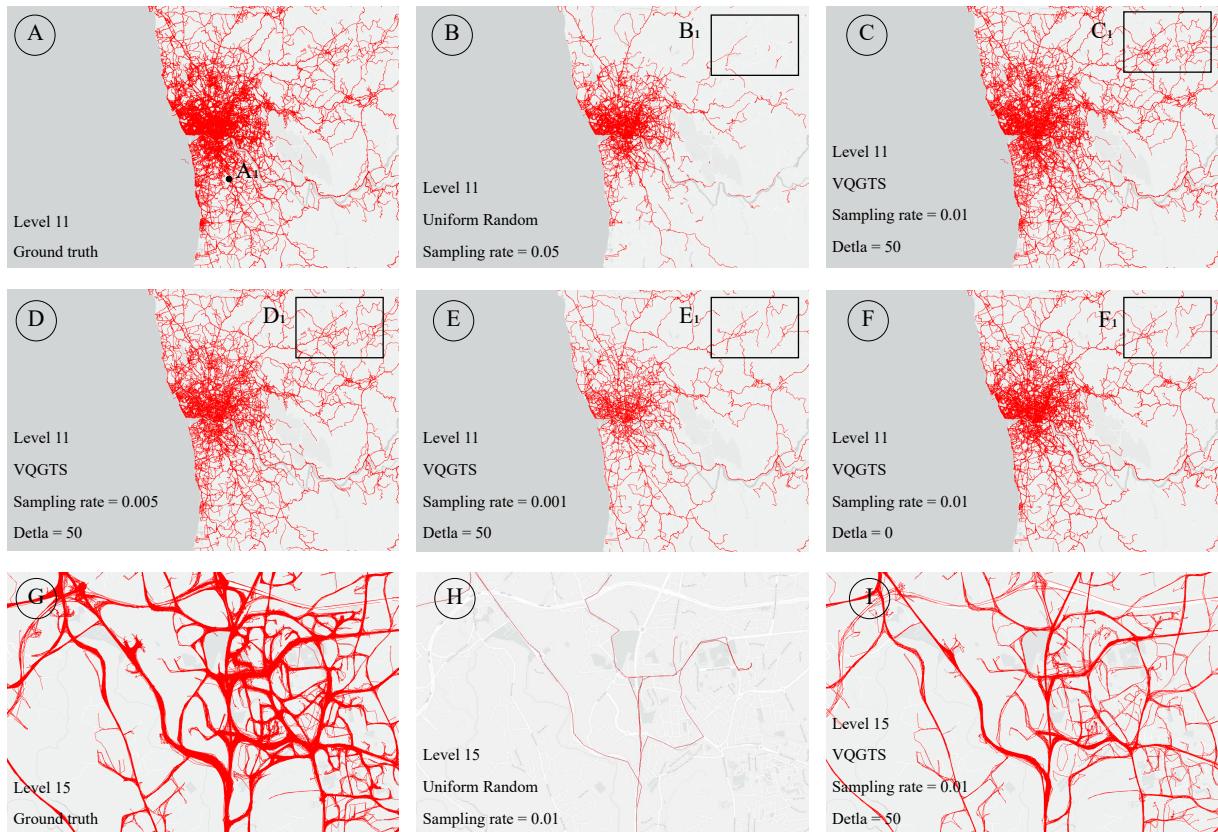


Fig. 1. Experiment results: A and G:the ground truth at level 11 and 15; B and H: uniform random sampling at level 11 and 15; C, D, E, F and I: VAGTS with different parameters

**Abstract—** In this work, we present a novel sampling approach for line-based visualization of large trajectory data, targeting at maximizing the visual fidelity of sampled trajectory visualization. We introduce a new measurement of fidelity loss using pixel-oriented proximity between two visualization images. From the metric we formulate the sampling as an NP-hard optimization problem, and develop an efficient algorithm – VFMS??, to approximate the optimal solution. VFMS encapsulates various control parameters including sampling rate and level of details (LODs), which complement with color coding to empower the effectiveness of trajectory visualization. We conduct a formal study to compare VFMS with state-of-the-art sampling methods on several large trajectory datasets. The experiments show that our approach improves visual fidelity by xxx% (numbers?) meanwhile keeps computational efficiency. Qualitative user studies further demonstrate the feasibility of our approach in XXX (concrete applications?).

**Index Terms**—Trajectory visualization, data sampling, visual quality

## 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<sup>1</sup> and Tableau<sup>2</sup> support advanced database management systems as a “backend” for

<sup>1</sup><https://www.tibco.com/products/tibco-spotfire>

<sup>2</sup><https://www.tableau.com/>

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 data size become very large. Using Porto taxi data <sup>3</sup> as an example, Table 1 demonstrates the rendering time at each dataset size. It shows that normal method takes more than 14 minutes 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.

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, **visualization aware trajectory sampling(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.

<sup>3</sup><http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html>

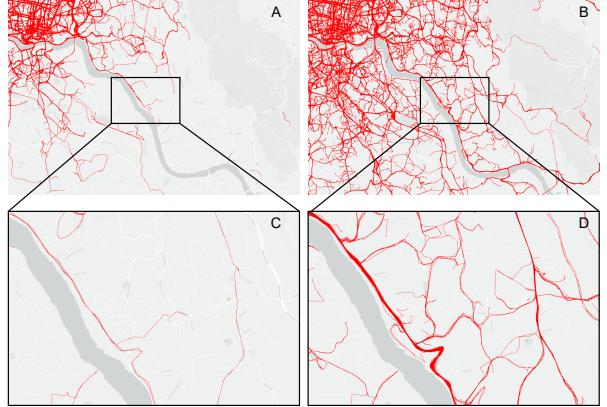


Fig. 2. Trajectory sampling generated by uniform random sampling(A,C) and VQGTS(B,D) at same sampling rate. In both high-level(A,B) and low level(C,D) view, our approach preserved more detail information about the trajectories especially for the sparse regions.

- 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.

## 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 multi-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 overrawing 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 first 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,

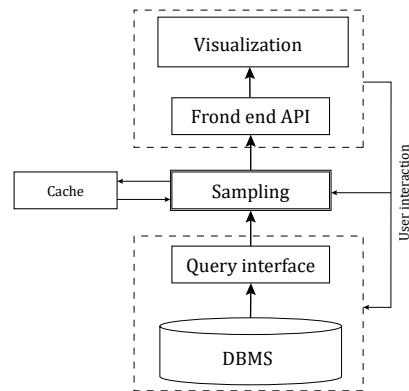


Fig. 3. A visualization framework involving sampling layer between the front-end and database management system.

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.

### 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  $R \subseteq T$ , 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 a 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  $R \subseteq T$ , 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  $loss(T, R) = \frac{V(T) - V(R)}{V(T)}$ , where  $V()$  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  $loss(T, R)$  is minimized, i.e.,

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

### 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  $\mathcal{U}$  be the universal set of all pixels in the canvas. For each trajectory  $T_i \in T$ , it consists of a set of pixels in the canvas, e.g., it is a subset of  $\mathcal{U}$ . Thus, the subset  $R$  also is a subset of  $\mathcal{U}$  as  $R = \bigcup_{R_i \in R} R_i$ .

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

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

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

It is a typical set cover maximization problem<sup>4</sup>, 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  $R \leftarrow \emptyset$ 
2: while  $|R| < k$  do
3:    $R_{tmp} \leftarrow \text{RAND}(T - R)$ 
4:    $R \leftarrow R \cup \{R_{tmp}\}$ 
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  $R \subseteq T$ , 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<sup>5</sup> 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  $T_i \in T$  is a set of pixels at highest zoom level in the canvas.

<sup>4</sup>[https://en.wikipedia.org/wiki/Maximum\\_coverage\\_problem](https://en.wikipedia.org/wiki/Maximum_coverage_problem)

<sup>5</sup><https://www.google.com/maps/preview>

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  $R \cup T_i$  at each iteration, as Line 3 shown in Algorithm 2. It terminates after  $k$  iterations and returns  $R$  as result set for rendering.

---

##### Algorithm 2 VQGTS( $T, k$ )

```

1: Initialize result set  $R \leftarrow \emptyset$ 
2: while  $|R| < k$  do
3:    $R_{tmp} \leftarrow \text{argmax}_{T_i \in T} R \cup T_i$ 
4:    $R \leftarrow R \cup \{R_{tmp}\}$ 
5: Return  $R$ 
```

---

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).*

*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.,  $b_i = \sum_{j=1}^i a_j$ ), 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.

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

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  $b_k \geq (1 - (1 - 1/k)^k) \cdot OPT \geq (1 - 1/e) \cdot OPT \approx 0.632 \cdot OPT$ .  $\square$

##### 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  $O(k \cdot m \cdot |T|)$ , where  $m$  is the maximum length of the trajectory in dataset  $T$ .*

*Proof.* The complexity analysis is straight forward as all trajectories (i.e.,  $|T|$ ) 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  $O(k \cdot m \cdot |T|)$ .  $\square$

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  $S \subset S'$  and  $T_i \notin S'$ , it holds  $|S \cup T_i| - |S| \geq |S'| \cup T_i| - |S'|$ .*

*Proof.* Let the contribution value of trajectory  $T_i$  to a given result set  $S$  as  $|S \cup T_i| - |S|$ . It is the new covered pixels of  $|T_i|$ , i.e.,  $|T_i| - |T_i \cap S|$ . It holds  $T_i \cap S \subseteq T_i \cap S'$  as  $S'$  is a superset of  $S$ . Thus, we have  $|T_i| - |T_i \cap S| \geq |T_i| - |T_i \cap S'|$ , it is equivalent to  $|S \cup T_i| - |S| \geq |S' \cup T_i| - |S'|$ .  $\square$

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  $R = \emptyset$ . 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  $R = \{T_3\}$  (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  $O(|T| + k \cdot m \cdot t \log |T|)$ , where  $t$  is the maximum number of updating among all  $k$  iterations and  $t \ll |T|$ .*

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

**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<sup>+</sup>

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<sup>+</sup>) 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<sup>+</sup> (see Algorithm 3) by incorporating a parameter  $\delta$  during trajectory selection process in VQGTS to achieve the above goal. In particular, we employ the parameter  $\delta$  to model the end user's interpretation ability at the most high level of details. Surprisingly, our advance approach VQGTS<sup>+</sup> 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<sup>+</sup> 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<sup>+</sup> 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  $\delta$  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  $\delta$  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<sup>+</sup>( $T, k, \delta$ )

---

```

1: Initialize result set  $R \leftarrow \emptyset$ 
2: Initialize augmented result set  $R^+ \leftarrow \emptyset$ 
3: while  $|R| < k$  do
4:    $R_{tmp} \leftarrow \text{argmax}_{T_i \in T} \delta R \cup T_i$ 
5:    $R \leftarrow R \cup \{R_{tmp}\}$ 
6:    $R^+ \leftarrow R^+ \cup \text{augment}(R_{tmp}, \delta)$ 
7: for each  $T_i$  in  $T - R$  do            $\triangleright$  Popularity encoding
8:    $R_j \leftarrow \text{argmin}_{R_i \in R} \text{augment}(R_i, \delta) - T_i$ 
9:    $R_j.cnt += 1$ 
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.,  $T - R$ . 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<sup>+</sup> 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<sup>+</sup> 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  $\delta$ . 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<sup>+</sup> 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.

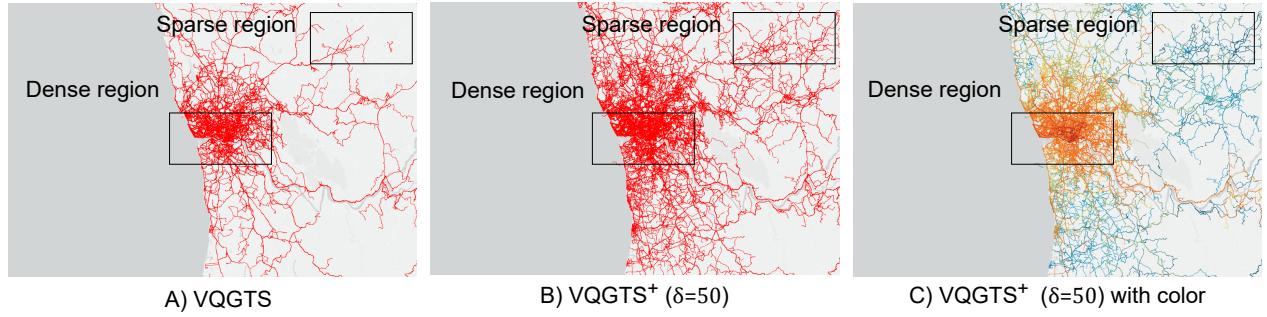


Fig. 4. Detail-aware visual quality guaranteed trajectory sampling, A: VQGTS, B: advanced VQGTS (VQGTS<sup>+</sup>), C: VQGTS<sup>+</sup> and popularity encoded by color.

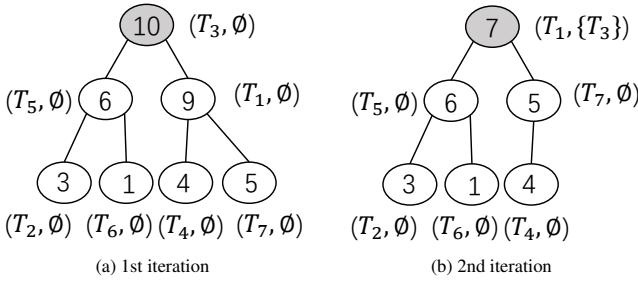


Fig. 5. Lazy updating manner illustration

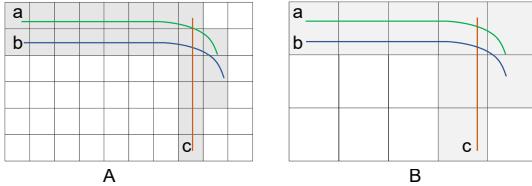


Fig. 6. Zoom resolution2

## 6 VISUALIZATION AND IMPLEMENTATION

Platform/language/space-time consume

## 7 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.

### 7.1 Experimental results

In this section, we will briefly evaluate our method by case studies. We will first make a brief comparison between the VQGTS and uniform random sampling at different granularities. Then we will show how VQGTS further improves the visual quality by taking the [representative](#) into consideration. At last, we will extend our method to more dataset and demonstrate the effectiveness of propsoed method.

#### 7.1.1 VTGS and uniform random sampling

Our first example uses taxi trajectories collected from 442 active taxis in the city of Porto, Portugal<sup>6</sup>. Figure 7 shows the comparison among the ground truth, uniform random sampling and the VQGTS. Both random sampling and VQGTS method take 0.01 as the sampling rate. In the overview(map scale at 11), based on the ground truth(shown as figure 7 A), the random sampling(shown as figure 7 C) has a very poor the visual quality especially for the boundary region because

<sup>6</sup><http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html>

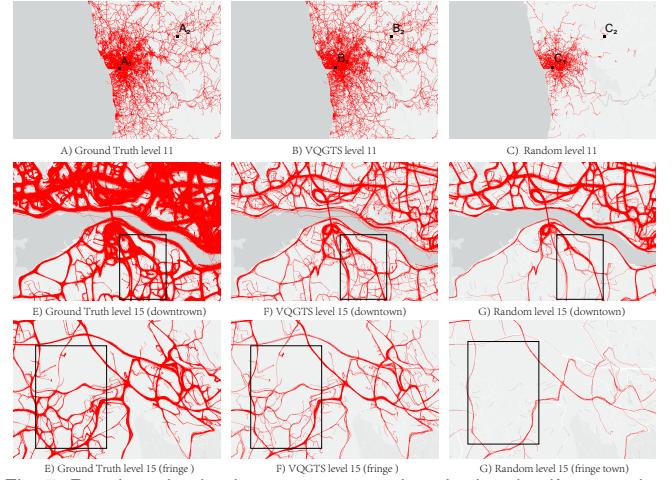


Fig. 7. Result evaluation between proposed method and uniform random sampling. The images in the three columns indicate the visualization results for full dataset, proposed method, and uniform random sampling respectively. The visualizations in the first row shows the overview. The visualizations in the second the third row indicate the detail level visualization of region 1 and region 2.

very few trajectories are preserved by the sampling methods. That's because in the city, most of the taxi trips concentrate to the downtown, thus the sampled trajectories are more likely located in the downtown, which lead to a serious visual information loss. But our method(shown as figure 7 B) are visually very close to the ground truth even in the margin regions. We further compare the visualization at the detail level(map scale at 15). The figures in second rows depict the detail view of the ground truth, VQGTS and sampling at the region shown in Figure 7 a, which is the city center of Porto. Comparing with VQGTS, random sampling preserve the general trajectory framework but miss the details(As the figure 7F, G shown). However, for the fridge of the city where the trajectories are spare distributed, the VQGTS has clear advantage than random sampling because it well preserve the detail structure of trajectories as shown in the black region in the figure 7 H and I. This case demonstrates our proposed sampling method works well in both overview and detail view compared with the random sampling method.

#### 7.1.2 Representativeness parameters analysis

Figure 5 demonstrates the effectiveness of our method in the level of overview. The VQGTS<sup>+</sup> can preserve more details of the trajectory especially in the sparse regions and the VQGTS<sup>+</sup>. We further explore how the value of  $\delta$  affects the visualization results under different parameters including the spatial locations, the sampling rates and map resolution. Figure 8 shows the overview of the visualization visual quality among all these parameters. The x axis indicate the map resolution from the detail view to overview(11 to 20). The y axis indicate the

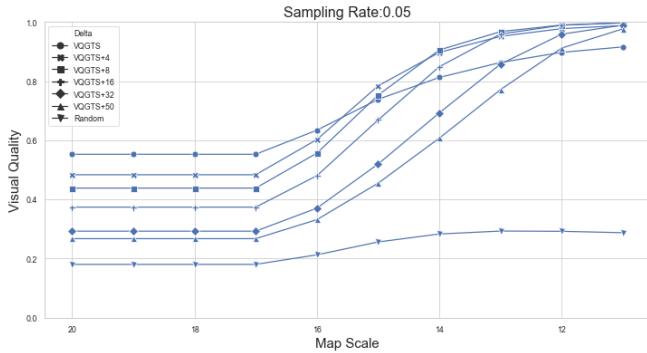


Fig. 8. Visual quality chart. X axis indicates map scale from detail view to overview; y axis indicate the visual quality.

visual quality and the line with glyph indicate the sampling algorithms including uniform random sampling, VQGTS and VQGTS<sup>+</sup> with different representative parameters. The figure shows the both VQGTS and VQGTS<sup>+</sup> have better performance than random sampling and all of these linechart show the trend that from the detail view to the global view, the visual qualities keep increasing.

We further analyze the **representative** parameters by drilling into the real cases. Figure 9 demonstrates the visualization among a dense region(Figure 9 A) and a sparse region(Figure 9 B) respectively, all these trajectories are visualized under the map level of 14 and the sampling rate of VQGTS<sup>+</sup> is 0.005. In the dense region, the ground truth are shown with a serious visual clutter, which is hard to reveal the trajectory distribution patterns. The VQGTS<sup>+</sup> with VQGTS<sup>+</sup> set to 4 and 50(Figure 9 A<sub>1</sub> and A<sub>2</sub>) alleviate the visual clutter. In comparison, sampling result of the VQGTS<sup>+</sup> with larger  $\delta$  have thinner bundles and result in clearer trajectory structure shown as region a and b of figure (Figure 9 A<sub>1</sub>). As for the sparse region (Figure 9 B), the sampling result of VQGTS<sup>+</sup> with different  $\delta$  setting are very close to each other in visual. When narrowing down to the specific region, we further notice that the VQGTS<sup>+</sup> with large  $\delta$  can preserve the detail structure as shown in the Figure 9 c and d.

Sampling results generated by VQGTS<sup>+</sup> can clearly show the trajectory structure compare with the ground truth, but the trajectory amount are still hard to be preserved directly by the visualization. We further encode the repressiveness by using color, which can be used to reveal the trajectory amount at of the bundle. As shown in figure 9 e, the region in figure 9 B<sub>4</sub> is messed up, the region in B<sub>1</sub> and B<sub>2</sub> can roughly indicate the trajectory amount by the thickness of the bundles, while trajectory distribution in B<sub>3</sub> is encoded with both color the bundle thickness which can show the trajectory distributions.

## 7.2 Expert overview

## 8 DISCUSSION

Dicussion section.

## 9 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<sup>+</sup> 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<sup>+</sup> 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.

For future work, we first plan to improve the algorithm efficiency by leveraging the advanced database techniques such as the indexing technique or use GPU acceleration.

## 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.
- [2] L. Battle, M. Stonebraker, and R. Chang. Dynamic reduction of query result sets for interactive visualzaton. 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. Hamrahan. 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. Kłosowski, 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. Tissiores, 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. Ntouts, 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.

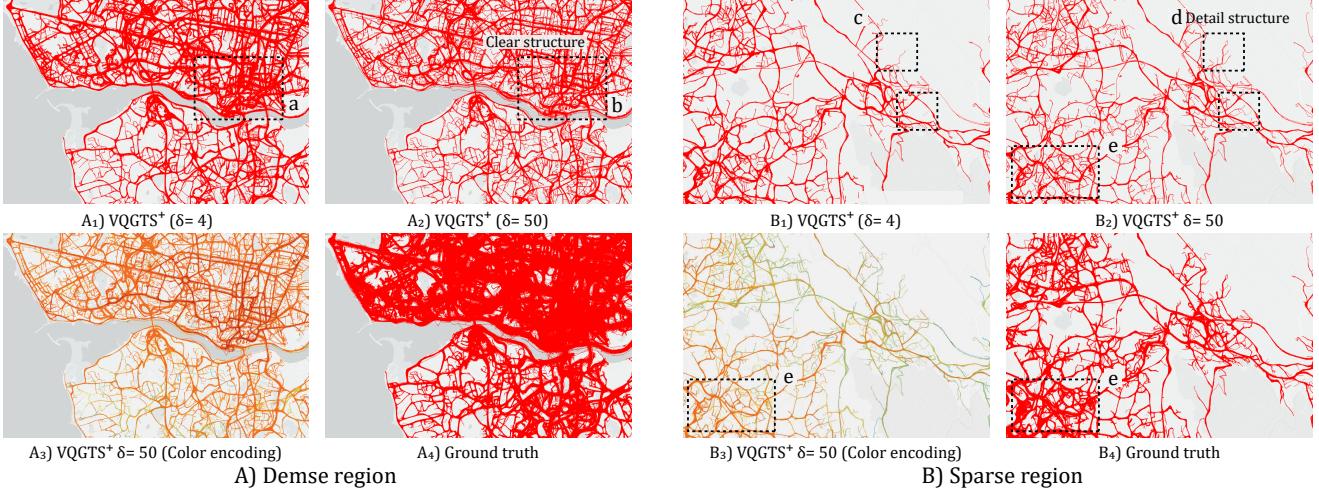


Fig. 9. Visualization at dense and sparse region respectively.

- [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. Muiigg, 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.