Visual Quality Guaranteed 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—Line-based visualization has been widely used to present trajectory data. However, when the data size grows very large, it will take a considerable amount of time to render the graphics. Sampling techniques can mitigate the problem by directly shrinking the dataset, thus to reduce the rendering work of graphic devices. However, the traditional sampling techniques may lead to the loss of visual information and cannot directly be applied in line-based trajectory visualization. This work presents a novel approach, visual quality guaranteed sampling(VQGTS), specifically for the data reduction of large trajectory data visualization. We first propose a loss function to measure the visual quality loss between the visualization of the sub-dataset and the whole dataset. Then we minimize the loss function by transforming it as an optimization problem and propose efficient solutions. Several parameters and color encoding which can empower the effectiveness are discussed. The experiments on several large trajectory datasets demonstrate the effectiveness of the proposed method. The user study further shows the usability of our methods in the exploration of large trajectory data.

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 **(size) of trajectory data can be collected every day, which records (distance) by sampling locations. The analysis over these databases can be applied in many fields such as traffic management [28], urban planning, route recommendation [36] and location-based services [15, 35].

Visualizing trajectories is a challenging task. 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 [] and Tableau []

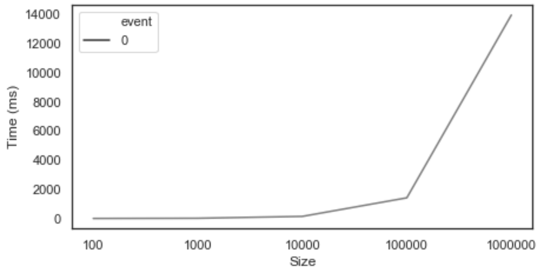


Fig. 2. The latency time for generating line-based visualization at each datasize.

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 [27,33], edge bundling [26,34] and density map [14,24]. 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 [] as an example, figure 2 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 [25]. 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 good sampling)))

In our method, we extend the motivation of visualization-aware sampling to trajectory dataset. We propose a novel sampling strategy, **v**isualization **a**ware **t**rajectory **s**ampling(VATS), that produces high-visual-quality line-based trajectory visualization at certain degree(arbitrary) zooming resolutions. In this paper, we first proposed the visual fidelity loss function which effectively evaluates the visual loss of the sampling method. Then we minimize the loss function by transforming this problem to an optimization problem. Several solutions for efficiently solving the optimization problem are discussed. Figure 3 depicts an comparison among the ground truth, uniform random sampling and our proposed method. By limiting the sampling set size, the proposed method generates a higher-fidelity visualization and support the multi-resolution very well.

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.

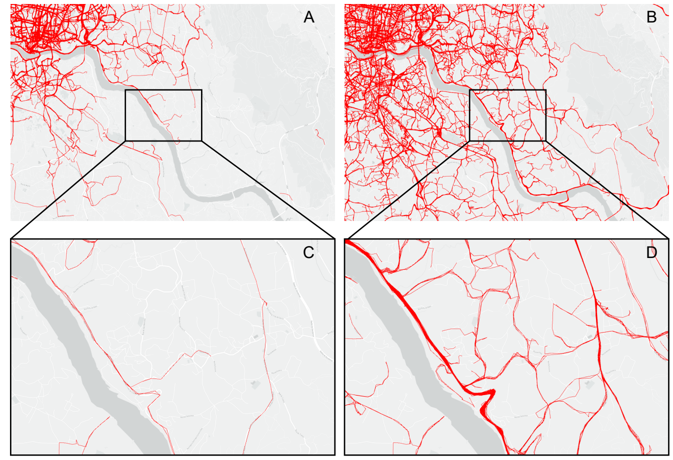


Fig. 3. Trajectory sampling generated by uniform random sampling(A,C) and VQGTS(B,D) at same sampling rate. In both high-level(A,B) and low level(C,D) view, our approach preserved more detail information about the trajectories especially for the sparse regions.

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 [4,16,32], 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 [3,31]. 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 [23] and hurricane trajectories [1]. Moreover, advanced interaction techniques [9,13], sampling techniques [] and edge bundling techniques [34] 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 [29] and spatial clustering results [27].

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. [22] 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 [29] 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 [27] 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 [30] 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 [21]. 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 [20]. Some techniques further focus on the clustering and sampling of trajectory segments instead of the whole trajectories [17].

3 Problem Formulation

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

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* $loss(\mathsf{R},\mathsf{T})$ *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.,

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

3.1 Hardness analysis

In real-world applications, the pixels in canvas will be rendered by different colors according to the specified visualization scheme. For the sake of presentation, we analyze the hardness of our research objective with a simple render manner of visualization result. In particular, for each pixel in the canvas, it will be rendered if there is a trajectory pass through it, otherwise it will not be rendered. Suppose each pixel in the canvas has an unique id, let $\mathscr{U}$ be the universal set of all pixels in the canvas For each trajectory $\mathsf{T}_i \in \mathsf{T}$, it consists of a set of pixels in the canvas, e.g., it is a subset of $\mathscr{U}$. Thus, the subset R also is a subset of $\mathscr{U}$ as $\mathsf{R} = \cup_{\mathsf{R}_i\in\mathsf{R}}\mathsf{R}_i$.

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

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

It is equivalent to select sized-k trajectory set R from T which $\cup_{\mathsf{R}_i\in\mathsf{R}}\mathsf{R}_i$ is maximized. It is a NP-hard problem as we proved in Lemma 1.

Lemma 1 (NP hard)**.** *The sampling-based trajectory visualization problem (see Problem 1) is NP-hard.*

We omit the proof of Lemma 1 as it is a typical set cover maximization problem[1], which is a well-known NP-hard problem in literature.

[1] `https://en.wikipedia.org/wiki/Maximum_coverage_problem`

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)

```
Initialize result set R ← ∅
while |R| < k do
    R_tmp ← RAND(T − R)
    R ← R ∪ {R_tmp}
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 pixel size will be updated at different zoom level. For example, Google map[2] 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 highes 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.

Algorithm 2 VQGS(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
```

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

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

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

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 2.

Lemma 2 (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 3.

Lemma 3 (Submodularity)**.** *Given a trajectory* T_i *and two result sets* S, S', *where* $\mathsf{S} \subset \mathsf{S}'$ *and* $\mathsf{T}_i \notin \mathsf{S}'$, *it holds* $|\mathsf{S} \cup \mathsf{T}_i| - |\mathsf{S}| \geq |\mathsf{S}' \cup \mathsf{T}_i| - |\mathsf{S}'|$.

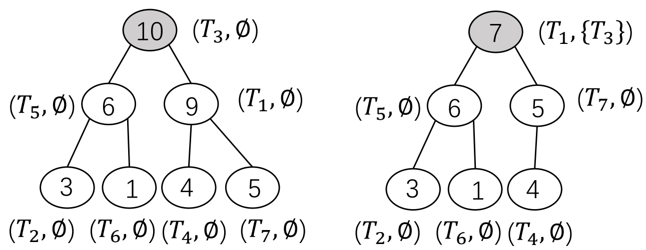
Proof. Let the contribution value of trajectory T_i to a given result set S as $|\mathsf{S} \cup \mathsf{T}_i| - |\mathsf{S}|$. It is the new covered pixels of $|\mathsf{T}_i|$, i.e., $|\mathsf{T}_i| - |\mathsf{T}_i \cap \mathsf{S}|$. It holds $\mathsf{T}_i \cap \mathsf{S} \subseteq \mathsf{T}_i \cap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

With the help of submodularity property, it reduces a lot of unnecessary trajectory updating computations. In particular, we maintains a max-heap for the number of uncovered pixels of each trajectory w.r.t., result set R. Figure 4(a) shows a tiny max-heap example about the numbers of uncovered pixels of each trajectory from T_1 to T_7 with result set $\mathsf{R} = \emptyset$. At the 1st iteration, the root node of the max-heap will be selected, i.e., T_3 in Figure 4(a). At the 2nd iteration, the number of uncovered pixels of the root node T_1 is updated to 7 w.r.t. result set $\mathsf{R} = \{\mathsf{T}_3\}$ (see gray node at Figure 4(b)). T_1 will be selected at the 2nd iteration without updating the number of uncovered pixels in other trajectories, i.e., all white nodes at Figure 4(b) as the submodularity property holds.

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

Lemma 4 (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* $t \ll |\mathsf{T}|$.

[2] https://www.google.com/maps/preview



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

Fig. 4. Lazy updating manner illustration

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.

We next elaborate an "one-to-many" strategy to further optimize the visual quality of our proposal. Recalling we use the highest zoom level to define the pixel size in the canvas. Thus, our visual quality guaranteed sampling algorithm is zoom-level oblivious, e.g., it guarantees the visual quality of result set R at every zoom level. However, users always do not use/need the highest zoom level in visualization applications. For example, Google map shows city and streets at zoom level 1 and 15, respectively [3]. Motivated by the above observation, we devise "one-to-many" strategy by introducing a visual tolerance parameter δ to optimize the visual quality for users. Specifically, suppose the pixel with location (x, y) in canvas is covered by result set R, the "one-to-many" strategy will ignore all the pixels around (x, y) within δ offset distance, i.e., all pixels from $(x - \delta, y - \delta)$ to $(x + \delta, y + \delta)$ will be skipped. We will demonstrate the effectiveness of the visual tolerance δ in experimental evaluations.

5 Implementation

Platform/language/space-time consume

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 Experimental results

- Data description
- Anomaly case
- Visual quality case

6.2 User study

- Visual similarity
- Identify outliers
- Trustiness

6.3 Expert overview

7 Discussion and future work

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 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, E. Kotsifakos, E. Frentzos, and Y. Theodoridis. Clustering trajectories of moving objects in an uncertain world. In *2009 Ninth IEEE international conference on data mining*, pp. 417–427. IEEE, 2009.

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

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

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

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

[3] `https://developers.google.com/maps/documentation/`

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

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

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

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

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

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

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