Visual Quality Guaranteed Sampling for Big Trajectory Data Visualization

Category: Research

Paper Type: please specify

Fig. 1. In the Clouds: Vancouver from Cypress Mountain. Note that the teaser may not be wider than the abstract block.

Abstract—Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Index Terms—Radiosity, global illumination, constant time

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. 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.

Using ** dataset as an example, figure 2 demonstrates the rendering time at each dataset size, which shows that normal method takes more than ** minutes to generate the visualization, which is far beyond the human-acceptable response time for the interactive exploration [25].

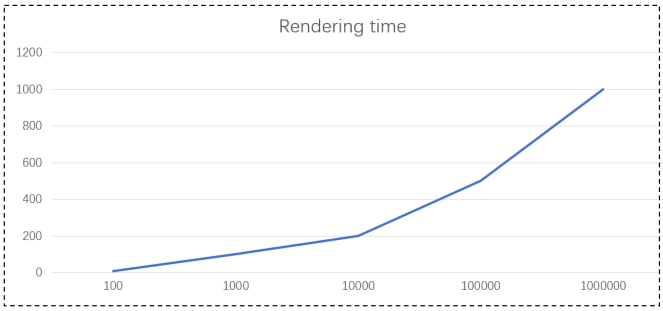


Fig. 2. x-data size; y-rendering time

To handle the big dataset, many visualization products such as Spotfire [] and Tableau [] support advanced database management systems as a "backend" for the efficient data processing the query. 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

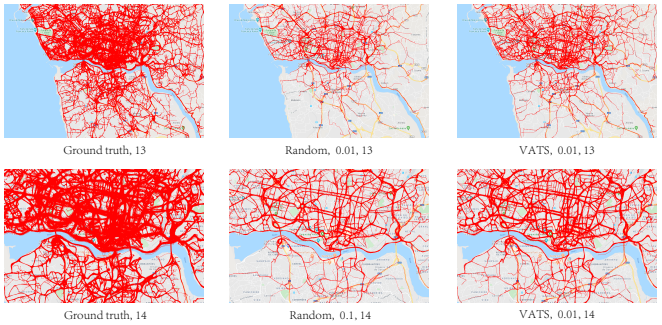


Fig. 3. three columns (ground truth, random sampling, proposed method), two rows(top level, middle level)

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.

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.

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 visualization of large dataset. Chan et al. present ATLAS [5], which leverage 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 categoried according to how to generate the spatial partitions. For exmaple, 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 regions-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

Sampling methods reduce the data volume by directly selecting the representative items. 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 data items. Some works design advanced sampling algorithms to preserve the meaningful data items according to the analyzing requirement such as the mulit-class data analysis [6] and hierarchical exploration []. 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 first [19] and then select the most representative

trajectories from each cluster, which highly depend on the distance calculation [20] and clustering algorithms. Some techniques further focus on the clustering and sampling of trajectory segments instead of the whole trajectories [17].

3 Problem Statement

3.1 Problem Identification

- Definition of visual quality
- Evaluate visual quality

3.2 VATS Overview

4 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 academic and industry. The reasons are (i) the size of trajectory data is very large (e.g., several GB in an hour), (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 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. Specifically, we focus on visualization-aware sampling method to visualize the large-scale trajectory dataset efficiently and effectively. The major challenges of our research problem are: (i) how to define "visualization-aware" theoretically? (ii) how to solve visualization-aware sampling problem efficiently.

In this work, we first formate "visualization-aware" formally by defining the loss function between the ground truth and sampled visualization result. With the loss function, we then analyze the hardness of the problem, and devise visualization result guaranteed solution for it lastly.

Problem 1 *Given a large-scale trajectory dataset* T *and an integer* k*, the visualization-aware sampling method for large-scale 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.*

In general, there are many ways to define the loss function *loss* between visualization quality/utility difference between the sampled trajectory subset R and original dataset T from the user's perspective. For example, [18] defined point-based loss function for very large scatter points visualization. In this paper, we define the loss function by the fact from the visualization quality from the perspective of users. Intuitively, the visualization quality difference of two trajectory sets visualization results depends on the user specified visualization level of details (a.k.a., LOD).

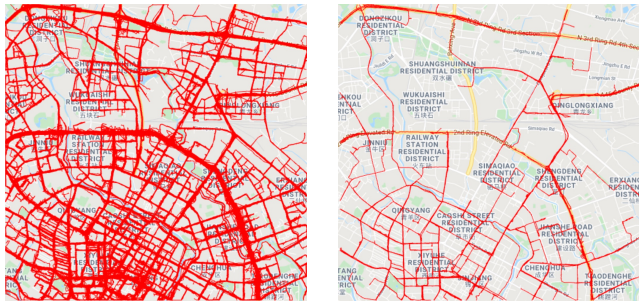
Given an empty canvas (e.g., displaying device) with a user specified level of details, the visualization process is rendering the trajectories into canvas with the given level of details (e.g., the number of pixels in each row and each column). Considering a data objects set T and a subset of data objects $\mathsf{R} \subseteq \mathsf{T}$, The visualization quality loss between R and T is defining as the different pixels of the visualization results of R and T in the canvas with specified LOD.

Given a trajectory data set T and an integer k, the research objective of our visualization-aware sampling method is selecting a sized-k subset of trajectories R which minimize the visualization quality loss function $loss(\mathsf{T},\mathsf{R})$. Formally, the loss function is defined as $loss(\mathsf{T},\mathsf{R}) = \mathsf{V}(\mathsf{T}) - \mathsf{V}(\mathsf{R})$. Thus, our research objective is:

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

4.1 Hardness analysis

For the sake of presentation, we analyze the hardness of our research problem with simple render manner of visualization result. Specifically, the visualization result only test whether there is a trajectory in dataset on every pixel in the canvas. If yes, it will render that pixel in the canvas, otherwise it will not. However, the rendering color of each pixel on the canvas is related to the total number of trajectories on it in practice, we will elaborate shortly.



(a) Dataset T (b) Uniform sampling R

Fig. 4. Visualization results of trajectory set in Chengdu

With the above simple render manner, we suppose each pixel in the canvas has an unique id, $\mathscr{U}$ is 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 the number of different pixels between the two canvases which rendered by T and R, respectively. Obviously, the visualization result of T is a constant value, denotes as C/ Our research objective can be transformed as follows:

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

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.

Lemma 1 (NP hard) *Given a trajectory dataset* T *and an integer* k*, the visualization-aware sampling problem, i.e.,* $\max_{\mathsf{R}\subseteq\mathsf{T},|\mathsf{R}|=k}\cup_{\mathsf{R}_i\in\mathsf{R}}\mathsf{R}_i$ *is NP-hard.*

Bo: We will prove it by reducing maximal set cover problem to it. Discuss with Qiaomu about the general format in VIS.

5 Problem solving

In this section, we first introduce the uniform random sampling algorithm for visualization-aware sampling problem in Section 5.1. We then propose a greedy algorithm for it in Section 5.2. However, both random sampling algorithm and greedy algorithm do not guarantee the approximate ratio of the visualization result. We then propose an approximate algorithm for it in Section **??**.

5.1 Uniform Random Sampling

The straight forward solution for the visualization-aware sampling problem is randomly select k trajectories from T, then render these selected k trajectories into the canvas. The uniform random sampling algorithm has excellent performance. However, the trade-off is that it does not provide any guarantee on the visualization results. Trajectory data T [**?**] consists of almost 10K trajectories which collected by Didi company. The visualization result of the whole dataset T is illustrated in Figure 4(b). Figure 4(b) shows the visualization result of uniform random sampling result of T with $k = 100$. The difference between the visualization results in Figures 4(a) and (b) are obvious from user's perspective.

5.2 Greedy Algorithm

In order to improve the visualization quality of visualization-aware sampling problem, we devise greedy algorithm in this section. In general, the visualization result quality is related to the user zoom level. For example, Google map[1] provides 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). The size of each pixel in the canvas is defined by the highest zoom level. For each trajectory $\mathsf{T}_i in \mathsf{T}$ is a set of pixels in the canvas. The pesudocode of the greedy algorithm is presented in Algorithm 1. Specifically, it finds the trajectory T_i in T which maximize the union set of $\mathsf{R} \cup \mathsf{T}_i$ at each iteration. It terminates after k iterations and returns R to graphics device for rendering.

Algorithm 1 Greedy(T,k)

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

Theorem 1 *Algorithm 1 provides a* $1-(1-1/k)^k \geq (1-1/e) \approx 0.632$ *approximation result for the visualization-aware sampling problem.*

5.3 Performance Optimizations

Bo: Direction I (for visual performance): pixel cover criteria, i.e., if we render pixels at (x,y), then all pixels in $[(x-\delta, y-\delta), (x+\delta, y+\delta)]$ will be skipped directly.

Bo: Direction II (for time cost): trajectory representation, i.e., we formulate the universal set by the roads in the map, then, before we call road-matching to generate the set of each trajectory, after that we then incurs greedy algorithm.

6 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

- Data description
- Anomaly case
- Visual quality case

7.2 User study

- Visual similarity
- Identify outliers
- Trustiness

7.3 Expert overview

8 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.

[1] `https://www.google.com/maps/preview`

[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.

[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.