OD-Wheel: Visual Design to Explore OD Patterns of a Central Region

Min Lu[1,2] Zuchao Wang[1] Jie Liang[1] * Xiaoru Yuan[1,2] †

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University
2) Center for Computational Science and Engineering, Peking University

Abstract

Understanding the Origin-Destination (OD) patterns between different regions of a city is important in urban planning. In this work, based on taxi GPS data, we propose OD-Wheel, a novel visual design and associated analysis tool, to explore OD patterns. Once users define a region, all taxi trips starting from or ending to that region are selected and grouped into OD clusters. With a hybrid circular-linear visual design, OD-Wheel allows users to explore the dynamic patterns of each OD cluster, including the variation of traffic flow volume and traveling time. The proposed tool supports convenient interactions and allows users to compare and correlate the patterns between different OD clusters. A use study with real data sets demonstrates the effectiveness of the proposed OD-Wheel.

Index Terms: H.5 [INFORMATION INTERFACES AND PRESENTATION]: User Interfaces—Graphical user interfaces;

1 Introduction

It is important to understand the movement of human beings and objects in large cities. Such movement, known as the Origina-Destination (OD) pattern, can be indicated from many different type of spatial-temporal data. In this work, we use taxi GPS data to study the Origin-Destination (OD) patterns.

Taxi GPS data is a typical kind of spatial temporal data. As considerable number of taxis are incessantly travelling around the city everyday, taxi GPS data are considered representative of city traffic. Therefore, it is widely studied in visualization and data mining communities. For example, by studying the flow volume and travel time of taxis, regular commuting patterns can be extracted [18]. Hot OD spots are extracted and the route diversity are explored [13]. A route's travel time variation is explored from spatial and temporal aspects [15]. By studying the traffic jam, irregular traffic congestion patterns can be identified [24]. Besides, as land-use patterns can be inferred [17], taxi GPS data also finds its applications in urban planning problems. Instead of studying all ODs as a whole, our analysis focuses on a central region. We allow users to select a subset of trajectories. Then, we group them into OD clusters and explore the dynamics of these OD clusters.

To support above analysis, (1) we develop an intuitive visual query interface to filter taxi trajectories based on a central region, (2) we employ a clustering algorithm to group taxi trajectories into OD clusters, (3) we propose a visual design to describe, compare and correlate the dynamic patterns on those OD clusters. Our contributions include:

- A novel visual design, OD-Wheel, to visualize the temporal dynamics of OD clusters.

- A comprehensive visual analytic tool to explore the OD patterns related to a central region.

*e-mail: {lumin.vis, zuchao.wang, christy.jie}@gmail.com
†e-mail: xiaoru.yuan@pku.edu.cn

Following a review of the related work in Section 2, we give an overview of our system and data in Section 3 followed by spatial and temporal filters in Section 4. In Section 5, we explain the visual design on OD-Wheel in detail. In Section 6, we introduce the implementation details and illustrate a use study to show the effectiveness of our system. Meanwhile, we have a discussion in the end of Section 6 before the conclusion.

2 Related Work

Trajectory data can be transformed into OD data with aggregation methods [3]. Such OD data can be visualized with a set of techniques, including flow map, OD matrix and OD map. Flow map [22] shows the origins and destinations as nodes on a geographic map. The entity flows on the ODs are represented as directed links connecting the origin and destination nodes. It is a special type of node-link diagram. Wang et al. [25] propose a system centering on dynamic graph network to visualize the traffic flow between RFID. Flow map is very intuitive, but it suffers from serious visual clutter. Possible solutions to reduce such visual clutter include edge-filtering [20], edge-bundling [11], node hierarchy [7] or allowing only one origin or one destination [19]. OD matrix represents origins and destinations as rows and columns of a matrix [2]. Each cell encodes a OD, whose flow magnitude is represented as color of the cell. Due to lack of spatial information, OD matrix are usually used in conjunction with a geographic map. OD map [27] makes a balance between clutter free and spatial information preservation. It partitions a geographic region to subregions with a 2D grid and define ODs between the subregions. In our work, we focus on the ODs related to a central region, and group those ODs into clusters. We display the dynamics of traffic flow volume and travel time on the clusters with a hybrid linear-circular design.

To show the temporal information, Aigner et al. [1] gives a systematic view of the visualization of time-oriented data. Animation methods are usually adopted. For example, Whisper [5] visualizes the information diffusion in social media with animated pathways between origin tweet and retweet. Although being intuitive, animations usually do not work well for analysis tasks [21]. Therefore, there are many static methods proposed. If the data have some known periodicity, then calender view [23] and spiral view [26] can be good options. However, for general temporal data, timeline is the most popular methods. In a timeline, the x-axis represents time, and y-axis represents an attribute that changes with time. To compare different temporal data, each data can be visualized by a timeline. Then those timelines can be juxtaposed [16] or superimposed [10] for visual comparison. In case of juxtaposition, similar timelines can be put close to each other [14]. Temporal data analysis can benefit from rich user interactions. Zhao et al.'s KronoMiner [28] support clicking based time range selection and dragging based data comparison. Their system is based on a circular layout of timelines.

Temporal data visualization techniques have been used to analyze OD data. In Ferreira et al.'s taxi trip exploration work [6], once users select multiple ODs and execute the query, the dynamics of trip number on each OD will be shown in a timeline. TripVista [8] present a direction encoded ThemeRiver [9] to illustrate the dynamic change of trafc ow direction. In Boyandin et al.'s Flowstrates [4], they analyze the refugeee flow with a specially designed

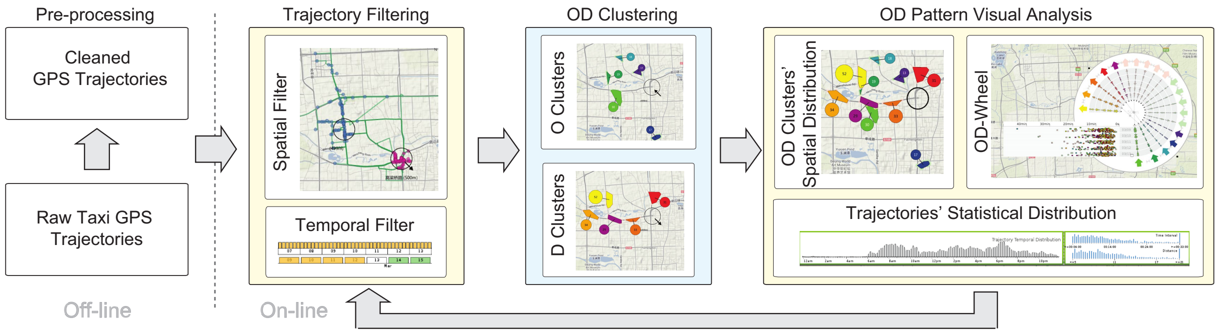

Figure 1: System Pipeline: the whole tool consists of four parts, from the left to the right, off-line trajectory preprocessing, interactive trajectory filter, automatic OD clusters extraction and OD pattern visual analysis.

three part interface. Two maps are used for origin and destination selection, while a heatmap is embedded between them to compare the flow volume between different ODs. Here the heatmap is in fact a juxtaposition of timelines. In our work, we use a circular design for flow volume comparison, as that in KronoMiner. In addition, we also embed a linear timeline component for travel time comparison. Our visual design and the whole implemented whole is aiming to analysis tasks and supporting interactions specificaly for OD exploration.

3 Data and System Overview

Our GPS dataset is from real taxi operations recorded in Beijing. In 24 days, from March 2nd to 25th, 2009, the trajectories of 28,519 taxis are collected, consisting of 379,107,927 sampling points, with an overall data size of 34.5GB. Each sampling point contains record of *time*, *latitude*, *longitude*, *speedmagnitude*, *direction*, plus a boolean *passengerState*, indicating whether there are passengers in the taxi. The sampling rate is approximately one point per 30 seconds.

To support exploring OD patterns of a selected central region, the system consists of four parts: data preprocessing, trajectory filtering, OD clustering and OD pattern visual analysis, as shown in Figure 1.

Initially, following the preprocessing steps [24], GPS data is cleaned and matched to road-network off-line. The system takes the preprocessed trajectories as input. Users do interactive filtering on the trajectories by direct manipulation from spatial and temporal aspects. A suit of circular filters provides flexible spatial constraints. A two-layer temporal filter is designed to set temporal constraints. After filtering, selected trajectories are fed as input to an adaptive DBSCAN clustering algorithm [17]. Trips are grouped into OD clusters automatically. To explore the patterns among OD clusters, three views are proposed to give a multilevel visualization. As an overview, map with convex hulls shows clusters' spatial distribution and line plots display statistics distribution (i.e. trajectory number, travel distance and travel time) over time. In detail end, OD-Wheel visualizes temporal dynamic of each cluster at finer temporal granularity. Interactions on OD-Wheel support users to compare and correlate temporal dynamic among clusters. Users can switch back to modify filtering constraints and then explore iteratively.

4 Trajectory Filter

In this section, we first introduce the model of filtering. Then we introduce the design considerations and corresponding visual design.

4.1 Filter Model

The filter model consists of atomic spatial and temporal queries.

From the spatial aspect, similar to TrajectoryLenses [12], circle is selected as filter shape. More than the three criteria (origin, destination and waypoint) in TrajectoryLenses, criteria of six possible relationships between trajectory and circular filter are provided: *origin*, *destination*, *origin/destination*, *passing*, *inclusive*, and *exclusive*. Moreover, for two or more filters, direction can be assigned between filters to select trajectories following certain direction. Complex filtering can be built by combining atomic queries. For example, users can set multiple filters to select trajectories travelling through several regions.

From the temporal aspect, a two-level temporal filter supports to query from date and time level.

When analysing OD patterns, the filter with *origin* or *destination* option is automatically detected as the central region.

Figure 2: Interactions on Circular Filter: different functions are invoked by hovering on corresponding regions. Direction between filters is assigned by dragging from one to another.

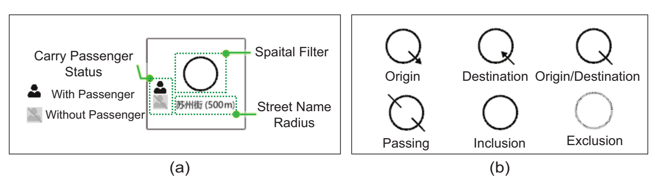

Figure 3: Circular Filter Design: (a) circular filter glyph: carry passenger status, street name, radius and spatial filter with location constrains are explicitly encoded. Gray without-passenger status is invalid here. (b) six types of spatial filters (the left-top circle in glyph) with different location constrains.

4.2 Filter Design

The spatial filter is developed in map-based visualization. There are two design considerations.

Usage Simplicity Free of keyboard and menu, parameter tuning is integrated into the circular filter. as Figure 2(a) shows, when mouse hovering on certain region, different function is waked and corresponding manipulating handle is visible. Specifically, direction between filters is assigned by dragging from one to another.

Semantic Visibility For the ease of parameter perception, parameters are explicitly encoded in circular filter. As shown in Figure 3(a), carry/without passenger status is indicated in left-bottom.

Text below informs the region's center street name and its radius. The right top shows spatial filter with current location constraint (e.g. inclusion constrain here). Figure 3(b) shows filters with different location constraints.

The temporal filter supports filtering date and time (Figure 4(b)). In date part, weekdays are colored in white and weekends are in green. In time part, the granularity is 10 minutes. A date or time range can be defined by dragging and moving, which is colored in orange. Double clicking in blank area cancels the selection.

Figure 4: Date-time Temporal Filter: the oranges indicate those selected time range and the greens represent the weekends.

5 OD Pattern Visual Analysis

After filtering trajectories of a central region, an iterative DBSCAN algorithm [17] is employed to extract O clusters and D clusters. The system provides a multi-level visual analysis on OD patterns.

5.1 Overview

The system provides the spatial and temporal overview of OD clusters. The black frame ring is the circular filter at the central region. Spatially, as Figure 6(a) shows, clusters are represented as convex hulls. O clusters are assigned with warm color list and D clusters are cold, as the color legend shows in the Figure 6(a). To each convex hull, a circle with a number inside is attached. The size of circle encodes the trajectory number and the number inside circle is shows its trajectory number. OD clusters pair close enough to each other indicates a region in bidirectional traffic relationship with the central region. Line plots give the statistical distribution of trajectories. Figure 6(b)(c) display the distribution of average trajectory number on weekdays and at weekends respectively. In Figure 6(d), bar plots show the histograms of average travel distance and travel time. In bar plot, to save the horizontal space, the bar in the end collects extreme values which are larger than a threshold.

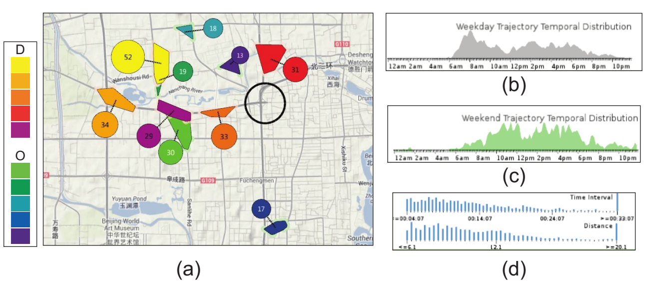

Figure 6: Overview: (a) clsuters' spatial view. (b) average trajectory distribution on weekdays. (c) average trajectory distribution at weekends. (d) travel time and travel distance histograms.

5.2 OD-Wheel

OD-Wheel is designed to explore the temporal dynamic of OD clusters, to help user study the traffic pattern of the central region. In this work, although the temporal dynamic refers to traffic flow volume and travel time, the method can be extended to explore other variations.

5.2.1 Visual Design

The main idea behind OD-Wheel is to warp a part of linear view to circular one. There are several considerations to do warping. Firstly, between clusters, there may be relationship to be visualized. Straight line connecting clusters along circle is much easier to trace than polyline or arc connecting clusters in a linear layout. Secondly, clusters can be moved around the ring flexibly to do comparison with other clusters. Because circular layout has the drawback of distortion, linear view is retained for accurate analysis. Sharing the same time axis makes it intuitive to correlate variations between linear view and circular view.

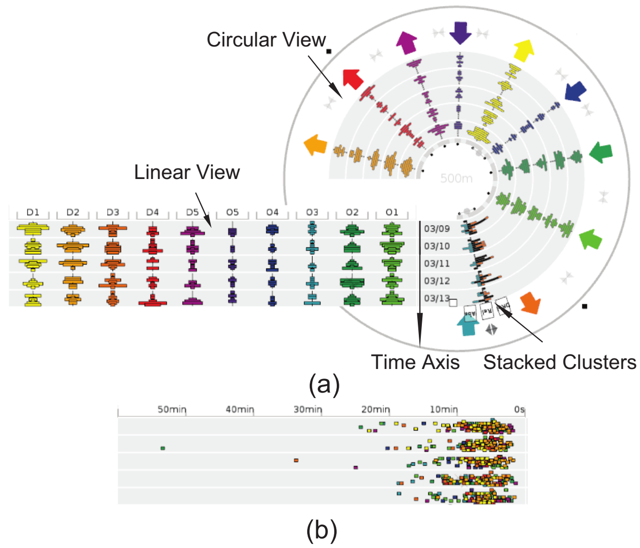

Figure 7: Visual Design of OD-Wheel: (a) linear and circular view displays the temporal distribution of travel flow volumes. (b) an alternative to linear view which plots the distribution of travel time.

Figure 7(a) shows OD-Wheel in detail. Linear view and a circular view share the same time axis, i.e. the outwards radial axis in the circular view and the top-to-bottom vertical axis in linear view. Each OD cluster is visualized as a stack of bars along time axis. The width of bar encodes the temporal dynamic value during corresponding time interval. In Figure 7(a), both circular view and linear view visualize show distribution of traffic flow volume over time. In linear view, the plot can be extended to other variations, for example, Figure 7(b) is the temporal distribution of travel time. In circular view, the inner connection line between OD clusters indicates they belong to a same region, which are smaller than a certain distance threshold (500m), shown in the middle of circle. The arrow glyphs outside the ring distinguish O clusters and D clusters. The inwards arrow represents a O cluster and the outwards arrow represents a D cluster. The gray glyph between clusters is a button which is introduced in Section 5.2.2.

5.2.2 Interaction on OD-Wheel

To explore dynamic among clusters, several interactions are developed for easy comparison. To compare with any cluster, a cluster can be relocated by dragging and moving around ring. Pressing gray button between pair of clusters stacks the two clusters to a common middle line, as the right bottom part in Figure 7(a) shows. Difference between the two clusters along time is explicitly encoded by black bars. When mouse hovering on bars, its time interval pops out. When two clusters match up with each other (i.e. the distance between them is smaller than the distance threshold), (re)clicking on the connecting line between cluster pair brings(recovers) these clusters to(from) neighbours.

In the circular view, dragging mouse upwards or downwards in inner circle can adjust the distance threshold.

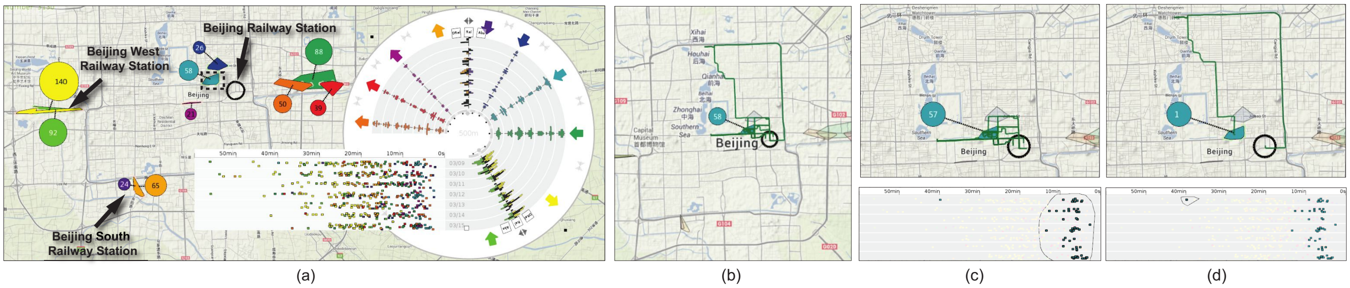

Figure 5: Travel Time Exploration of a Central Region: region around Beijing Railway Station is selected as central region. (a) the top five O and D clusters are plotted on map and OD-Wheel visualizes the temporal dynamics of traffic flow volume and travel time. Its bidirectional traffic flows with Beijing Railway Station and Beijing West Railway Station are stacked, to compare traffic volume in opposite direction; (b) selecting the blue region in dotted box, trajectories travelling are shown; (c, d) travel routes with normal travel time and abnormal travel time are plotted respectively.

To correlate variation between circular view and linear view, they cooperate in brush-and-link manner. Hovering on the arrow glyph in circular view highlights corresponding items belonging to selected cluster in linear view. Selecting some items in linear view, corresponding ones are highlighted in circular view.

6 RESULTS AND DISCUSSION

We set up experiments to explore the OD patterns and give a use study to show how the system works. Then we discuss limitations and extensions of the method.

6.1 Preprocessing and Implemented Details

A quad-tree space division is built to index the trajectories. Filtering is decomposed into three steps: coarse filter according to quad-tree index, related trajectories loading and fine filter according to the setting constraints. To maintain interactive filtering, when dynamic filtering, the first N trajectories (e.g. $N = 100$) are queried and rendered, to ensure the smoothness of interactions. Only when settling down, the whole dataset is queried. We apply the iterative DBSCAN [17] to do OD clustering.

Our system is implemented using C++/QT. The experiments are conducted on an Intel(R) Core(TM)2 2.66GHz Laptop with 4GB RAM and a NVIDIA Geforce GTX 470 GPU. The preprocessing is run off line. After the preprocessing, system supports interactive visual analysis.

6.2 Use Study

With interactive filter and OD-Wheel, our system supports to analyse the temporal patterns within and over OD clusters.

With filtering function, as Figure 5(a) shows, the region around Beijing Railway Station is selected as the central region. Its top five O and D clusters are plotted on the map. With 500m distance threshold, two OD pairs are detected, around Beijing West Railway Station and Beijing South Railway Station respectively. Others are in unidirectional traffic relationship with central region. In circular view, bidirectional traffic flows with the two railway stations are stacked respectively. For both two railway stations, the traffic flow going out from Beijing Railway Station is larger than that coming into Beijing Railway Station. And Beijing Railway Station has more travel volume with Beijing West Railway station than that with Beijing South Railway station. In the linear view, the dot plot displays the distribution of travel time. Overall, we observe that traveling from purple and blue region to central region takes averagely shortest time because of short distance to central region, while the yellow and light green regions take the longest time. On the other hand, the time cost on March 15 spreads more densely than that on other days.

By selecting specific OD region on the map, trajectories between central region and selected cluster are highlighted. Figure 5(b) shows trajectories travelling from the central region to the blue region, which is marked in dotted box in As Figure 5(a). Its distribution of travel time is plotted in Figure 5(c, d). Most of travel cost is around 10min but one outlier is around 40min on March 9. To check whether correlation with travel route, normal time cost dots and abnormal time cost dot are selected and corresponding trajectories are plotted. In Figure 5(c), almost every trajectory goes from blue region to central region straightly. However, in the Figure 5(d), the trajectory take nearly around 40min travel in a long distance.

6.3 Discussion

In the experiment, we apply our method to taxi trajectories, which can be directly applied to other vehicles' trajectories. With filter criteria settings on trajectories, user is able to select OD specifically. Regardless of the filter criteria setting on trajectories, OD-Wheel can be extended to discrete OD data.

As mentioned in Section 6.1, increasing of data volume has little impact on the first fetching but intensifies the latency when querying the whole dataset. For the ODWheel, the number of OD clusters is suggested no more than 5. As the increasing of observed OD clusters, the visual space for one O or D cluster decreases.

Our work has some limitations. Firstly, user may input invalid queries such as two *Origin* filters set at same time. Our system can not detect invalid queries automatically. Current solution is to make circular filter design as intuitive as possible so user can check if the query is invalid or not manually. Second, in our method, we only consider OD clusters and remove trajectories which don't belong to clusters. However, trajectories with low dense distribution may reveal other traffic patterns.

7 CONCLUSION

In this paper, we propose a visual design OD-Wheel to explore a central region's OD patterns. OD-Wheel integrates a linear view and a circular view, which supports to observe, compare and correlate temporal variants. Combining with intuitive filtering and automatic clustering, an interactive visual analysis system supports user to explore traffic pattern and detect abnormality.

ACKNOWLEDGEMENTS

The authors wish to thank Data Tang and OpenStreetMap for providing the data, and the anonymous reviewers for their valuable comments. This work is supported by NSFC No. 61170204 and the National Program on Key Basic Research Project (973 Program) No. 2015CB352503.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer, 2011.

[2] G. Andrienko and N. Andrienko. Spatio-temporal aggregation for visual analysis of movements. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 51–58, 2008.

[3] N. Andrienko and G. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Trans. Vis. Comput. Graph.*, 17(2):205 –219, 2011.

[4] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne. Flowstrates: An approach for visual exploration of temporal origin-destination data. *Comput. Graph. Forum*, 30(3):971–980, 2011.

[5] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2649–2658, 2012.

[6] N. Ferreira, J. Poco, H. Vo, J. Freire, and C. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city cab trips. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2149–2158, 2013.

[7] D. Guo. Flow mapping and multivariate visualization of large spatial interaction data. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1041 –1048, 2009.

[8] 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 *Proceedings of IEEE Pacific Visualization Symposium*, pages 163–170, 2011.

[9] S. Havre, B. Hetzler, and L. Nowell. Themeriver: visualizing theme changes over time. pages 115–123, 2000.

[10] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, Mar. 2004.

[11] D. Holten and J. van Wijk. Force-directed edge bundling for graph visualization. *Comput. Graph. Forum*, 28(3):983–990, 2009.

[12] R. Krüger, D. Thom, M. Wörner, H. Bosch, and T. Ertl. Trajectorylenses - a set-based filtering and exploration technique for long-term trajectory data. *Comput. Graph. Forum*, 32(3):451–460, 2013.

[13] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing. Discovering spatio-temporal causal interactions in traffic data streams. In *Proc. ACM SIGKDD*, pages 1010–1018, 2011.

[14] X. Liu, Y. Hu, S. North, T.-Y. Leea, and H.-W. Shen. Correlatedmultiples: Spatially coherent small multiples with constrained multidimensional scaling. Technical report, Ohio State University, 2013.

[15] M. Lu, Z. Wang, and X. Yuan. Trajrank: Exploring travel behaviour on a route by trajectory ranking. In *Proceedings of IEEE Pacific Visualization Symposium*, 2015.

[16] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. Liverac: Interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1483–1492. ACM, 2008.

[17] G. Pan, G. Qi, Z. Wu, D. Zhang, and S. Li. Land-use classification using taxi gps traces. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):113–123, 2013.

[18] C. Peng, X. Jin, K.-C. Wong, M. Shi, and P. Li. Collective human mobility pattern from taxi trips in urban area. *PLoS ONE*, 7(4):e34487, 2012.

[19] D. Phan, L. Xiao, R. Yeh, and P. Hanrahan. Flow map layout. In *IEEE Symposium on Information Visualization*, pages 219 – 224, 2005.

[20] A. Rae. From spatial interaction data to spatial interaction information? geovisualisation and spatial structures of migration from the 2001 UK census. *Computers, Environment and Urban Systems*, 33(3):161 – 178, 2009.

[21] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1325–1332, 2008.

[22] W. Thompson and S. Lavin. Automatic generation of animated migration maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 33:17–28, 1996.

[23] J. van Wijk and E. van Selow. Cluster and calendar based visualization of time series data. In *IEEE Symposium on Information Visualization*, pages 4 –9, 1999.

[24] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. van de Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2159–2168, 2013.

[25] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, and Q. Wu. Visual exploration of sparse traffic trajectory data. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1813–1822, 2014.

[26] M. Weber, M. Alexa, and W. Muller. Visualizing time-series on spirals. In *IEEE Symposium on Information Visualization*, pages 7 –13, 2001.

[27] J. Wood, J. Dykes, and A. Slingsby. Visualization of origins, destinations and flows with OD maps. *Cartographic Journal*, 47:117–129, 2010.

[28] J. Zhao, F. Chevalier, and R. Balakrishnan. Kronominer: using multi-foci navigation for the visual exploration of time-series data. In *Proc. of ACM SIGCHI 2011*, pages 1737–1746, 2011.