Uncovering the Aggregation Pattern of GPS Trajectory based on Spatiotemporal Clustering and 3D Visualization

C. K. Liu [1,2,3,*], K. Qin [4], K. Chen[4], R. Ma [1,2,3]

[1] Changjiang Survey, Planning, Design and Research Co., Ltd, Wuhan, China – (liuchengkun, marui@cjwsjy.com.cn)
[2] Changjiang Spatial Information Technology Engineering Co., Ltd, Wuhan, China
[3] Water Resources Information Perception and Big Data Engineering Research Center of Hubei Province, Wuhan, China
[4] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China – (qink, xcmkun @whu.edu.cn)

KEY WORDS: GPS trajectory, Spatiotemporal clustering, 3D visualization, Aggregation pattern, Similarity measurement

ABSTRACT:
Constrained by road network structure, travel choice and city function zoning, GPS trajectory data exhibits significant spatiotemporal correlation. Unveiling the clustering and distribution patterns of GPS trajectory can help to better understand the travel behaviour as well as the corresponding spatial and temporal characteristics. This paper proposes an approach to identify and visualize the aggregation pattern from GPS trajectory data. Firstly, slow feature trajectory sequences are extracted from raw taxi trajectory data. Together with taxi states information, these sequences are processed as shorter length tracks for faster discovery of cluster similarity. Thereafter, the temporal and spatial similarity and dissimilarity metrics between the trajectories are established, and the temporal and spatial distances between the trajectories are defined to form a space-time cylinder model. Next, based on the idea of density clustering, the DBSCAN spatiotemporal expansion of trajectory data is proposed. Feature trajectory sequences are then clustered into groups with high similarity. Finally, for a more intuitive understanding of the trajectory aggregate distribution, time dimension info of each point in the sequences is used as Z axis, thus the sequences are stretched on the map in different colour for 3D visualization. The proposed method is validated by a case study of taxi trajectory data analysis in Wuhan City, China.

1. INTRODUCTION

With the rapid development of positioning and communication technologies, increasing amount of traffic information has been collected in diverse ways. As a result, traffic patterns, especially aggregation patterns, can be more accurately identified to find hot spot, congestion, accident and all other types of sudden gathering behaviour. Uncovering the aggregation patterns behind them can help increase safety, optimize urban planning and providing guidance for traveling route selection.

Among existing traffic data collection methods, floating car is a reliable and cost-effective tool for collecting traffic data (Fabritiis et al., 2008). To generate traffic information for every road segment, speed and position information of floating cars is usually collected at regular time intervals (e.g., 30s). With attached information about the states of passenger and car's engine, floating car data enables us to identify feature sub-trajectory of specific meanings like getting-on/drop-off hotspot (Zhao et al., 2015) and traffic congestion (Liu et al., 2015). In this paper, we focus on slow sub-trajectory sequences, these shorter length tracks can help faster discover similarity and find clusters.

To obtain the aggregation pattern, this study uses the idea of the space-time cylinder model (Birant et al., 2007), proposes the temporal distance and spatial distance measurement methods for trajectory, extends the DBSCAN (Density-based spatial clustering of applications with noise) algorithm for spatiotemporal clustering, and visualizes the cluster results in 3D view stretched by time dimension.

The remainder of this paper is organized as follows. Section 2 presents related work about trajectory similarity measurement and spatiotemporal clustering. Section 3 describes the proposed spatiotemporal clustering and 3D visualization method to extract aggregation patterns from GPS trajectories. Experimental results are reported in section 4 as a validation of the proposed approach. Section 5 concludes the paper and indicates the future works.

2. RELATED WORK

GPS trajectory is a simple and efficient way to describe the moving object by tracking the object's position from one frame to the next. And the analysis of trajectory has long been a research focus in diverse fields of study. In the context of intelligent surveillance systems (ITS) (Tian et al., 2017), trajectory clustering is a crucial technology in many applications including activity analysis (Morris et al., 2011), path modelling (Zhang, et al., 2009), anomaly detection (Dee etal., 2008), and traffic monitoring (Aköz et al., 2014).

Trajectory clustering is the process of extracting the similarity, anomaly and valuable patterns from the trajectory data, and it has been widely used in research and engineering. Most existing trajectory analysis methods can be categorized into similarity-based models and Probabilistic Topic Models (PTM) (Arfa et, al., 2019).

The main steps of similarity-based approaches are similarity-matrix calculation and similarity-matrix based trajectory clustering. In the first step, similarities between each two sequeces are obtained via a similarity function and then stored into a $N*N$ matrix, where N is the total number of available trajectories. Well-known similarity measurement methods for trajectory analysis include Euclidean distance, dynamic time wrapping (DTW) (Keogh, et al., 2000), Hausdorff distance (Atev, et al., 2010), and Longest Common Sub-Sequences (LCSS) (Vlachos, et al., 2002). After that, a standard clustering algorithm is adopted to cluster the trajectories into K clusters based on their similarities. Typical clustering algorithms include spectral clustering (Porikli, 2004), agglomerative clustering (Day, et al., 1984), and fuzzy c-means (Wei, et al., 2006). The main disadvantage of this approaches lays on that it requires the number of clusters to be known in advance.

To solve the problem of unknown number of clusters, the subsequent part of this article will expand the density clustering method from the perspective of time and space.

3. METHOD OF TRAJECTORY CLUSTERING AND 3D VISUALIZATION

3.1 Density based space-time cylinder model

To obtain clusters from data set with unknown number of categories, traditional methods generally use density clustering. A typical method for this purpose is DBSCAN, density-based spatial clustering of applications with noise. In this algorithm, if the distance d between two points is less than d_0, then they are density reachable to form a cluster; if a point is not density reachable to any other clusters, then it is treated as noise.

When the sample points carry with not only space properties, but also time properties, researchers proposed ST-DBSCAN method to extend the space-time distance calculation. A widely accepted approach is the space-time cylinder model, as shown in figure 1. As can be seen from the figure, the cylinder is centred on the space-time object point, d_0 is the radius of the circle, and $2T_0$ is the height of the cylinder.

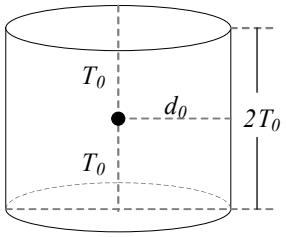


Figure 1. Space-time cylinder model for spatiotemporal density-connected

In this model, each point forms a spatiotemporal neighbourhood based on time and space thresholds, if the time distance and spatial distance of two points meet the threshold condition at the same time, then they are considered to be spatiotemporal reachable.

Inspired by this method, for the trajectory data, in order to calculate the space-time density reachability between trajectory sequences, the trajectory sequence can be abstracted into an object, and use the space-time cylinder model to build trajectory spatiotemporal neighbourhood. From this perspective, the subsequent calculations only need to define the temporal and spatial distance between two trajectory sequences.

3.2 Similarity measurement on time and space

GPS trajectory is space-time sequence made of a set of points with position and time information. Trajectory sequence *Traj* is represented as:

$$Traj = \{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\} \quad (1)$$

where x_i, y_i = GPS point coordinates
t_i = GPS point collected time

Considering that the time information of the trajectory sequence is simplified to $[t_1, t_n]$, this paper proposes a method for calculating temporal similarity S and Dissimilarity D based on time overlapping in the time dimension.

For two trajectories of different time spans, it can be divided into three cases as shown in Table 1, which are two trajectories do not overlap in the time dimension, partially overlap, and one trajectory is a subinterval of another one.

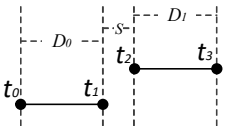
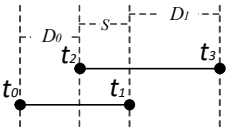
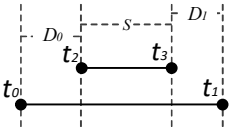
ID	Description	Schematic diagram	Similarity and Dissimilarity Metrics
(1)	two trajectories do not overlap in the time dimension		$S = t_1 - t_2 < 0$ $D_0 = t_1 - t_0$ $D_1 = t_3 - t_2$
(2)	two trajectories partial overlap in the time dimension		$S = t_1 - t_2 > 0$ $D_0 = t_2 - t_0$ $D_1 = t_3 - t_1$
(3)	one trajectory is a subinterval of another one in the time dimension		$S = t_3 - t_2 > 0$ $D_0 = t_2 - t_0$ $D_1 = t_1 - t_3$

Table 1. Temporal distance description method for spatiotemporal trajectory

After the three proposed similarity and dissimilarity indicators S, D_0 and D_1 are calculated, normalize it, thus the temporal distance *T-Dist* between two trajectories is defined as follow:

$$T\text{-}Dist = b \cdot (D_0 + D_1)/(T_1 + T_2)) - a \cdot (S / \min(T_1, T_2)) + c \quad (2)$$

where T_i = time duration of the sequences
a = similarity coefficient value
b = dissimilarity coefficient value
c = adjustment coefficient value

Considering that the similarity coefficient is more important to form a cluster, set the value $a = 1$, $b = 0.5$, $c = 1$, then the value range of *T-Dist* is $[0, +\infty)$. When two timespans are completely equal, the minimum value 0 is obtained; when the time-spans of the two sequences are completely different, the farther the time separates, the larger the value of *T-Dist* will be.

As for spatial distance, consider two trajectory sequences for P and Q respectively, the spatial information of them are defined as follow:

$$P = \{p_1, p_2, p_3, p_4\}$$
$$Q = \{q_1, q_2, q_3, q_4, q_5\} \quad (3)$$

where p_i, q_i = GPS point coordinates

Due to the sampling interval of the trajectory, the path between the two points in a certain trajectory cannot be accurately obtained. Considering the short interval of the sampling points, this paper uses a straight line to estimate the intermediate path between the two points. Then the shortest distance from each point p_i in sequence P to sequence Q as well as each point q_i in

sequence Q to sequence P can be calculated as shown in Table 2.

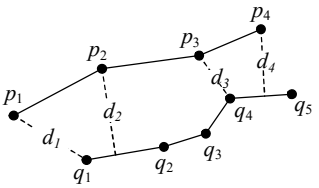
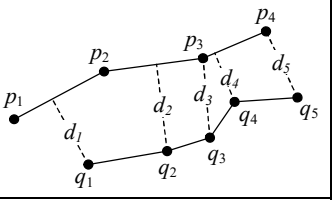
ID	Description	Schematic diagram
(1)	min distance from each trajectory point in the sequence P to Q	
(2)	min distance from each trajectory point in the sequence Q to P	

Table 2. Spatial distance description method for spatiotemporal trajectory

Based on the spatial distance series in Table 2, the similarity and dissimilarity can be measured by the similar trajectory specific gravity and the average distance. Assuming that the distance threshold is d_0, taking the spatial distance calculation from trajectory P to Q as an example, the metrics S_P and D_P are defined as shown in the following equations.

$$S_P = \sum_{i=1}^{n} (d_i < d_0) / n \tag{4}$$

$$D_p = \sum_{i=1}^{n} d_i / n \tag{5}$$

where S_P = similarity metric from trajectory P to Q
D_P = dissimilarity metric from trajectory P to Q
d_i = distance from point pi in sequence P to Q
d_0 = threshold as the distance is close
n = number of points in sequence P

The calculated metric S_P is used to measure the proportion of the trajectory points that satisfy the distance threshold to the entire trajectory, while D_P is used to measure the degree of offset from one trajectory to another.

Like the calculation of the temporal distance, after the spatial similarity and dissimilarity description between the trajectories are obtained, it needs to be normalized, thus the spatial distance between the two trajectories is obtained as:

$$S\text{-}Dist = b \cdot (D_P + D_Q)/2d_0 - a \cdot (S_P + S_Q)/2 + c \tag{6}$$

where a = similarity coefficient value
b = dissimilarity coefficient value
c = adjustment coefficient value

Similarly, considering that similarity is more important, set the value a = 1, b = 0.5, c = 1, Then the value range of *S-Dist* is [0, +∞). When trajectories P and Q are completely coincident, D = 0, S = 1, then *S-Dist* = 0; and when the two trajectories are far apart in space, the D value becomes larger, the S value becomes smaller, and the *S-Dist* value gradually increases.

3.3 Trajectory clustering algorithm expanded on DBSCAN

After defining the similarity measurement methods on time and space, the density-based algorithm DBSCAN is adopted for spatiotemporal expansion. Considering the DBSCAN algorithm as the benchmark algorithm, the temporal and spatial distances are then applied to the distance metrics, and the definition of the ε neighbourhood is extended to obtain the spatiotemporal neighbourhood of the trajectory sequences for space-time density reachable definition.

The proposed spatiotemporal clustering algorithm is described in Figure 2.

Algorithm: Trajectory spatiotemporal clustering extended from DBSCAN

Input:
Trajs = { $Traj_0$, $Traj_1$, …, $Traj_n$} —— trajectory collection
MinPts —— minimum count to from a core trajectory P
d_0 —— the distance threshold from one sample point in trajectory P to Q
S-Dist —— spatial distance threshold
T-Dist —— temporal distance threshold

Output: trajectory clusters

Algorithmic Process:
1. mark all trajectory as *unvisited*
2. randomly select an *unvisited* trajectory P and mark it as *visited*
3. calculate the temporal and spatial distance between P and other trajectories
4. If at least *MinPts* trajectories are in P's spatiotemporal neighbourhood (spatial distance < *S-Dist*, temporal distance < *T-Dist*)
5. create new cluster C, put trajectory P into C
6. randomly select an *unvisited* trajectory Q, judge the spatiotemporal neighbourhood with P's
7. merge or generate new clusters
8. output trajectory cluster C
9. Else mark P as *noise*
10. Until no *unvisited* trajectory left

Figure 2. Trajectory spatiotemporal clustering algorithm extended from DBSCAN

3.4 3D Visualization method for clustering results

Aggregate regions can be found from spatiotemporal clustering results by overlay analysis, however, this traditional two-dimensional way can only show the superposition distribution of the aggregated area in space, which cannot truly reflect the spatial and temporal distribution of the aggregated area. Therefore, according to the characteristics of the GPS trajectory, the time point of the trajectory is visualized as a Z value in a time period.

For a clearer explanation of the proposed 3D visualization method, Figure 3 takes a typical traffic intersection as an example, and separately gives the clustering results and noise in traditional 2D view and the proposed 3D view.

There are four case trajectory sequences shown in this figure, which are the red solid line a, the grey dotted line b, the blue solid line c, and the yellow dotted line d. The trajectory line a and c are in the same cluster, while b and d are treated as noise.

In the left 2D view, the noise line *b* is very similar to trajectory *a* and *c*. But in the right 3D view, we can clearly notice that the line *b* has the totally different time-span from the other sequences.

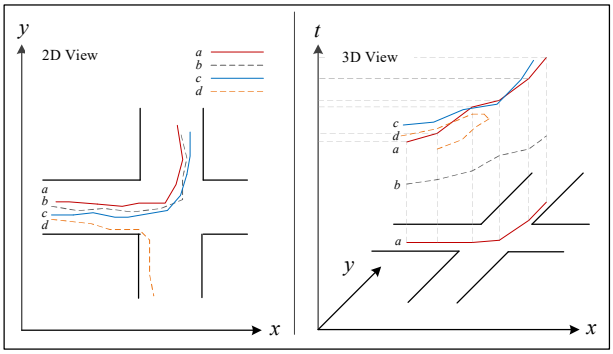


Figure 3. Trajectory clustering results visualization in traditional 2D view and proposed 3D view

4. EXPERIMENTAL RESULTS

The experiment uses 12,000 taxis' trajectories collected on May 1st (holiday) and May 12th (weekday) 2014 in Wuhan, a large city in China. The 12,000 taxis transmitted their location and speed data every 60 second with service status, and generated over 14 million trajectory records every day. With attached information about the states of passenger and car's engine, we extract 21,416 slow trajectories on weekday and 34,156 slow trajectories on holiday.

Figure 4 gives an overview about the distribution of the extracted slow trajectories on May 1st. The trajectory color in this figure indicates the average speed rate. The redder the color, the slower the rate.

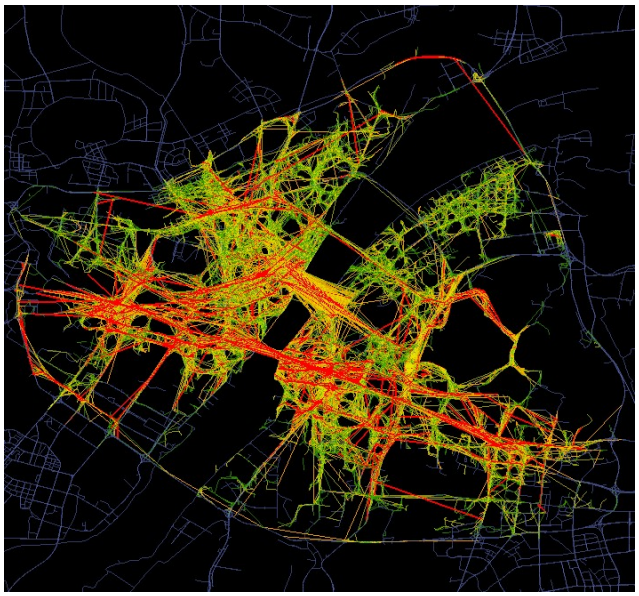
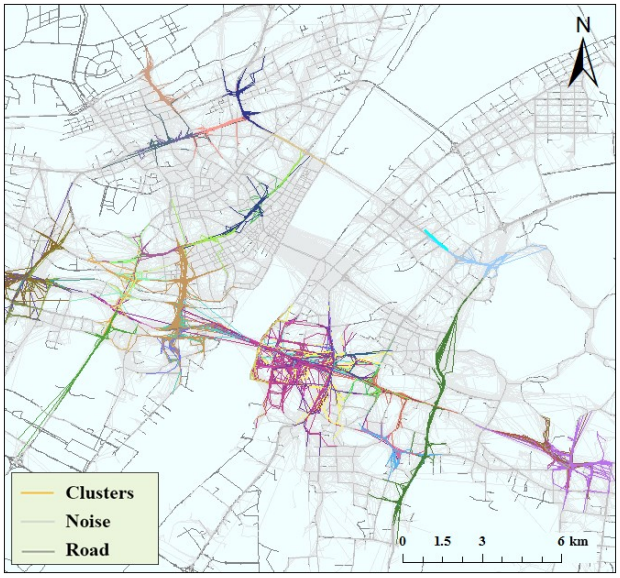


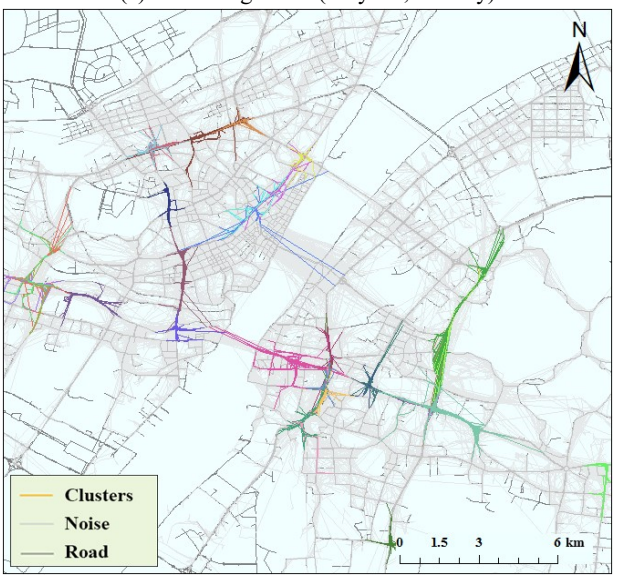
Figure 4. Slow trajectories overview on May 1st

To extract clusters from the slow trajectories, the parameter selection is strengthened to limit the distance of the track space, and the constraint on the time distance is relaxed. In the time measurement, considering that two trajectories with similar timespans require one third time coincidence as $S = 1/3$, and the dissimilarity $D_P + D_Q = 3/4$, then the time distance threshold *T-Dist* = 1. While in the space measurement, assuming that when one trajectory point to another trajectory's distance d_0 is less than 100 meters, they are considered as close. At the same time, demanding that the number of tracks satisfying the d_0 condition to be at least 2/3, and the average distance of all points in the sequences does not exceed 2/3 d_0, then the spatial distance threshold *S-Dist* after normalization is 2/3.

Additionally, Set the number of slow trajectories that make up the cluster to at least 40, then the clusters are extracted using the proposed method. Clustering results in 2D view are shown in Figure 5.



(a) Clustering result (May 1st, holiday)

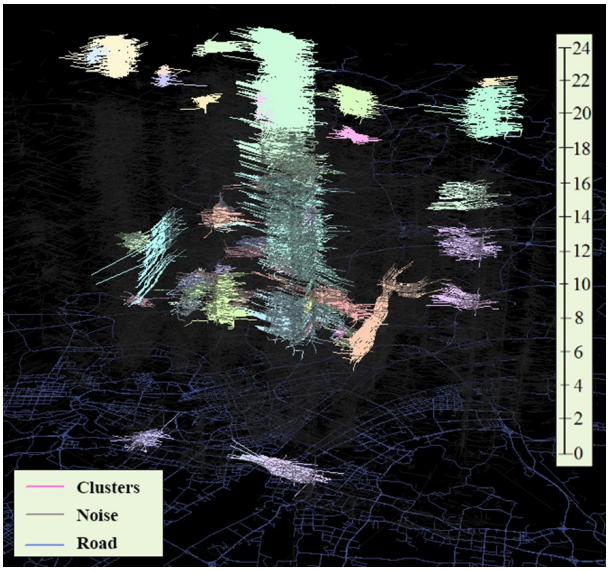


(b) Clustering result (May 12th, weekday)

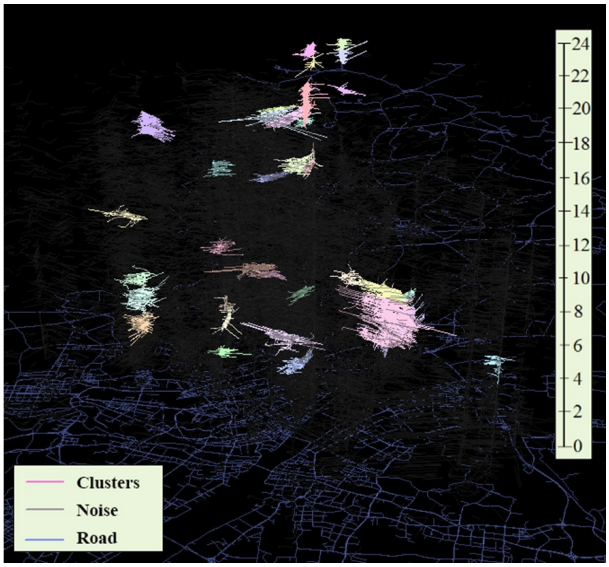
Figure 5 Trajectory clustering results in 2D view

In this 2D view, trajectory clusters are shown in different colors, forms specific aggregation area, while the grey ones indicate the noise which are not evidently aggregated. These aggregated trajectories imply some specific traffic information like congestion, accident, road maintenance, etc. The

From the 2D clusters distribution, we can see some difference in trajectory position and quantity, but cannot understand the time dependent patterns. For better recognition of the spatiotemporal patterns, the time dimension is stretched to 24 hours timespan in 3D view as shown in Figure 6. From this figure, we can clearly tell the cluster distribution in time and space, and know the degrees of aggregation by timespans and quantity.



(a) Clustering result (May 1st, holiday)



(b) Clustering result (May 12th, weekday)

Figure 6 Trajectory clustering results in 3D view

5. CONCLUSIONS

In this paper, a method is proposed to identify and visualize the aggregation pattern from spatiotemporal trajectories. This method draws on the idea of space-time cylinder model and DBSCAN algorithm, defines the temporal and spatial distance metrics for similarity measurement, expands the density-based clustering algorithm in spatiotemporal trajectory, and stretches the time dimension of the trajectory sequences for 3D visualization. The experiment selects taxi GPS trajectory in Wuhan as the main data source, verifies the accuracy and validity of this proposed method using dataset on working days and holidays.

However, a couple of difficulties still exists in the parameter optimization of similarity measurement and clustering accuracy of long-distance sequences. In the future research, we would like to focus on these problems, try to give more reasonable parameters in similarity measurement, and to strive to solve the long trajectory partially similar problem.

ACKNOWLEDGEMENTS

We would like to thank the constructive comments from the anonymous referees, and we appreciate the financial supports from the National Key R&D Program of China (No. 2017YFC1501206 and 2017YFC1502601).

REFERENCES

Fabritiis C., Ragona R., Valenti G., 2008: Traffic estimation and prediction based on real time floating car data. *The 11th IEEE International Conference on Intelligent Transportation Systems*, 197-203.

Zhao P. X., Qin K., Zhou Q., Liu C. K., Chen Y. X., 2015: Detecting hotspots from taxi trajectory data using spatial cluster analysis. *The 4th International Workshop on Spatiotemporal Computing*, 131-135.

Liu C. K., Qin K., Kang C. G., 2015: Exploring time-dependent traffic congestion patterns from taxi trajectory data., *The 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services*, 41-46.

Birant D., Kut A., 2007: ST-DBSCAN: An algorithm for clustering spatiotemporal data. *Data & Knowledge Engineering*, 60(1), 208-221.

Tian B., Morris B. T., Tang M., Liu Y., Yao Y., Gou C., Shen D., Tang S., 2017: Hierarchical and networked vehicle surveillance in its: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 18, 25-48.

Morris B.T., Trivedi M. M., 2011: Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 2287-2301

Zhang T., Lu H., Li S. Z., 2009: Learning semantic scene models by object classification and trajectory clustering. *2009 IEEE international conference on computer vision and pattern recognition*, 1940-1947.

Dee H. M., Velastin S. A., 2008: How close are we to solving the problem of automated visual surveillance. *Machine Vision and Applications*, 19, 329-343.

Aköz O., Karsligil M., 2014: Traffic event classification at intersections based on the severity of abnormality. *Machine Vision and Applications*, 25, 613-632.

Arfa R., Yusof R., Shabanzadeh P., 2019: Novel trajectory clustering method based on distance dependent Chinese restaurant process. *PeerJ Computer Science*, 5, e206.

Keogh E., Pazzani M. J., 2000: Scaling up dynamic time warping for data mining applications. *Proceedings of the 6th*

ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM, 285-289.

Atev S., Miller G., Papanikolopoulos N. P., 2010: Clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems* 11, 647-657.

Vlachos M., Kollios G., Gunopulos D., 2002: Discovering similar multidimensional trajectories. *The 18th IEEE international conference on data engineering*, 673-684.

Porikli F., 2004: Learning object trajectory patterns by spectral clustering. *IEEE International Conference on Multimedia & Expo*, 154-159.

Day W. H. E., Edelsbrunner H., 1984: Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1, 7-24.

Weiming H., Xuejuan X., Zhouyu F., Xie D., Tieniu T., Maybank S., 2006: A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 1450-1464.