A Vehicle Trajectory Analysis Approach Based on the Rigid Constraints of Object in 3-D Space

Wen Jiang, Zhang Zhaoyang(✉), Song Huansheng, and Pang Fenglan

School of Information Engineering, Chang'an University, Xi'an 710064, China
{2014224015, zhaoyang_zh, 2014124037}@chd.edu.cn,
1192750414@qq.com

Abstract. A reliable and effective trajectories similarity metric is one of key factors for vehicle trajectories clustering problem. A trajectory clustering algorithm based on the rigid constraints of vehicles in 3-D space is proposed in this paper, which conducts vehicle trajectories clustering effectively and precisely by using a new 3-D trajectories similarity metric. Based on two key procedures, camera calibration and a reconstruction of 2-D trajectories in 3-D space, a valuable principle that the heights of the trajectories have a linear relationship between them is found through using the kinematic properties of vehicle rigid body in moving. A more valuable information need to be pay attention is that the height of two trajectories that with displacement difference satisfies a plane surface character in 3-D space when conducts a height enumeration. The experimental results show that the trajectories are very stable and reliable for clustering and event detection when reconstructing their relative position in 3-D world coordinate system.

Keywords: Trajectory clustering · Rigid constraints · Camera calibration · 3-D reconstruction

1 Introduction

Trajectory clustering is an important topic in video analysis, the purpose of which is to assign a common cluster label to individual trajectories, and it is widely used in activity surveillance, traffic flow estimation and emergency response [1, 2].

A trajectory is typically defined as a data sequence, which is consisting of several concatenated state vectors from tracking, and that means it is an indexed sequence of positions and velocities in a given time window [3]. Trajectory data can provide effective spatial-temporal information of objects for event analysis [4]. Clustering is one of the most effective approaches for trajectory data analysis [5]. It has been studied for several decades, and many methods, such as k-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), are proposed for trajectory data analysis. For trajectory based event analysis, a key and important step is defining a similarity measurement for two trajectories with different lengths, which has a greater impact on the result. The reason is that length difference induced by the kinematic properties of moving objects makes assessing similarity between two or more trajectories exceedingly difficult [4]. To address the problem of similarity measurement

© Springer Nature Singapore Pte Ltd. 2016
T. Tan et al. (Eds.): CCPR 2016, Part I, CCIS 662, pp. 42–52, 2016.
DOI: 10.1007/978-981-10-3002-4_4

between trajectories, Fu et al. [6] proposed a method to resample trajectories with different lengths to ones with equal lengths. Then the similarity between two trajectories can be computed by the distances between corresponding points on two trajectories. Stefan Atev et al. [7] proposed a trajectory-similarity measure based on the Hausdorff distance, and a modification strategy is given to improve its robustness through accounting for the fact that trajectories are ordered collections of points. Abraham et al. [8] deal the length difference by applying a spatial-temporal similarity measure with given Points of Interest (POI) and Time of Interest (TOI), in which the spatial similarity is treated as a combination of structural and sequence similarities and evaluated by using the techniques of dynamic programming. Thus the similarity set is formed and is used to broadcast trigger-based messages to vehicles in a neighborhood area for future route and information-sharing activities. Hao et al. [9] proposed a length scale directive Hausdorff (LSD-Hausdorff) trajectory similarity measure. Wang et al. [4] proposed a novel descriptor named trajectory kinematics descriptor to represent trajectories based on the kinematic properties from the point-of-view of Frenet-Serret frames. Madhuri Debnath et al. [10] proposed a new framework to cluster sub-trajectories based on a combination of their spatial and non-spatial features. This algorithm combines techniques from grid based approaches, spatial geometry and string processing. Yang Fan et al. [11] proposed a novel 3D visualization tool, in which the Agglomerative Information Bottleneck (AIB) based clustering scheme is illustrated comprehensively, to help users understand the clustering approach vividly and clearly.

The advantage of the methods discussed above is that the similarity between two trajectories with different length can be calculated directly and easily. This is due to some preprocessing steps, such as shortening trajectories with longer length or stretching trajectories with shorter lengths to the ones with same length, formulating trajectories by parameter models to perform the similarity measurement. In other words, the original trajectories are modified and only the position information of the trajectories is used for similarity calculation in these methods. However, the kinematic property of trajectories, which is a typical feature for moving objects, is still neglected [4].

To further improve the robustness of trajectory clustering and to obtain an optimal cluster number, how to define a proper and effective distance measurement for trajectories is still a key and challenging work for trajectory clustering schemes.

The emphasis in this paper is on obtaining a complete trajectories information, which includes points and kinematic property, through a test method of height enumeration. Two properties, the heights of the trajectories satisfy linear properties and two trajectories with displacement difference satisfies a plane surface character in 3-D space when conduct a height enumeration, are found. Based on these properties, the 3-D information of trajectories in 2-D image can be reconstructed completely, and a stable and reliable data for trajectories clustering and events detecting is established. The rest of this paper is organized as follows: Sect. 2 briefly introduces the camera calibration. Section 3 introduces the 3-D reconstruction of 2-D trajectories. Section 4 shows the procedures of trajectories clustering. In the last section, the experimental results and conclusions are presented.

2 Camera Calibration

Camera calibration can help to deal with perspective distortion of object appearance on 2-D image plane which induces a very difficult problem for most feature based 2-D image processing methods. A calibrated camera makes it possible to recover discriminant metrics robust to scene or view angle changes, and it is greatly helpful for some applications like classification and tracking among multi-cameras. Additionally, with camera calibrated, we can use of prior information of 3-D models to estimate real 3-D poses of objects in videos and make object detection or tracking more robust to noise and occlusions [12].

Assume that $P_W(x_W, y_W, z_W)$ is a 3-D coordinate point on target in the 3-D world coordinates system, $P_C(x_C, y_C, z_C)$ and $p_I(u_I, v_I)$ are its corresponding points in the camera coordinates system and the imaging plane respectively. Then the relationship between the world coordinate system and the camera coordinate system can be formulated as

$$\begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix} \tag{1}$$

where R is a rotation matrix with size 3×3, and can be marked as $R \triangleq \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$.

Based on Eq. (1), a transformation from world coordinate to image plane can be formulated as

$$z_C \begin{bmatrix} u_I \\ v_I \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} x_W \\ y_W \\ z_W \\ 1 \end{bmatrix} \tag{2}$$

where K is the camera internal parameters with a matrix form $K = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$,

T is the translation parameters and can be marked as a 3×1 matrix with a form $T = [\, t_1 \quad t_2 \quad t_3 \,]^{\mathrm{T}}$, R and T compose the external parameter matrix of a camera.

The Eqs. (1) and (2) represent the model of a given camera, and the internal parameters and external parameters can be calculated accurately using the vanishing points based recovery method [12]. Then the camera calibration is done and the relationship between world coordinate and image plane is established exactly based on (1) and (2). Figure 1 shows a calibration results for a given scene, the original point of world coordinate is set as the perpendicular foot point of the camera, the direction perpendicular to road plane (blue), the vanishing direction along the road direction

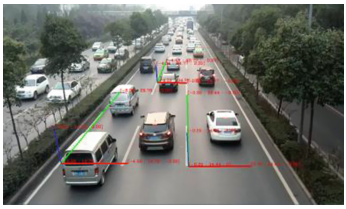


Fig. 1. The result of camera calibration (Color figure online)

(green) and the vanishing direction orthogonal to the road direction (red) are set as z-direction, x-direction and y-direction of the world coordinate respectively.

3 3-D Reconstruction of the 2-D Trajectories

Trajectories, providing spatial-temporal information, are the best feature to describe the kinematic properties of foreground objects. A trajectory is typically defined as a data sequence, which is consisting of several concatenated state vectors from tracking, and that means it is a indexed sequence of positions and velocities in a given time window. The drawback of trajectories is each node belongs to it is described by a 2-D kinematic model. Generally, a trajectory can be represented as a point data set as follows:

$$Tra_{2D}(m,t,t-k) = \{p_I(m,t), p_I(m,t-1), \cdots, p_I(m,t-(k-1))\} \tag{3}$$

where m is the number of a given trajectory in trajectory data set, k is the number of nodes on the given trajectory, $p_I(m,t)$ and $p_I(m,t-k)$ is the image coordinate of trajectory node in the current frame and the previous k frame, respectively.

In order to obtain complete 3-D information of a trajectory, the height information is necessary. However, the real height information is hard to get directly from 2-D trajectory in image plane. Fortunately, for a rigid moving object, the points on its surface have a same property, the nature of synchronous motion. Thus, we can get trajectories information under 3-D description, as follows:

$$Tra_{3D}(m,t,t-k) = \{P_W(m,t), P_W(m,t-1), \cdots, P_W(m,t-(k-1))\} \tag{4}$$

and two trajectories belong to a same rigid moving object body, with sequence number m and n, meet two constraints as their specific properties, as follows:

$$DisDif(m,n,t-i) = Dis(Tra_{3D}(m,t,t-i)) - Dis(Tra_{3D}(n,t,t-i)) = 0 \tag{5}$$

and

$$\begin{aligned} &Dis(Tra_{3D}(m,t,t-i)) = |P_W(m,t)P_W(m,t-i)| \\ &= \sqrt{(x_W(m,t)-x_W(m,t-i))^2 + (y_W(m,t)-y_W(m,t-i))^2 + (z_W(m,t)-z_W(m,t-i))^2} \end{aligned} \tag{6}$$

where $P_W(m,t)$ and $P_W(m,t-i)$ is the 3-D world coordinate point of trajectory node in the current frame and the previous i frame, $|P_W(m,t)P_W(m,t-i)|$ is their distance in 3-D space, and $DisDif(m,n,t,i)$ is the displacement difference between two trajectories, $Dis(Tra_{3D}(m,t,t-i))$ is the Euclidean distance between two points on the current frame and the previous i frames for the trajectory with sequence number m in 3-D world coordinate system, x_W,y_W and z_W are the 3-D coordinate value of a node.

Based on the Eq. (2), if the world coordinates of a point on target, $P_W(x_W, y_W, z_W)$, is given, we can get its image coordinates $p_I(u_I, v_I)$ directly, and the relationship between them can be expressed as a transformation as follows:

$$p_I = f \bullet P_W, \tag{7}$$

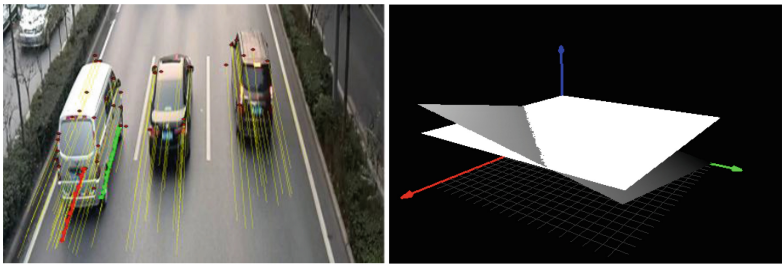
where f can be calculated by Eqs. (1) and (2). Furthermore, if the height of a point on the image plane is given, we can get its 3-D world coordinates using the relationship as follows:

$$P_W = F^{-1} \bullet (p_I \oplus h) \tag{8}$$

Where F^{-1} is the inverse transformation implied in (2). That means, we can calculate the corresponding 3-D coordinates of a point in 2-D image plane when give its height in 3-D space. However, the real height value of a point in 2-D image is unknown. Fortunately, the height of an object is a bounded value. Thus, for two trajectories belong to the same rigid moving object, we can conduct a test using the method of height numeration to estimate the real value of this height. In this test, we can calculate a displacement difference between two trajectories under two supposed trajectory heights. When enumerating all possible combination of two heights, a series of displacement difference can be obtained using the following formulation:

$$\begin{cases} Trajectory\, m : h_m = 0, 1, 2, \cdots, 199cm \\ Trajectory\, n : h_n = 0, 1, 2, \cdots, 199cm \\ DisDiff(m,n,t,i)|_{h_m,h_n} = Dis(Tra_{3D}(m,t,t-i))|_{h_m} - Dis(Tra_{3D}(m,t,t-i))|_{h_n} \end{cases} \tag{9}$$

where the height enumeration value is choose between 0 cm and 199 cm due to the fact that the height of vehicle is smaller than 2 meters in most of monitoring scenes. Figure 2(a) shows the trajectory set in 2-D image and Fig. 2(b) shows the displacement difference between two trajectories (the red and green trajectory in Fig. 2(a)) that belong to a same rigid object under height enumeration. The white plane is a reference plane, which is given by the property of Eq. (5), and represents the displacement difference, $DisDif(m,n,t,t-i)$, is zero. The gray oblique plane is a calculated data plane using (9), which represents the displacement difference between two trajectories under height enumeration. In Fig. 2(b), a point on the gray plane means a value of displacement difference between the red and green trajectories in Fig. 1, which is calculated by giving the two trajectories a height values respectively. The red axis and green axis represent the height of red trajectory and green trajectory in Fig. 2(a), and the blue axis represents the displacement difference value. Figure 2 shows that if the



(a) (b)

Fig. 2. Trajectories and its displacement difference under height enumeration. (a) trajectories in 2-D image; (b) displacement difference under height enumeration for two trajectories(red one and green one in (a)) (Color figure online)

trajectories belong to a same rigid object, the heights of which satisfy a linear relationship

$$A \cdot h_m + B \cdot h_n + C = 0 \tag{10}$$

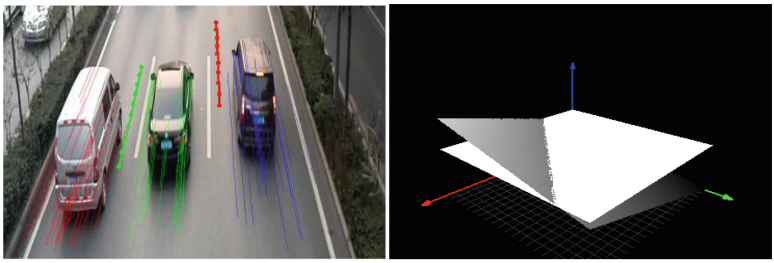
which is the intersecting line of the white plane and the gray plane, and the displacement difference between two trajectories satisfy

$$DisDif(m, n, t, t - i) = A \cdot h_m + B \cdot h_n + C \tag{11}$$

and show in the gray plane.

To illustrate the correctness of Eqs. (10) and (11) as a quantitative way, we set two supposed trajectories in the traffic sense that Fig. 2(a) shows. Based on calibrated camera data, for the two supposed trajectory, their coordinates of beginning point in 3-D space are (−4.2, 12.8, 1.0) and (0.4, 13.2, 2.0) respectively. The speed of the two trajectories is 50 km/H as the vehicle going forward, and the video frame rate is 25 frames per second (fps). The two supposed trajectories are shown in Fig. 3(a) and their displacement difference under height enumeration is shown in Fig. 3(b), in which their height satisfy the linear relationship −0.317Hg + 0.37 h − 0.423 = 0, where Hg and Hr is the height of first trajectory (the green one) and the second trajectory (the red one), respectively. It is worth noting that, for trajectory difference metric, Eq. (5), in 3-D space, the coordinates in x-axis and y-axis has no influence for the result. This means that trajectory Eq. (10) only depends on the relative height difference of two given trajectories when the speed is unchanged.

Actually, the real height for any trajectory in 2-D image is unknown. Hence the heights of trajectories cannot be calculated based on Eq. (10). Fortunately, the similarity measurement of two trajectories can be solved effectively as long as the relative height of them is given. That means if a reference trajectory can be found, the height of other trajectories can be calculated. In this paper, the heights of all the trajectories extracted from the 2-D image are set to zero, the corresponding trajectories in 3-D

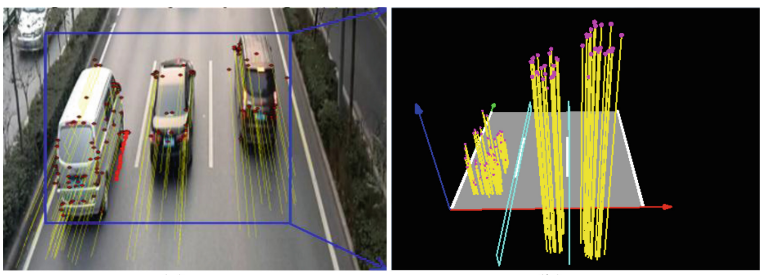


(a) (b)

Fig. 3. Two supposed ideal trajectories and its displacement difference under height enumeration. (a) Two supposed ideal trajectories in 2-D image; (b) displacement difference under height enumeration for two supposed ideal trajectories (the red on and the green one) (Color figure online)

space can constructed and each trajectory has a calculated moving distance. The trajectory that with a minimum moving distance has a relative small height value in 3-D space, thus the height of this trajectory can be set to zero and the heights of the other trajectories will be calculated exactly under the zero-value height trajectory assumption. Figure 4 shows the trajectories and their relative height calculated under zero-value assumption of the reference trajectory. In Fig. 4(a), the red trajectory is the one has the smallest height. Figure 4(b) shows the relative height of the other trajectories by setting the height of red trajectory in Fig. 4(a) to zero.

The Fig. 4 shows a valuable property that a speed difference of two trajectories belongs to two different vehicles induced a height difference under the rigid constrains. This property is useful for trajectory clustering because the vehicles are rigid moving objects on the road. And above all, the Euclidean distances between the corresponding points belong to two trajectories which from a same vehicle is a fixed value in 3-D space and the variance of those distances is zero. While for two trajectories from two different vehicles, the variance is not zero when a speed difference exists between two vehicles. Based on those properties and the 3-D coordinates of trajectory points, the trajectory similarity metric and clustering can be improved.



(a) (b)

Fig. 4. The result of trajectories relative height calculation. (a) Trajectories set with the red one denoted as the trajectory has the smallest height value; (b) relative heights of trajectories calculation under the height of red one in (a) is reset to 0. (Color figure online)

4 Trajectory Clustering

Based on the trajectory information in 3-D space and choosing the variance of the Euclidean distances between the corresponding points belong to two trajectories as a trajectory similarity metric, a clustering result can be obtained by the DBSCAN method [13]. Actually, the vehicle similar to a cuboid, and the size of the most vehicles in city with in the size 5.0m × 1.8m × 1.6m (long × width × height), thus we conduct a scale transformation to the trajectory points in 3-D space for a better performance of DBSCAN by considering the vehicle as a cube-like structure. The Table 1 shows the scale parameter we used, and the clustering result for Fig. 2 based on DBSCAN is shown in Fig. 5.

Table 1. Scale parameter for cuboid-like vehicles

Size of vehicle (cuboid-like structure)	Long	Width	Height
	5.0 m	1.8 m	1.6 m
Scale parameter	1.0	2.778	3.125
Size after transformed (cube-like structure)	5.0	5.0	5.0

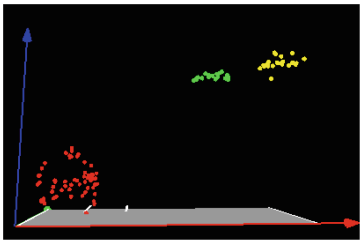
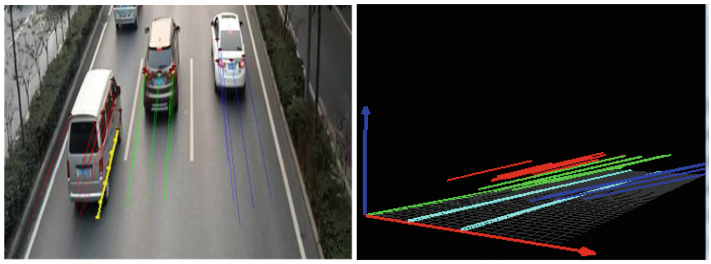


Fig. 5. Trajectory clustering result based on DBSCAN for Fig. 2

Based on the Eq. (8), the 3-D information of all trajectories in 2-D image can be reconstructed in a 3-D data space and a trajectory clustering can be done. Figure 6(a) shows a 2-D trajectory set and its clustering result, Fig. 6(b) shows their corresponding 3-D data set with normalized height information, in which the yellow trajectory is the one with the smallest height value (Fig. 6(a)) and the heights of the other trajectories is calculated by set yellow one's height to 0 (Fig. 6(b)).

As that is shown in Figs. 4 and 6, if the speeds of different rigid moving objects are not consistent with each other, the heights of the trajectories will present an obvious gap, and the trajectories can be clustering based on the gap difference. If the speeds of different rigid moving objects are consistent with each other, the trajectory clustering is also easy to take through using the lane-constrained, vehicle size and location. Additionally, if the trajectories come from a same rigid moving object, the height difference between them is stability, while if the trajectories belong to different rigid moving objects, the height difference between them has a large variation when the speed



(a) (b)

Fig. 6. Trajectories in 2-D image and their 3-D reconstruction. (a) The trajectories in 2-D image, the one with the smallest height is denoted as yellow; (b) the 3-D reconstruction of trajectories in (a). (Color figure online)

difference exists. For different realistic application environment, if feature descriptor for tracking is rich, DBSCAN (Density Based Spatial Clustering of Application with Noise) clustering method can achieve good results. If trajectories are sparse, it is best to make full use of lane-constrained, vehicle size and location information for clustering.

5 Experimental Results and Conclusions

Two video data were conducted to examine the effectiveness of the trajectory analysis method proposed in this paper. One is the video data of monitor scene for a beltway in city of Xi'an, and another is a monitor scene of Sanlu Outer Ring Road in city of Shanghai. Based on the proposed trajectory analysis method, the 3-D information of trajectories set can be obtained and described in a 3-D data space. Then combining with DBSCAN clustering algorithm, the vehicle segmentation can be realized effectively. In this processing scheme, the cuboid-like vehicle structure feature is using in DBSCAN clustering method. In this way, the vehicle segmentation result is a robust and accurate one. The rigid constrain used in the proposed method are chosen to estimate whether two trajectories belong to a same rigid moving vehicle or not. The estimation method based on Eq. (9) is an accurate numerical calculation problem, thus the results is accurate and reliable and can be used to trajectory based vehicle counting application. The vehicle counting accuracy is chosen as a quantitative index to evaluate the performance of the proposed method in realistic application environment (Fig. 7 shows two monitor scenes we used). The video data are tested on a Windows XP platform with a Pentium 4 3.2-GHz central processing unit (CPU) and 2-GB random access memory (RAM). The proposed algorithm is implemented with Visual C++ on a raw video format. Table 2 shows the video information and vehicle counting results based on the proposed method.

The result in Table 2 shows that in the situation with a larger scene and a lower resolution, the vehicle counting accuracy cannot up to 100 % even the 3-D information of trajectories are provided. It does not means the rigid constrains and 3-D information is low value information. Actually, the main factor induces the accuracy less

(a) (b)

Fig. 7. Two test monitor scene. (a) A monitor scene for a beltway in city of Xi'an; (b) a monitor scene for Sanlu Outer ring road in city of Shanghai.

Table 2. Vehicle counting result for video data

Monitor scene	A beltway in Xi'an	Sanlu road in Shanghai
Image size	1280*720	720*288
Descriptor for tracking	ORB tracking	Corner feature tracking
Total frame	755 frames	9320 frames
Total number of vehicles	69 vehicles	503 vehicles
Number of counted vehicles	67	497
Relative accuracy	97.1 %	98.8 %
Number of right counted	67	459
Absolute accuracy	97.1 %	91.2 %

than100 % is the provided trajectories are not good enough in some cases, such as incomplete trajectories induces by serious barrier, a lower distance resolution of height between trajectories which is caused by large scene, the segmentation error for large container trucks. Additionally, the proposed method is helpless for segmentation of two vehicles that moving synchronously with very small traffic spacing.

Fortunately for us, the value of the proposed method is that two important properties are found (based on Fig. 2) by using the rigid constrains for moving objects, and it provides a method for obtaining complete 3-D information of the trajectories. We test the effectiveness of the important properties by applying them for vehicle counting in two realistic monitor situations. The future work will concentrate on two aspects: one is attempting to use a richer image feature descriptor to vehicle tracing, the other one is reconstructing the surface of vehicles in 3-D space based on the 3-D coordinates of the feature and given a 3-D space model for moving objects.

Acknowledgement. This work was supported by the National Natural Science Foundation of China under Grants 61572083, Key Program of Natural Science of Shannxi under Grants 2015JZ018 and the Fundamental Research Founds for Central University under Grants 310824151034.

References

1. Morris, B.T., Trivedi, M.M.: A survey of vision-based trajectory learning and analysis for surveillance. IEEE Trans. Circ. Syst. Video Technol. **18**(8), 1114–1127 (2008)
2. Hu, W.M., Tian, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. **34**(3), 334–337 (2004)
3. Sivaraman, S., Trivedi, M.M.: Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis. IEEE Trans. Intell. Transp. Syst. **14**(4), 1773–1796 (2013)
4. Wang, W.C., Chung, P.C., Cheng, H.W., Huang, C.R.: Trajectory kinematics descriptor for trajectory clustering in surveillance video. In: IEEE International Symposium on Circuits Systems, pp. 1198–1201 (2015)
5. Mitsch, S., Müller, A., Retschitzegger, W., Salfinger, A., Schwinger, W.: A survey on clustering techniques for situation awareness. In: Ishikawa, Y., Li, J., Wang, W., Zhang, R., Zhang, W. (eds.) Web Technologies and Applications. LNCS, vol. 7808, pp. 815–826. Springer, Heidelberg (2013)
6. Fu, Z., Hu, W., Tan, T.: Similarity based vehicle trajectory clustering and anomaly detection. In: Proceedings of the IEEE International Conference on Image Processing, September 11–14, vol. 2, pp. II-602-605 (2005)
7. Atev, S., Miller, G., Papanikolopoulos, N.P.: Clustering of vehicle trajectories. IEEE Trans. Intell. Transp. Syst. **11**(3), 647–657 (2010)
8. Abraham, S., Lal, P.S.: Spatio-temporal similarity of network-constrained moving object trajectories using sequence alignment of travel locations. Transp. Res. Part C **23**, 109–123 (2012)
9. Hao, J.Y., Gao, L., Zhao, X.: Trajectory clustering based on length scale directive Hausdorff. In: Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013
10. Debnath, M., Tripathi, P.K., Elmasri, R.: A novel approach to trajectory analysis using string matching and clustering. In: Proceedings of IEEE 13th International Conference on Data Mining Workshops, The Dallas, TX, December 7–10, pp. 986-993 (2013)
11. Fan, Y., Xu, Q., Guo, Y.J., Liang, S.: Visualization on agglomerative information bottleneck based trajectory clustering. In: Proceedings of 2015 IEEE 19th International Conference on Information Visualisation, Barcelona, July 22–24, pp. 557-560 (2015)
12. Zhang, Z.X., Li, M., Huang, K., Tan, T.: Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance. In: proceedings of IEEE Conference on Computer Vision and Pattern Recognition, the Anchorage, AK, June 23–28, pp. 1-8 (2008)
13. Uncu, O., Gruver, W.A., Kotak, D.B., Sabaz, D.: GRIDBSCAN: GRId density-based spatial clustering of applications with noise. In: Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, October 8–11, vol. 4, pp. 2976-2981 (2006)