Sustainable Cities and Society 42 (2018) 547–557

Contents lists available at ScienceDirect

Sustainable Cities and Society

journal homepage: www.elsevier.com/locate/scs

Vehicle motion segmentation using rigid motion constraints in traffic video

Xuan Wang*, Huansheng Song, Qi Guan, Hua Cui, Zhaoyang Zhang, Haiying Liu

School of Information Engineering, Chang'an University, Xi'an, China

ARTICLE INFO

Keywords:
Motion segmentation
Trajectory clustering
Camera calibration
Rigid motion constraints
Inverse projection mapping
Spectral clustering

ABSTRACT

Vehicle motion segmentation is an important task in Intelligent Transportation Systems (ITS) for the sustainable development of smart city. In this paper, a novel trajectory clustering method is proposed using the rigid motion constraints (RMC) for the vehicle motion segmentation. Firstly, camera calibration is done for converting the 2D images to a copy one in 3D world. Then, the inverse projection image (IPI) can be obtained with a known 3D height information and the 2D image. Secondly, we present and verify the RMC of the feature point trajectories in the same rigid object and different rigid objects. By analyzing simulation results, we use the RMC as a measure to define the similarity function. Finally, spectral clustering is adopted to achieve the trajectory clustering, which contains two stages: initial clustering and inter-class merging. By applying this clustering method in actual traffic scene, much more stable clustering results can be obtained. Experimental results on traffic video demonstrate that the proposed method has a good performance in different traffic scenes. Moreover, its accuracy can reach 96% compared with other methods.

1. Introduction

In the past several years, vehicle motion segmentation and behavior analysis of traffic video have emerged as an important research area (Song et al., 2014; Sivaraman and Trivedi, 2013; Kanhere and Birchfield, 2008). We have proposed a real-time vehicle behavior analysis system in Song et al. (2014), which used the motion feature of trajectories to analyze the vehicle behavior. However, the trajectory clustering still remains an open problem in Song et al. (2014). In order to improve the effectiveness of the system, this paper focuses on the problem of the trajectory clustering in real complex traffic scenes.

Vehicle trajectory, which contains the motion information of the object, is often used to segment the motion vehicle with similar spatial and dynamic patterns. Thus, trajectory clustering has been an extremely active research area in the Intelligent Transportation Systems (ITSs) in recent years (Atev et al., 2010; Besse et al., 2016; Wang et al., 2017). Numerous trajectory clustering methods have been proposed to solve this problem for different application scenes. The state-of-the-art clustering methods are mainly including *k*-means (Coates and Ng, 2012), Robust Clustering Using LinKs (ROCK) (Guha et al., 1999), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996), WaveCluster (Sheikholeslami et al., 1998) and COBWEB (Fisher, 1987).

Many methods cluster the trajectory data based on distance measures between trajectories. The researchers focus on clustering the trajectories which have similar paths based on the distance between trajectory objects. A number of work has been done to give the definitions of trajectory distances. For instance, Berndt and Clifford (1994) proposed a distance measure method for trajectories and general time series, called Dynamic Time Warping (DTW). Later, Vlachos et al. (2002) used the Longest Common Subsequence (LCSS) to measure the similarity of trajectories. Chen and Ng (2004) and Chen et al. (2005) used the edit distance to define the distance function, called Edit distance with Real Penalty (ERP) and Edit Distance on Real sequence (EDR) respectively. These methods defined the similarity distance in the same way, but the difference is that they use different cost functions. Another approach attempts to obtain the geometric features of trajectories, such as their shape. Shape-based distances like Hausdorff distance (Hausdorff, 1914) and Fréchet distance (Fréchet, 1906) are used mostly to compare the similarity of the trajectories. Moreover, other researchers use different methods to do some research on trajectory clustering. For instance, Jung et al. (2008) proposed an abnormal event detection method based on trajectory clustering. It contains two stages: the extracted trajectories were clustered using the statistical histogram of 4D motion during the training stage of the data set, and it used the 4D motion histogram to cluster trajectory in the test stage. Saleemi et al. (2009) used a multivariate non-parametric probability density function to model the behavior of the moving objects to cluster different motions. Morris and Trivedi (2008) used an unsupervised method to learn the normal motion patterns of the objects in the scene, and coded

* Corresponding author.
E-mail address: jessica036@126.com (X. Wang).

https://doi.org/10.1016/j.scs.2018.07.026
Received 7 August 2017; Received in revised form 26 April 2018; Accepted 26 July 2018
Available online 11 August 2018
2210-6707/

the motion features of the space-time trajectories by hidden Markov models to realize the detection of abnormal trajectories. Piotto et al. (2009) proposed an abnormal event detection method based on semantic trajectory in order to realize trajectory classification and abnormal event recognition. Veeraraghavan and Papanikolopoulos (2009) proposed using a random context-free grammar to model the behavior of moving targets through semi-supervised learning methods. Johnson and Hogg (1995) modeled the probability distribution of trajectories using two interconnected competing neural networks.

Since the traffic video sequence is continuous, we can get the motion information of vehicles to deal with the clustering. Based on this condition, it brings some new ideas and methods for the trajectory clustering. The researchers of Han et al. (2015) group the clustering methods into two types: subspace-based methods (Elhamifar and Vidal, 2013; Li and Vidal, 2015; Liu et al., 2013) and affinity-based methods (Han et al., 2015; Li and Vidal, 2014; Guo et al., 2013; Dragon et al., 2012). Subspace-based methods cluster different objects based on a data matrix constructed by using all feature point trajectories. For instance, Elhamifar and Vidal (2013) proposed Sparse Subspace Clustering (SSC) to cluster data points that lie in a union of low-dimensional subspaces. Li and Vidal (2015) presented a unified optimization framework for learning both the similarity matrix and the clustering segmentation. Liu et al. (2013) used the Low-Rank Representation (LRR) to seek the lowest rank representation among all the candidates that can represent the given data. Although these methods have performed good on the experiment of the Hopkins 155 dataset, subspace-based methods may be not able to solve some problems in practice. Specifically, when the objects are occluded by other objects in the traffic scenes, subspace-based methods may have poor performance. In this condition, the feature point trajectories are missing and it cannot be recovered completely with the state-of-the-art methods. Thus, subspace-based methods have low robustness in real-time traffic scenes. In contrast, affinity-based methods use an affinity matrix constructed from affinities between pairs of feature point trajectories to cluster. In recent work, Li and Vidal (2014) used the epipolar constraint to compute the Randomized Voting (RV) score for each feature point and the scores are accumulated to cluster the feature points. Similarly, Guo et al. (2013) solved the 3D clustering problem using the epipolar constraint of two perspective views. Moreover, Dragon et al. (2012) built a bottom-up frame-to-frame motion segmentation framework on multiple time scales, which used the motion-split-and-merge (MSAM) algorithm for initial clustering and used the multi-scale motion clustering (MSMC) approach for trajectories with various lengths, time spans and missing data.

Although these state-of-the-art methods have obtained good performance on the trajectory dataset, subspace-based methods may fail to handle the some problems (e.g., noise and occlusion). In contrast, these problems have little influence in the affinity-based methods. Therefore, the proposed method belongs to affinity-based methods. However, the main challenge of affinity-based methods is that the number of clustering needs to be predefined. Meanwhile, the computation cost cannot to be too high for real-time application.

In this paper, we focus on proposing and validating the rigid motion constraints of the vehicle for trajectory clustering. We present an effective and robust method for the problem of vehicle trajectory clustering, which contains two stages: initial clustering and inter-class merging. The main contribution of this paper are as following:

- IPI is the copy image of converting the 2D images to 3D space. It deals with the effects of occlusion on 2D image objects, which uses 3D trajectory points to achieve the clustering.
- RMC is proposed to be regarded as a measure of similar trajectories, which is used to define a similarity function for the construction of affinity matrix.
- We adopt a strategy to estimate the number of the clustering for the spectral clustering and obtain the initial clustering results. Then inter-class merging steps are employed to improve the experimental results.

The rest of this paper is organized as follows. Section 2 defines the mapping relationship of inverse projection. The rigid motion constraints are analyzed in Section 3. Section 4 displays the vehicle trajectory clustering method. Experimental results are shown in Section 5 and finally Section 6 draws our conclusions.

2. Inverse projection transformation

In this paper, 3D information of the object are used to analyze the problem of clustering. Thus, it is necessary to calibrate the camera for gaining the affine camera model. Then, the affine matrix can be calculated to recover the 3D information. Assume that $P_W(X_W, Y_W, Z_W)$ is a 3D coordinate point in the world coordinate system, and $p_I(u, v)$ is its corresponding point in the image plane. Therefore, the relationship between the world coordinate system and the image coordinate system can be established by Zheng and Peng (2014):

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}, \tag{1}$$

where K is the camera internal parameters, R and t compose the external parameter matrix of the camera. The internal and external parameters can be calculated accurately by the recovery method of vanishing points (Zheng and Peng, 2014). As shown in Fig. 1, three

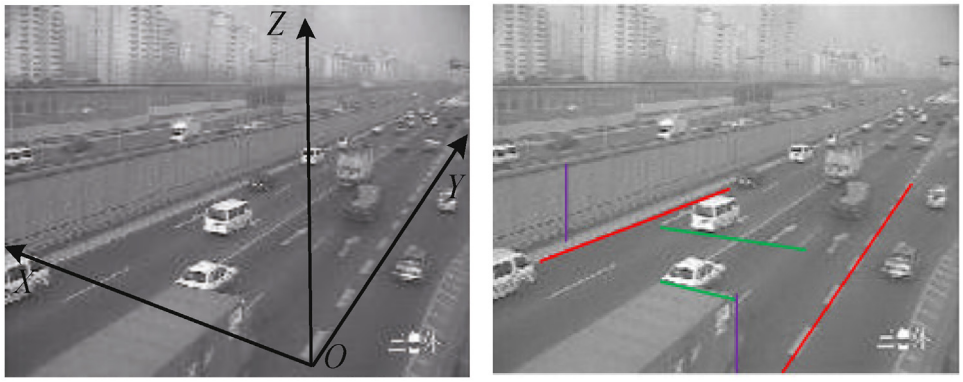


Fig. 1. Camera calibration. (a) World coordinate system. (b) Calibration process.

vanishing points are formed in the image using three sets of parallel lines in the directions of *X*-axis, *Y*-axis, and *Z*-axis respectively. In addition, a known height (the 3D height of camera) is required to calculate the scale factor. Then, the camera calibration matrix can be calculated.

It is known that the object in 2D image lose the original geometric feature, and it has changed of scale and motion state. Thus, it is necessary to recover the 3D information. However, if the feature point on 2D image is projected to 3D space, the inverse projection result is not unique. Only if its 3D height information is known, the inverse projection result can be got uniquely by (1), then we can get the inverse projection image (IPI).

In this paper, we only focus on the feature point trajectories. Since the trajectories, which provide spatial-temporal information, are the best feature to describe the motion properties of objects. The feature point trajectory can be defined as a data sequence, which consists of several concatenated state vectors from tracking. It means that the trajectory is a indexed sequence of feature point positions in a given time window. Thus, 2D feature point trajectory can be represented as a point data set as follows:

$$T_{2D}(i,\ n) = \{p(i,\ 1),\ p(i,\ 2),\ p(i,\ 3),\ ...,p(i,\ n)\}, \tag{2}$$

where i is the number of a given trajectory in the data set, n is the number of points on the trajectory, $p(i,\ n)$ is the 2D coordinate of trajectory point. Since the 3D height of the feature point trajectory in the rigid object is constant, the description of the 3D feature point trajectory can be got as

$$T_{3D}(i,\ h,\ n) = \{P(i,\ 1),\ P(i,\ 2),\ P(i,\ 3),\ ...,P(i,\ n)\}, \tag{3}$$

where $P(i,\ n)$ is the 3D coordinate of trajectory point. However, the real 3D height information is hard to get directly from 2D trajectory in image plane. Fortunately, for a moving rigid object, the feature points on its surface have a same property, that is RMC.

3. Rigid motion constraints relationship analysis

Rigid object is an idealized solid for which its deformation is neglected. In other words, the relative position of any two given points on the rigid object is fixed regardless of external forces exerted on it. However, there is no such an object as perfectly rigid in fact, since object will appear varying degrees deformation on the action of forces in relativity. But for moving vehicle, its deformation, which is extremely small relative to the whole vehicle, can be neglected during the movement. In this paper, vehicle is regarded as a rigid object in the traffic scene, and its movement is regarded as rigid object movement to be studied and analyzed.

There are two characteristics for the movement of rigid object: (a) Any two points maintain the same distance relative to each other during the movement; (b) All points on a rigid object move with the same displacement and velocity. The two characteristics are called Rigid motion constraints (RMC). Then we use two ideal 3D trajectories to analyze the RMC for the same and different rigid objects.

Assuming that there are two feature points in the same rigid object, they are tracked accurately during the movement. The two trajectories both have n continuous trajectory points. Then, RMC can be expressed by

$$C_1(i,\ j,\ n) = \int_1^{n-1} |D_{P(i,t)}^{P(i,t+1)} - D_{P(j,t)}^{P(j,t+1)}|\ \mathrm{dt} \tag{4}$$

$$C_2(i,\ j,\ n) = \int_1^{n-1} |D_{P(i,t+1)}^{P(j,t+1)} - D_{P(i,t)}^{P(j,t)}|\ \mathrm{dt}, \tag{5}$$

where $D_{P(i,t)}^{P(i,t+1)}$ is the Euclidean distance in 3D space between the trajectory points of the same trajectory in adjacent time, $D_{P(i,t)}^{P(j,t)}$ is the 3D distance between the two feature points in the same time.

In terms of traffic video, it is difficult to obtain a continuous trajectory due to the influence of camera sampling frequency. As shown in Fig. 2, the trajectory points are discrete with the same time interval, the equation (4) and (5) can be transformed into

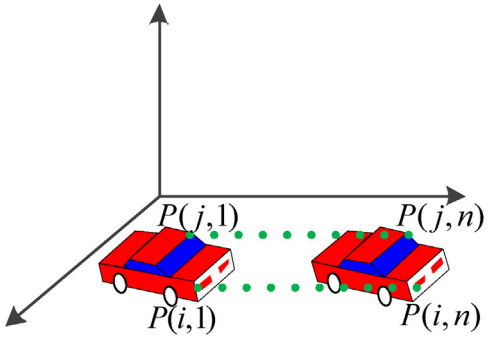


Fig. 2. Two simulative trajectories with discretization points of the same rigid object.

$$C_1(i,\ j,\ n) = \sum_{t=1}^{n-1} |D_{P(i,t)}^{P(i,t+1)} - D_{P(j,t)}^{P(j,t+1)}| \tag{6}$$

$$C_2(i,\ j,\ n) = \sum_{t=1}^{n-1} |D_{P(i,t+1)}^{P(j,t+1)} - D_{P(i,t)}^{P(j,t)}|. \tag{7}$$

In theory, the two trajectories belong to the same rigid object, they should satisfy the RMC, i.e., $C_1(i,\ j,\ n) = 0$ and $C_2(i,\ j,\ n) = 0$. If the two trajectories belong to different rigid objects, they are not able to satisfy the RMC, i.e., $C_1(i,\ j,\ n) \neq 0$ or $C_2(i,\ j,\ n) \neq 0$. In terms of video image, only the 2D trajectories can be obtained. In order to analyze the RMC between two trajectories in the image plane, two ideal 2D trajectories in the image plane are needed to do the test. The steps are as following:

- In order to obtain the ideal 2D trajectory in the image plane, we assume that there are two 3D trajectories in a calibration traffic scene and the 3D coordinates are known. Then the 2D trajectory points can be calculated by (1). The two ideal 2D trajectories can be described as:

$$\begin{aligned} T_{2D}(i,\ n) &= \{p(i,\ 1),\ p(i,\ 2),\ p(i,\ 3),...,p(i,\ n)\} \\ T_{2D}(j,\ n) &= \{p(j,\ 1),\ p(j,\ 2),\ p(j,\ 3),...,p(j,\ n)\}. \end{aligned} \tag{8}$$

- We need to back-project the 2D trajectory points into 3D space in order to analyze the RMC. However, the 3D height information cannot be got directly from the real traffic video. Fortunately, the 3D height of vehicle is a bounded value. For this prior knowledge, we can construct a set of test using the method of 3D height enumeration (the range is 0-4 m and the interval is 0.01 m) to estimate the 3D height. Then two sets of 3D trajectories can be obtained by each 3D enumeration height:

$$\begin{aligned} T_{3D}(i,\ h_i,\ n) &= \{P(i,\ 1),\ P(i,\ 2),\ P(i,\ 3),...,P(i,\ n)\} \\ T_{3D}(j,\ h_j,\ n) &= \{P(j,\ 1),\ P(j,\ 2),\ P(j,\ 3),...,P(j,\ n)\}, \end{aligned} \tag{9}$$

  where $h_i = \{0,\ 0.01,\ 0.02,\ 0.03,\ \ldots,\ 4\}$, $h_j = \{0,\ 0.01,\ 0.02,\ 0.03,\ \ldots,\ 4\}$. It means that we can reconstruct N 3D trajectories for each 2D trajectory.
- Calculate the RMC between each two 3D trajectories. For each (h_i, h_j), there are two constraints $C_1(i,\ j,\ n)$ by (6) and $C_2(i,\ j,\ n)$ by (7). Finally, we can get a $N \times N$ matrix for each constraint. If h_i, h_j, $C_1(i,\ j,\ n)$ are regarded as X axis, Y axis, and Z axis, respectively, we can draw these points using the scatter plot. Then, a concave surface can be achieved by fitting these points, so do h_i, h_j, and $C_2(i,\ j,\ n)$.

Assuming that the two 2D trajectories belong to the same rigid object, it can be found that the surface of $C_1(i,\ j,\ n)$ has a intersecting line with the zero plane, so does the surface $C_2(i,\ j,\ n)$. Putting the two

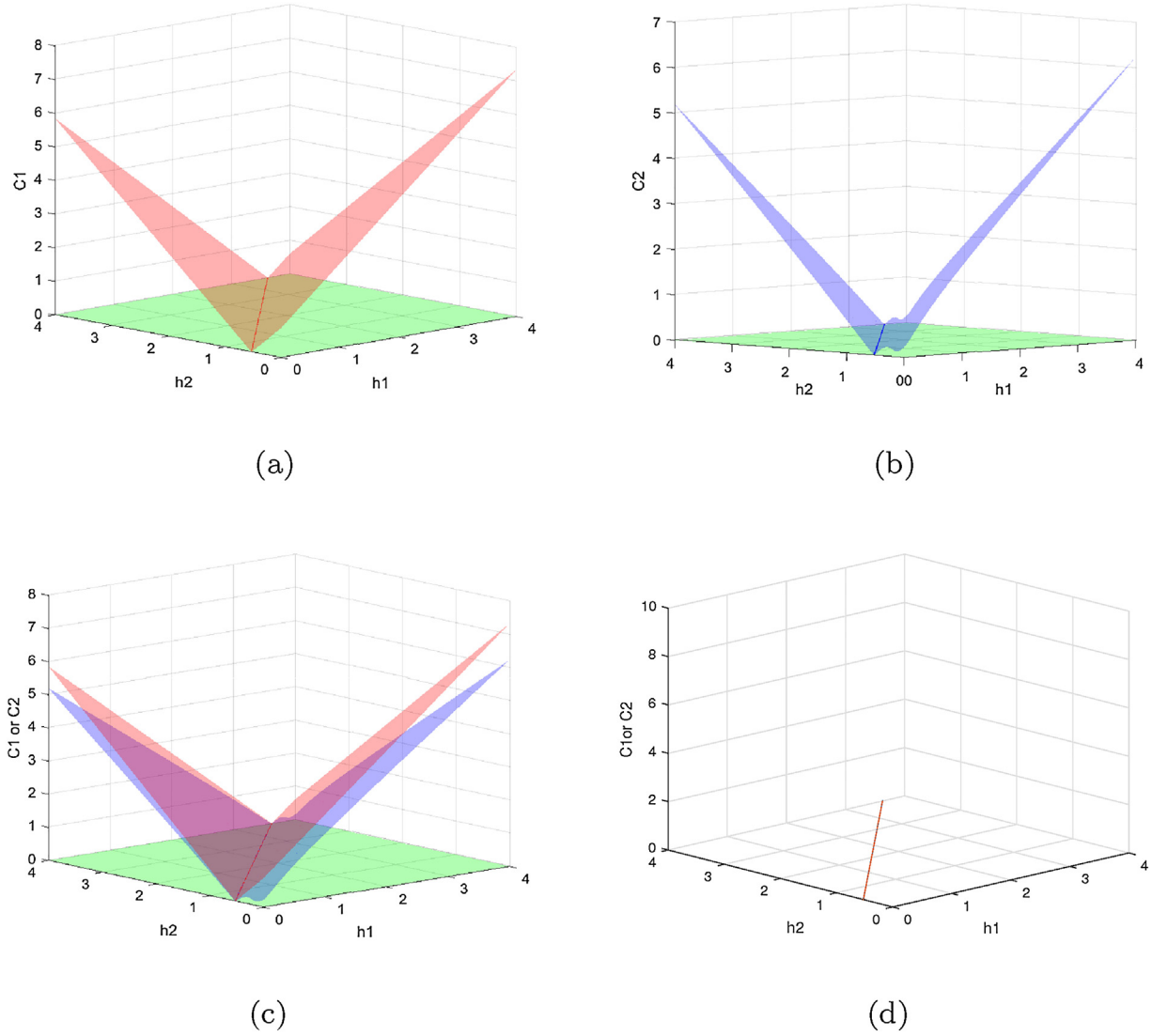


Fig. 3. Simulation the RMC for two ideal trajectories belonging to the same rigid object. (a) Fitting effect of h_i, h_j, and $C_1(i, j, n)$. (b) Fitting result of h_i, h_j, and $C_2(i, j, n)$. (c) Fitting result in the same coordinate system. (d) Fitting line of the lowest points.

surface in the same coordinate system, we can see that the two lines are overlapped, as shown in Fig. 3. A large number of experiments show that if the two 2D trajectories belong to the same rigid object, the two lines will be overlapped.

Assuming that the two trajectories belong to different rigid objects, they may have different linear velocities or different directions of motion. As shown in Fig. 4, there are two different situations: (a) The two trajectories have different velocities, but they have the same direction of motion; (b) The two trajectories have different directions of motion, but they have the same velocity. According to the above method, $C_1(i, j, n)$ and $C_2(i, j, n)$ of the two trajectories are calculated in the two motion cases, as shown in Fig. 5. It can be seen that if the two trajectories belong to different rigid objects, the two surfaces do not have an overlapped line, even does not intersect with zero plane. We have tested this motion characteristics in three different traffic scene and each scenes have been tested over 100 times. We find that all the rigid motions have the motion characteristics of RMC. Thus, it is suggested that RMC can be used to measure the similarity of trajectories. It means that RMC can be used to explain the possibility that the two trajectories belong to the same rigid object. In order to verify the effective of the proposed method, we use the real trajectory data, which is obtained by tracking from traffic video, instead of the ideal trajectory to test the proposed method.

As shown in Fig. 6, there are a series of trajectories in the traffic scene. They are divided into 4 groups, and each group belongs to the same car. We extract the top 25 of each tracked trajectory points to calculate the RMC. Some of the experimental results are shown in Figs. 7 and 8. From Fig. 7, it can be found that even though the two trajectories belong to the same vehicle, their fitting surface $C_1(i, j, n)$ and $C_2(i, j, n)$ do not have a overlapped line intersect at zero plane. The error results from the tracked trajectory data which has a certain deviation in the tracking process. Therefore, even though the two trajectories belong to the same vehicle, they are not completely parallel and their velocities are slightly different, i.e., $C_1(i, j, n) \neq 0$ or $C_2(i, j, n) \neq 0$. However, it can be found that even though the two surface, which is drawn by the two trajectories belonging to the same vehicle, does not have a overlapped line intersect at zero plane, the bottom points of the two surface can be fitted to two space lines, and the two lines has a height deviation ξ and angle offset θ within the error range. In this paper, we use the Outer Ring of Shanghai traffic scene, and threshold values are given by the statistical results of the experiment, i.e., $\xi \leq 0.2$ and $\theta \leq 5^{\circ}$. Therefore, if $\xi \leq 0.2$ and $\theta \leq 5^{\circ}$, it means that the two trajectories are considerable likely belong to the same car. It is able to use as a basis for constructing affinity matrix.

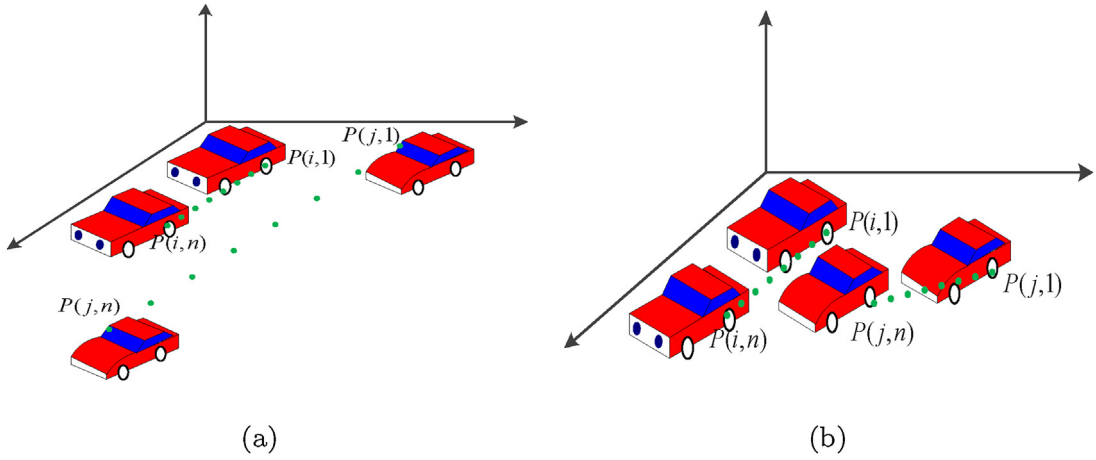


Fig. 4. Simulation trajectories of different rigid objects. (a) Two Trajectories have different velocities and the same direction of motion. (b) Two Trajectories have different directions of motion and the same velocity.

4. Trajectory clustering based on spectral clustering

Spectral clustering divides the weighted undirected graphs into two or more subgraphs using the eigenvalues and eigenvectors of the affinity matrix which associated with the graphs (such as adjacency matrix of the graph) to achieve the purpose of clustering. At present, the spectrum theory of graph is quite mature, and there are a lot of research results can be directly applied.

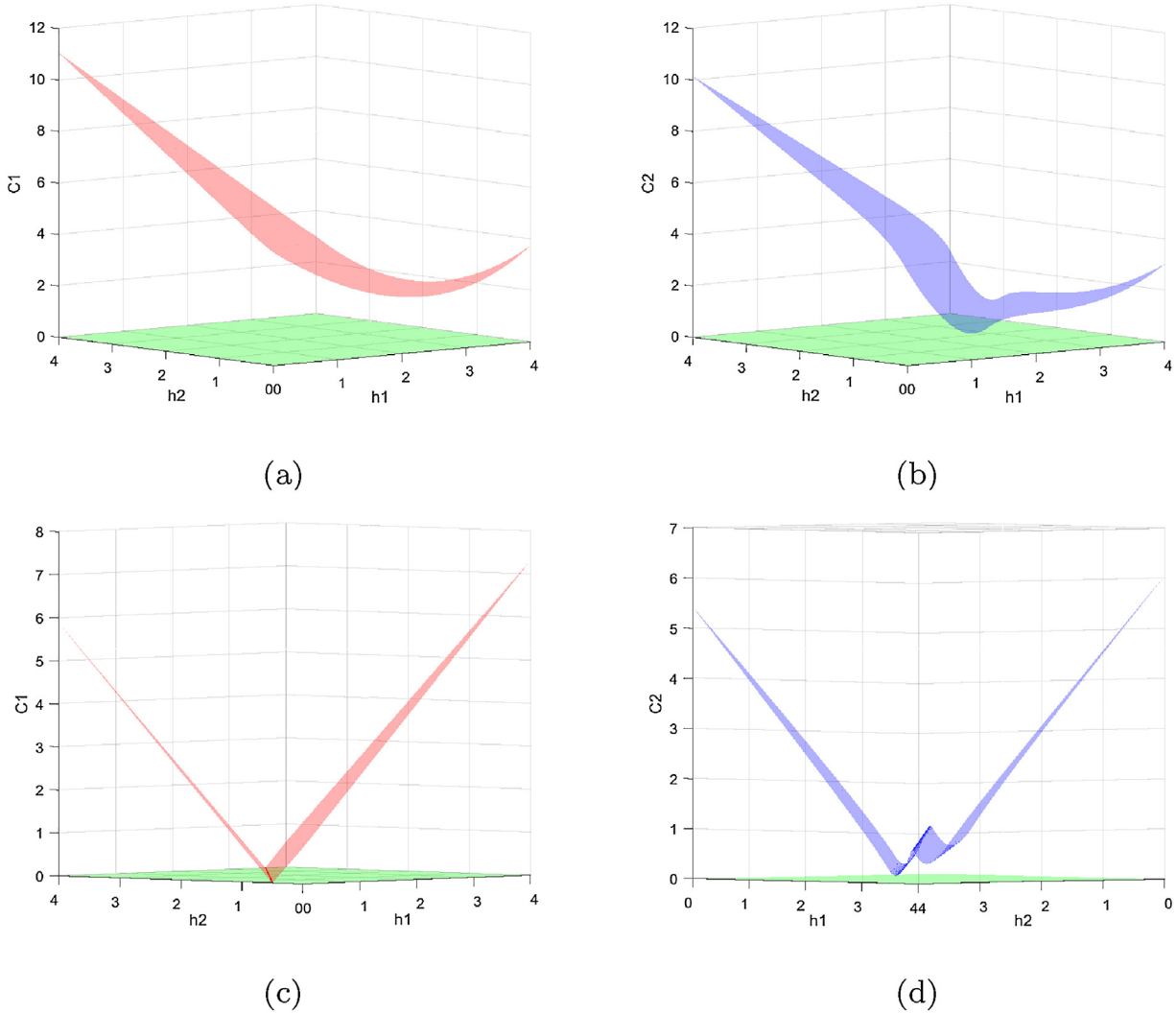


Fig. 5. Simulation the RMC for two ideal trajectories belonging to different rigid objects. (a) Fit h_i, h_j and $C_1(i, j, n)$ with different liner velocities. (b) Fit h_i, h_j and $C_2(i, j, n)$ with different liner velocities. (c) Fit h_i, h_j and $C_1(i, j, n)$ with different directions of motion. (d) Fit h_i, h_j and $C_2(i, j, n)$ with different directions of motion.

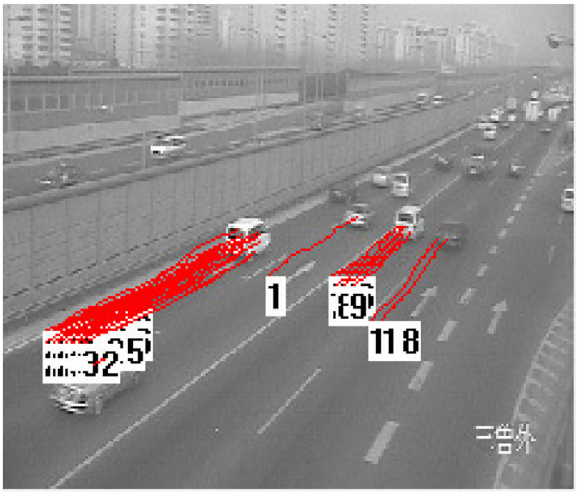


Fig. 6. Tracked trajectories.

4.1. construction of affinity matrix

Let $G = (V, E, \mathbf{W})$ is a weighted undirected graph, where V denotes the set of trajectory, E is the set of edges connecting the trajectory nodes, $\mathbf{W} = [w_{ij}]_{N\times N}$ is the affinity matrix with weights describing how likely two trajectories belong to the same group, N is the number of trajectory to be grouped.

According to the analysis of RMC, this paper uses it as a measure to define a similarity function so as to construct the affinity matrix. The similarity function is defined as:

$$w_{ij} = F(T_{2D}(i,t), T_{2D}(j,t)) = \begin{cases} 1 & \text{if } |\min(C_1(i,j,n)) - \min(C_2(i,j,n))| \le \xi \\ & \&\&|\arctan k_1 - \arctan k_2| \le \theta \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

where k_1 and k_2 indicate the slope of the fitting line as shown in Figs. 7 and 8.

Then, the affinity matrix can be constructed by the following steps:

- Use the enumeration 3D height h_i and h_j, we can reconstruct the 3D trajectories of $T_{2D}(i, t)$ and $T_{2D}(j, t)$. Because the height of vehicles is limited, the enumeration 3D height of this paper is in the range of 0-4 m. Using each enumerated height, we can recover a 3D trajectory. Thus, we can get two sets of 3D trajectories and both have N trajectories.
- Calculate the $C_1(i, j, n)$ and $C_2(i, j, n)$ of the each two 3D trajectories. Then, we can get a $N \times N$ matrix for both $C_1(i, j, n)$ and $C_2(i, j, n)$.
- Find the bottom points in the two matrices that is close to zero, respectively. Then we can get the fitting lines with the slopes k_1 and k_2, respectively.
- Calculate the w_{ij} by the similarity function of (10).

Finally, the affinity matrix $\mathbf{W}$ can be obtained, which is a $N \times N$ symmetric matrix.

Since the time complexity of the enumeration 3D height is $O(n^2)$, it is a large amount of computation which affect the real-time performance of the algorithm. From Fig. 3, it can be seen that the fitting surface is a concave and the cross section is a concave curve. Therefore, we can only enumerate the 3D height of one trajectory and use a trichotomy to search the lowest point of the other with the height range of 0-4 m. This method greatly reduces the computation, which can satisfy the real-time performance. As shown in Fig. 9, it can be found that there are slight differences between the two straight lines of the trichotomy search and the enumeration method, and the difference is within acceptable limits. The two fitting lines are overlapped using the enumeration method with the line equation $h_2 = 0.9597h_1 + 0.5201$. The lines are obtained by the optimized method with the two line equations $h_2 = 0.9596h_1 + 0.5201$, $h_2 = 0.9484h_1 + 0.4581$, respectively. The angle between the two lines is $\theta = 0.336^\circ$. This error can be modified by adjusting the threshold, thus the optimized method can be used to instead of the enumeration method. Meanwhile, the angle threshold should be changed, i.e., $\theta = 6^\circ$.

For the trajectory data extracted from the scene as shown in Fig. 10, there are four trajectories, of which ξ and θ are calculated using the optimized method as shown in Table 1. From Table 1, it can be found that some angles between the two trajectories is Null. This is because the height of vehicle cannot be over 4 m, but for the trajectories belonging to the different vehicles, the lowest point may be found at the boundary within the given range. It results in the slope of the fitted straight line is 90°. Thus, this case is marked as *Null*, i.e., $\theta = Null$. Therefore, the affinity matrix $\mathbf{W}$ can be constructed using the results of Table 1 by the similarity function as:

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (11)$$

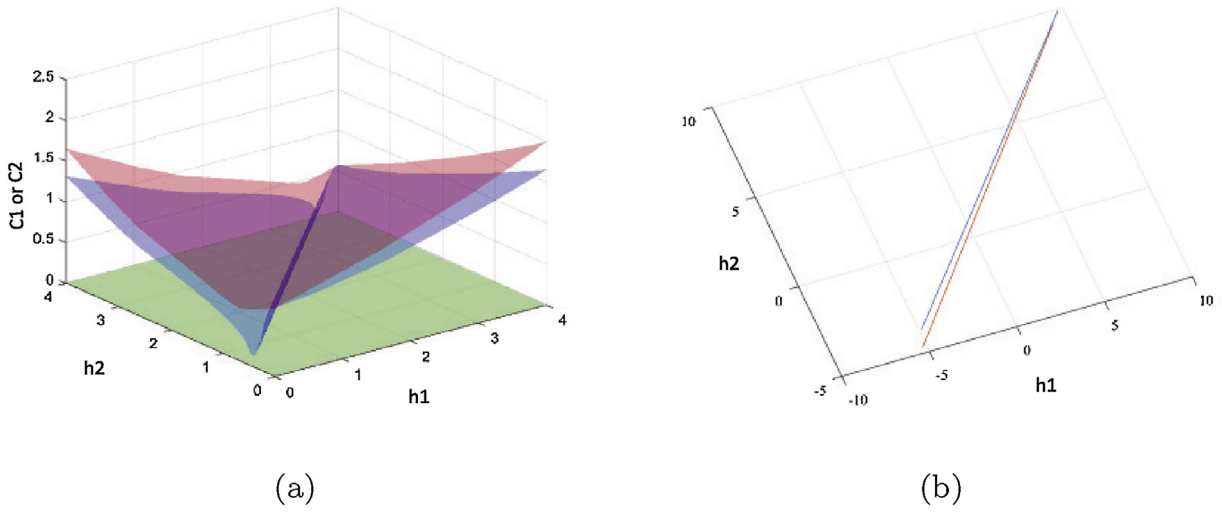


Fig. 7. RMC of two trajectories from the same vehicle. (a) $C_1(i, j, n)$ and $C_2(i, j, n)$ of the two trajectories. (b) Liner fitting results of the bottom points of (a).

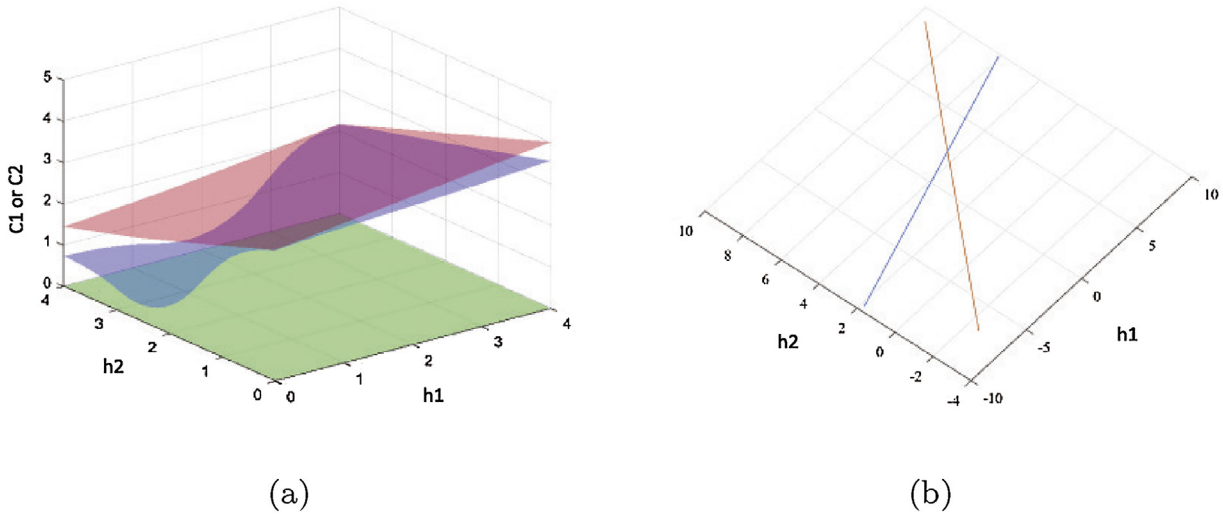


Fig. 8. RMC of two trajectories from different vehicles. (a) $C_1(i, j, n)$ and $C_2(i, j, n)$ of the two trajectories. (b) Liner fitting results of the bottom points of (a).

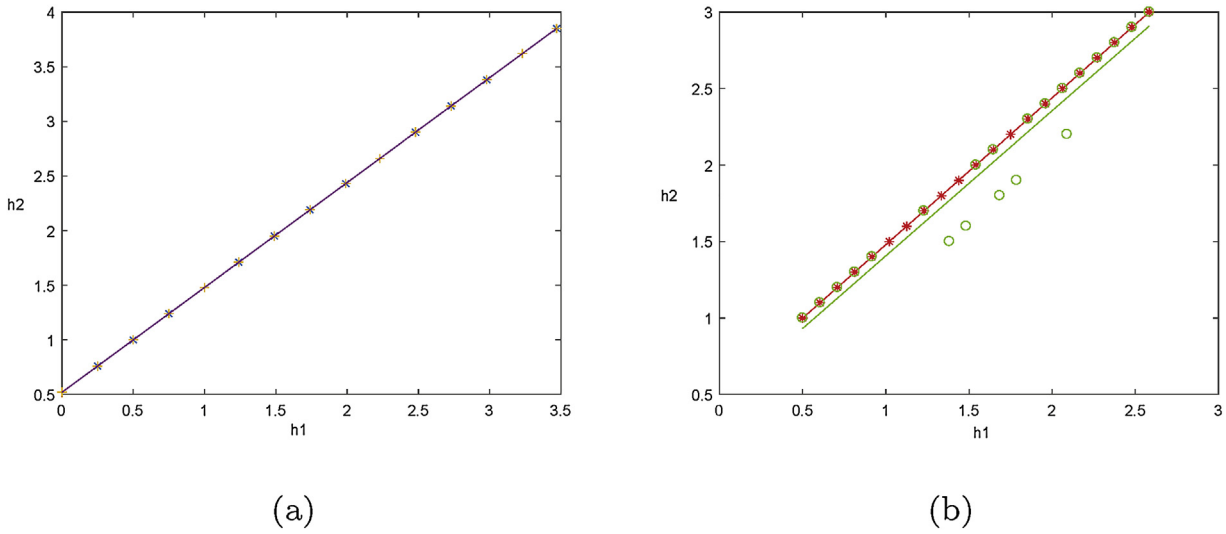


Fig. 9. Comparison of enumeration method and improved method. (a) Enumeration method. (b) Improved method.

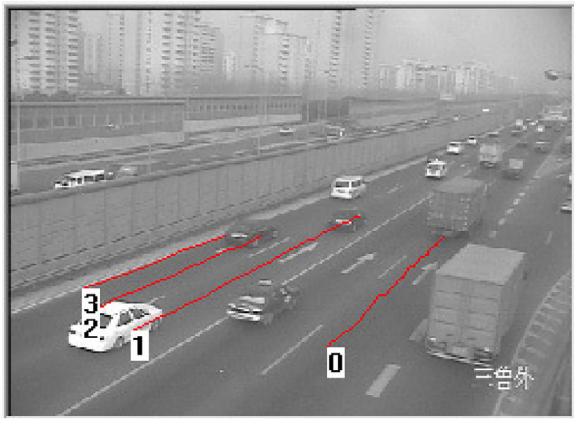


Fig. 10. Vehicle tracking trajectories.

4.2. Determine number of clustering

It is critical to determine the number k of clustering, which directly determines the trajectory data set to be divided into k categories. In this paper, the affinity matrix is handled directly. The eigenvalues of the affinity matrix should be arranged from largest to smallest. If the kth eigenvalue is much larger than the $(k + 1)$th eigenvalue, it is considered that k is the number of clustering in this trajectory data set. However, it is a problem that how to determine that the kth eigenvalue is much larger than the $(k + 1)$th eigenvalue. Fortunately, this paper employs a method to find the appropriate k and use the corresponding eigenvectors to construct the eigenvector space. The steps are as following:

Table 1
Results of ξ and θ.

Label	0 &1	0&2	0&3	1&2	1&3	2&3
ξ	1.7443	0.7945	0.55778	3.8603	2.5931	0.0879
θ	*Null*	*Null*	*Null*	25.35°	30.83°	1.97°

- Solve the eigenvalues $\{\lambda_i\}_{i=1,\ldots,n}$ and eigenvectors of affinity matrix.
- Sort the eigenvalues from largest to smallest, and calculate the sum of all eigenvalues:

$$S_n = \sum_{i=1}^{n} \lambda_i. \tag{12}$$

Table 2
Results of the eigenvalues and eigenvectors in Fig. 10.

Eigenvalues	Eigenvectors			
$\lambda_1 = 2$	0.0000	0.0000	1.0000	1.0000
$\lambda_2 = 1$	1.0000	0.0000	0.0000	0.0000
$\lambda_3 = 1$	0.0000	1.0000	0.0000	0.0000
$\lambda_4 = 0$	0.0000	0.0000	1.0000	1.0000

- Calculate the sum of the first k eigenvalues. If $\frac{S_k}{S_n} \geq \sigma$, the number of clustering can be defined as k. (In this paper, $\sigma = 95\%$ with which the clustering results reach the best in the test.)
- Construct the eigenvector space with the eigenvectors corresponding to the first k eigenvalues.

Table 2 shows the results of the eigenvalues and eigenvectors of affinity matrix **W**. It can be seen that $\frac{S_3}{S_n} = 100\% > 95\%$ while $\frac{S_2}{S_n} = 75\% < 95\%$, so the number of clustering is $k = 3$. It is the same as the number of real vehicles in the scene of Fig. 10. Meanwhile, the constructed eigenvector space is the first three eigenvectors in Table 2. The eigenvector space contains the segmentation information for the realization of spectral clustering algorithm.

4.3. Clustering method

4.3.1. Initial clustering

In this paper, k-means is used to cluster the eigenvectors in the eigenvector space. Its time-complexity is much less than dealing directly with the original data, since it is reduced dimension by constructing the eigenvector space. The computation will be much lower as the data is huge.

Combined with the main steps of spectral clustering and the practical application, the implementation steps of this paper are as following:

- Use the optimized method given in Section 4.1 to construct the affinity matrix **W** of the trajectory data set V.
- Determine the number of clustering k and construct the eigenvector space using the method given in Section 4.2.
- Employ the k-means to deal with the $N \times k$ eigenvector space. Then it outputs a $N \times 1$ dimensional clustering result matrix which giving the initial categories corresponding to the N trajectory objects.

4.3.2. Inter-class merging

The initial clustering can divide the trajectory objects into several categories. However, sometimes it can be found that the trajectories of the same vehicle is divided into multiple categories result from the error of constructed affinity matrix. Therefore, it is necessary to combine the results of initial clustering to merge the categories.

In this stage, the 3D information of feature points is need to be reconstructed in order to merge the different categories with the 3D vehicle models. The 3D vehicle models should be built by priori knowledge.

In terms of image plane, the movement of feature points are non-uniform motion, but the feature points back-projected on the zero plane ($Z_W = 0$) can be considered as uniform motion. In this case, the velocity of all feature points in the inverse projection plane (IPP) can be calculated, which is called back-projection velocity (BPV).

Assuming that the camera and the motion area are so far from each other, the camera and vehicle can be considered as in the same plane. As shown in Fig. 11, the vehicle moves from L_1 to L_2 in time t. Meanwhile, the vehicle feature points P and Q move from P_1 to P_2 and from Q_1 to Q_2, respectively. According to (1), the 2D points can be back-projected to the road plane with $Z_W = 0$ in 3D space. The back-projected points A_1, A_2, B_1, B_2 can be obtained, which are corresponding to P_1, P_2, Q_1, Q_2, respectively. According to similar triangle calculations, the similarity relation can be obtained:

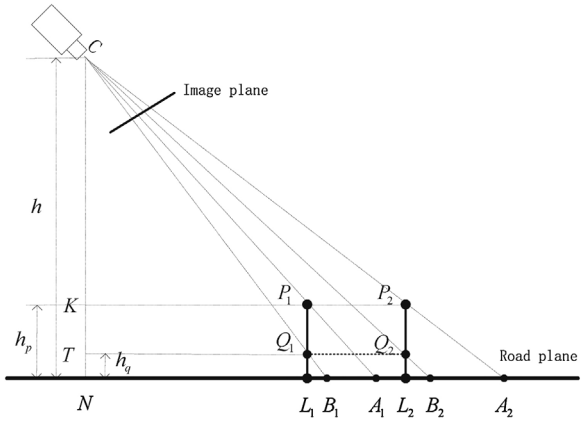


Fig. 11. Correlation between BPV and height.

$$\frac{P_1P_2}{A_1A_2} = \frac{h-h_p}{h} \quad \frac{Q_1Q_2}{B_1B_2} = \frac{h-h_q}{h}$$
$$\frac{v}{v_A} = \frac{h-h_p}{h} \quad \frac{v}{v_B} = \frac{h-h_q}{h}, \tag{13}$$

where v is the actual velocities of points P and Q. v_A and v_B are the back-projection velocities. From (13), it can be obtained that

$$\frac{v_A}{v_B} = \frac{h-h_q}{h-h_p}$$
$$h_p = \left(1 - \frac{v_B}{v_A}\right)h + \frac{v_B}{v_A}h_q. \tag{14}$$

Furthermore, if the 3D height of Q is zero, its BPV is considered as the actual velocity of the vehicle, i.e., $v_B = v$. Then

$$h_p = \left(1 - \frac{v}{v_A}\right)h. \tag{15}$$

Therefore, if the camera is set up high enough and the height h of camera is given, the 3D information of feature point can be reconstructed with its BPV and the real vehicle velocity. In this paper, we use this method to reconstruct the 3D information of all feature points, and then we merge the different categories belonging to the same vehicle by comparing with the 3D vehicle models. The specific steps of inter-class merging are as following:

- Assuming that $Class_i$ and $Class_j$ both have N trajectories, which are two categories in the results of initial clustering, the BPV of each trajectory can be calculated with $Z_W = 0$. The minimum BPV of each category is regarded as its reference velocity (RV) respectively and the corresponding feature point is considered as the reference feature point (RFP).
- Assuming that $Class_i$ and $Class_j$ belong to the same vehicle, the real velocity of the vehicle is estimated by the minimum BPV of the two categories, i.e., $v = \min(v_i, v_j)$. If $v = v_i$, $Class_i$ is the reference category (RC), $Class_j$ is the category to be merged (CTM), and vice versa.
- Reconstruct the 3D information of feature points for the two categories. The 3D height of each feature point can be calculated using v and its BPV by (15). Thus, the 3D information of the each feature point can be reconstructed by (1). Mark the 3D coordinates of the RFP as (X_i, Y_i, Z_i), and mark the CTM feature point P as (X_{jp}, Y_{jp}, Z_{jp}).
- Calculate the absolute distance between the CTM feature points and RFP, it can be expressed as:

$$D = \begin{cases} \Delta X = |X_{\text{jp}} - X_i| \\ \Delta Y = |Y_{\text{jp}} - Y_i| \\ \Delta Z = |Z_{\text{jp}} - 0|. \end{cases} \tag{16}$$

Table 3
Parameters of video sequences.

Sequences	Number	Size	Frequency
Outer Ring Road of Shanghai	22	720 × 288	25 FPS
Second South Ring road of Xi'an	23	1280 × 720	29 FPS
Highway of Hangzhou	31	1280 × 720	25 FPS

- Use the 3D vehicle models to determine whether the two categories need to be merged. At first, $\triangle Z$ is used to prejudge the 3D vehicle model. Then, if $\triangle X$ and $\triangle Y$ satisfy this model simultaneously, the parameter *CombineCount* is increased by 1, otherwise *CombineCount* is unchanged. In the end, if CombineCount $\geq \frac{2}{3}N$, $Class_i$ and $Class_j$ are merged and the number of category *ClassNum* is reduced by 1, otherwise, the *ClassNum* is unchanged.

5. Experiment results

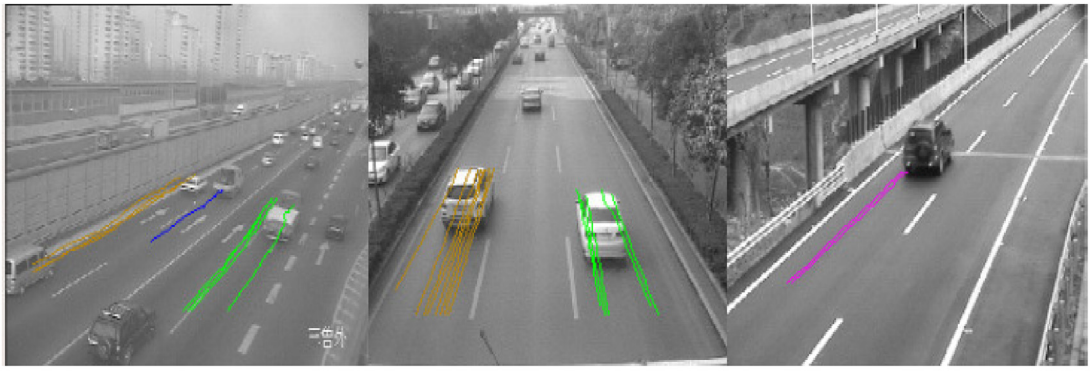
In this section, the experiment results are presented in different real video sequences in order to demonstrate the effectiveness and robustness of the proposed clustering method. These scenes are tested on a Windows 10 platform. The parameters of video sequences can be seen as Table 3. The proposed clustering method is implemented with Visual studio 2013 and MATLAB R2016b.

There are two parts in the experimental results. The clustering results are showed in Part A, which contains the initial clustering and inter-class merging. Part B presents a comparison with other methods.

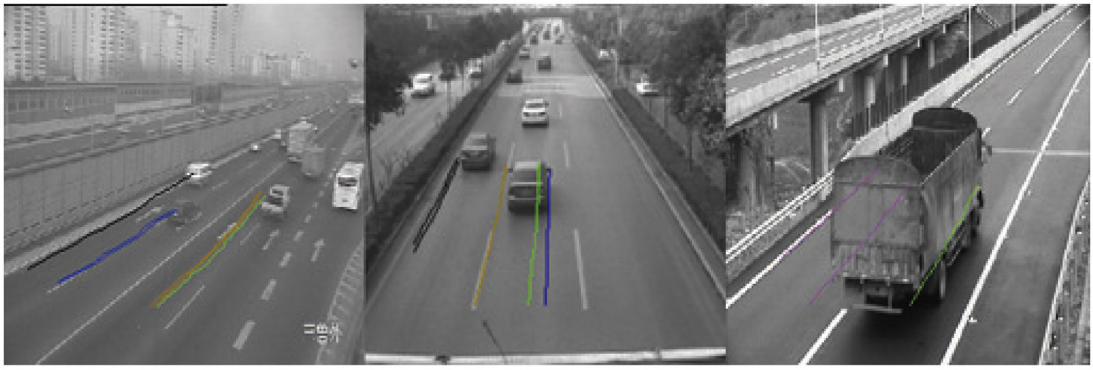
5.1. Results of clustering method

As shown in Fig. 12, there are two cases of the initial clustering results. In most cases, the initial clustering results are consistent with the number of vehicle, as shown in Fig. 12(a). However, sometimes the trajectory of a vehicle is divided into different categories, as shown in Fig. 12(b). With analyzing the experiment data, we find that the main reason for this case result from the construction of affinity matrix. In the steps of the construction of affinity matrix, we adopt an optimization method, which is given the range of the trajectory 3D height, to find the lowest point instead of the enumeration method. However, if the minimum value of the surface is not within the specified height range, it will lead to the error of fitting line. It considers that the similarity of the two trajectories is low, i.e., $w_{ij} = 0$. Then, it results in the error of clustering number k. Therefore, the inter-class merging needs to be done in order to solve the problem. The results of inter-class merging can be seen as Fig. 13. It can be found that the results of initial clustering are improved using the 3D vehicle models by the steps of inter-class merging. The 3D vehicle models are constructed by the prior knowledge, but they should be bigger than the real parameters as the reconstructed 3D information of the feature point is bigger than the real value. This is a system error just because we use the lowest BPV to replace the real vehicle velocity. The 3D vehicle models used in this paper are shown as Table 4.

Using the first scene of Fig. 12(b) as an example, there are 6 trajectories for 4 categories. We mark them in turn as Table 6. With the height of camera ($h = 14$ m), the image coordinates of the current feature points and the BPV of each trajectory can be calculated. The corresponding parameters are shown as Table 6. The clustering method executes every 25 frames in order to reduce the computation. Thus, it can satisfy the requirement of real-time traffic, and the running time



(a)



(b)

Fig. 12. Results of initial clustering.

Fig. 13. Results of inter-class clustering.

Table 4
Parameters of 3D vehicle models.

Type	*X*	*Y*	*Z*
Compact car	2 m	3.5 m	2 m
Mid-size car	2.5 m	5 m	2.5 m
Full-size car	3 m	10 m	4 m

Table 5
Running time.

Traffic scenes	Outer Ring Road of Shanghai	Second South Ring Road of Xi'an	Highway of Hangzhou
Time (s)	2.46	2.87	2.65

Table 6
Parameters of inter-class merging.

Label of category	Image coordinate	Back-projected velocity
1[1]	(277,161)	22.30 m/s
2[2]	(344,163)	26.28 m/s
2[3]	(350,164)	25.45 m/s
3[4]	(448,182)	20.59 m/s
3[5]	(457,180)	21.94 m/s
4[6]	(464,180)	22.46 m/s

can be seen as Table 5.

With the results of initial clustering, it can be found that the lowest BPV of each two categories is regarded as the estimation of the real vehicle velocity v. Using the category 3 and 4 as an example, the reconstructed 3D coordinates can be obtained by the steps of the Section 4.3.2. Compared with the 3D vehicle models, the two categories should be merged as shown in Table 7. Moreover, we defined the clustering errors as CE,

$$\mathrm{CE} = \frac{|N_{\mathrm{detection}} - N_{\mathrm{inspection}}|}{N_{\mathrm{inspection}}}, \tag{17}$$

where $N_{detection}$ is the total number of clustering of the video, $N_{inspection}$ is the total number of vehicles of the video. Experimental results on a number of traffic video sequences show that the results of merging can improved the accuracy of clustering algorithm, as shown in Table 8.

Table 7
Results of inter-class merging.

Label of category	Height	3D coordinate	Merging or not
3[4]	0.0 m	(14.0710,41.7055,0)	Merging
4[6]	1.1614 m	(12.3955,38.9499,1.1614)	

Table 8
Clustering errors of the method.

Traffic scenes	Detection/Inspection	Errors	CE
Outer Ring Road of Shanghai	1539/1599	60	3.75%
Second South Ring Road of Xi'an	1970/1892	78	4.12%
Highway of Hangzhou	1142/1089	53	4.87%

Table 9
Comparison with other method.

Methods	Hashemzadeh et al. (2014)	Wang et al. (2017)	Our method
Detection/Inspection	1268/1599	1766/1599	1539/1599
Errors	331	167	60
CE	20.70%	10.44%	3.75%
Computational cost	2.01 s	1.83 s	2.46 s

5.2. Comparison with other method

In order to show the improvements achieved by the proposed clustering method, it is compared with other methods as given in Table 9. These methods are applied in the traffic video of the Outer Ring Road of Shanghai. In addition, Hashemzadeh et al. (2014) clustered the moving objects by using motion statistics of moving feature points under the assumption that features from a single object will have the same motion vectors. In Hashemzadeh et al. (2014), the moving objects are pedestrians of which the features have different motion states, and only the torso-feature-points are used to calculate the motion vector. In this paper, we simplify its approach while the vehicle objects are rigid objects. In Wang et al. (2017), it computes the correlations of point pairs to obtain a correlation matrix and uses the correlation matrix to construct a sparse affinity matrix. Then, it uses spectral clustering to segment different motions via both the sparse affinity matrix and the estimated number of motions. However, the method of Wang et al. (2017) and Hashemzadeh et al. (2014) are both used in 2D image plane, which cannot handle the perspective effects encountered at low camera angles. The motion vector is not enough for the vehicle segmentation in complex traffic scene. In this paper, we combine the idea of the former two papers and use the RMC in 3D space to segment the moving vehicles. It can be seen that the proposed method has good robustness in complex traffic scenes.

6. Conclusion

In this paper, a novel method, which uses the rigid motion constraints as a measure, is proposed for vehicle trajectory clustering in traffic scenes. It is able to deal with the practical problem in vehicle motion segmentation effectively, such as object occlusion and the

unknown clustering number. RMC is analyzed for the trajectory points of the same rigid object and different rigid objects. More specifically, we make it possible for using the RMC as a measure for the similar trajectory in 3D space. Then, we define a similarity function for the affinity matrix. After that, spectral clustering is applied on the affinity matrix to achieve the trajectory initial clustering. In addition, we use inter-class merging steps to optimize the clustering results. Finally, we apply the proposed method in different traffic videos to analyze the experiment results. Moreover, the proposed method is compared with several other methods in the same poor traffic video. The experimental results show that the proposed method has the good performance and achieves considerable accuracy.

Acknowledgment

This work is supported by the National Natural Science Fund of China (No. 61572083), the Natural Science Foundation of Shaanxi Province (No. 2015JZ018, No. 2017JQ6064) and the Fundamental Research Funds for the Central Universities of China (No. 310824161006, No. 310824171003).

References

Atev, S., Miller, G., & Papanikolopoulos, N. (2010). Clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems, 11*(3), 647–657.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *Proceedings of AAAI-94 workshop on knowledge discovery in databases,* 359–370.

Besse, P. C., Guillouet, B., Loubes, J.-M., & Royer, F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems, 17*(11), 3306–3317.

Chen, L., & Ng, R. T. (2004). On the marriage of lp-norms and edit distance. *Proceedings of 30th int. conf. very large data bases,* 792–803.

Chen, L., Özsu, M. T., & Oria, V. (2005). Robust and fast similarity search for moving object trajectories. *Proceedings of ACM SIGMOD int. conf. management data,* 491–502.

Coates, A., & Ng, A. Y. (2012). Learning feature representations with k-means. *Neural networks: Tricks of the trade,* 561–580.

Dragon, R., Rosenhahn, B., & Ostermann, J. (2012). Multi-scale clustering of frame-to-frame correspondences for motion segmentation. *Proceedings of Eur. Conf. Comput. Vis.* 445–458.

Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(11), 2765–2781.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of 2nd international conference on knowledge discovery in databases and data mining,* 226–231.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clutering. *Machine Learning, 2*, 139–172.

Fréchet, M. M. (1906). Sur quelques points du calcul fonctionnel. *Rendicontidel Circolo Matematico di Palermo (1884–1940), 22*(1), 1–72.

Guha, S., Rastogi, R., & Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. *Proceedings of 15th international conference on data engineering,* 512–521.

Guo, Z. L. J., Cheong, L.-F., & Zhou, S. Z. (2013). Perspective motion segmentation via collaborative clustering. *Proceedings of IEEE Int. Conf. Comput. Vis.* 1369–1376.

Han, B., Liu, L., & Omiecinski, E. (2015). Road-network aware trajectory clustering: Integrating locality, flow, and density. *IEEE Transactions on Mobile Computing, 14*(2), 416–429.

Hashemzadeh, M., Pan, G., & Yao, M. (2014). Counting moving people in crowds using motion statistics of feature-points. *Multimedia Tools and Applications, 72*(1), 453–487.

Hausdorff, F. (1914). *Grundz uge der mengenlehre.* Germany: Springer.

Johnson, N., & Hogg, D. C. (1995). *Learning the distribution of object trajectories for event recognition. BMVC.*

Jung, C. R., Hennemann, L., & Musse, S. R. (2008). Event detection using trajectory clustering and 4-d histograms. *IEEE Transactions on Circuits and Systems for Video Technology, 18*, 1565–1575.

Kanhere, N. K., & Birchfield, S. T. (2008). Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transactions on Intelligent Transportation Systems, 9*(1), 148–160.

Li, C.-G., & Vidal, R. (2014). Rigid motion segmentation using randomized voting. *Proceedings of IEEE Conf. Comput. Vis. Pattern Recog.* 1210–1217.

Li, C.-G., & Vidal, R. (2015). Structured sparse subspace clustering: A unified optimization framework. *Proceedings of 2015 IEEE conference on computer vision and pattern recognition (CVPR),* 277–286.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 171–184.

Morris, B. T., & Trivedi, M. M. (2008). Learning, modeling, and classification of vehicle track patterns from live video. *IEEE Transactions on Intelligent Transportation Systems, 9*, 425–437.

Piotto, N., Conci, N., & Natale, F. G. B. D. (2009). Syntactic matching of trajectories for ambient intelligence applications. *IEEE Transactions on Multimedia, 11*, 1266–1275.

Saleemi, I., Shafique, K., & Shah, M. (2009). Probabilistic modeling of scene dynamics for applications in visual surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*, 1472–1485.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). Wavecluster: A multi-resolution clustering approach for very large spatial databases. *Proceedings of 24th annual international conference on very large data bases,* 428–439.

Sivaraman, S., & Trivedi, M. (2013). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems, 14*(4), 1773–1795.

Song, H., Lu, S., Ma, X., Yang, Y., Liu, X., & Zhang, P. (2014). Vehicle behavior analysis using target motion trajectories. *IEEE Transactions on Vehicular Technology, 63*(8), 3580–3591.

Veeraraghavan, H., & Papanikolopoulos, N. (2009). Learning to recognize video-based spatiotemporal events. *IEEE Transactions on Intelligent Transportation Systems, 10*, 628–638.

Vlachos, M., Gunopulos, D., & Kollios, G. (2002). Discovering similar multidimensional trajectories. *Proceedings of IEEE 18th international conference on data engineering,* 673–684.

Wang, T. L. H., Yan, Y., Chin, T.-J., & Zhao, W.-L. (2017). Motion segmentation via a sparsity constraint. *IEEE Transactions on Intelligent Transportation Systems, 18*(4), 973–983.

Zheng, Y., & Peng, S. (2014). A practical roadside camera calibration method based on least squares optimization. *IEEE Transactions on Intelligent Transportation Systems, 15*(2), 831–843.