



Vehicle behavior analysis using reconstructed 3D parameters for road safety

Xuan Wang ^{a,*}, Jindong Xu ^a, Yongchao Song ^a, Qiang Zheng ^a, Jun Lv ^a, Weiqing Yan ^a, Qing Cai ^b, Zhe Dai ^c

^a Yantai University, Yantai 264005, China

^b The Chinese University of Hong Kong, Shenzhen 518172, China

^c Chang'an University, Xi'an, 710064, China



ARTICLE INFO

Keywords:

Vehicle segmentation

Feature point detection

Trajectory clustering

Vehicle behavior analysis

ABSTRACT

Road traffic safety is a very important issue in the field of intelligent transportation system (ITS). Vehicle segmentation and behavior analysis are an significant part for solving this problem. However, 2D image detection technology is difficult to reconstruct missing information of damaged image. In this paper, a bottom-up analysis method is employed to study the related technical problems, and it will provide a strong data foundation for road traffic safety. Firstly, the M-BRISK descriptor algorithm is proposed to describe the local feature points. Secondly, we propose a 3D feature analysis method based on rigid motion constraints for vehicle trajectory. Thirdly, a similarity measure method is proposed for trajectory clustering. Finally, we used the obtained 3D information of vehicles to analyze the vehicle behavior to find the abnormal vehicles for road traffic safety. The experimental results confirm that the M-BRISK descriptor performs well comparing with the state-of-the-art feature descriptors, and the proposed clustering method improves the accuracy of the trajectory clustering. Moreover, the vehicle motion information contained in the trajectory data can be analyzed to recognize vehicle behavior. The presented work in this paper provides an important foundation for vehicle abnormal behavior detection for road traffic safety.

1. Introduction

Vehicle motion segmentation and vehicle behavior analysis are important research areas in road traffic safety. The increasing coverage of traffic video surveillance results in extensive developments in the key technologies for traffic video analysis. Currently, many major roads have been equipped with monitoring devices, and video surveillance has become the most direct and effective way for real-time monitoring of the road traffic safety (see, Fig. 1). With the development of computer hardware devices, many video analysis methods are studied to improve the safety performance of road traffic, e.g., Alonso et al. (2019), Ali et al. (2019), Dai et al. (2019), Wang et al. (2018), Jeon et al. (2009) and Fan et al. (2019).

There are several key technologies in the intelligent development of traffic monitoring systems. These techniques are used to analyze the traffic video. In general, these techniques are based on either of the top-down or bottom-up approaches. The specific processes are shown in Figs. 2 and 3.

The top-down approach has obvious advantages in high-definition video and smooth traffic environments. If the vehicle targets are not occluded, the methods based on this approach often provide high detection and tracking accuracy using an appropriate classifier. They

however have high computational complexity and low operating efficiency hence it might be difficult for them to meet the requirement of real-time processing. In addition, the actual traffic scene is difficult to predict and the mutual occlusion of the vehicle and other various environmental factors greatly affect the robustness of these methods.

The bottom-up approach uses a partial-to-integral analysis method based on the feature point detection of the vehicle target. This approach then gradually completes the object segmentation process even if the vehicle target is partially blocked during the motion. That is because the other local feature points can be still detected to complete the tracking task. After that, the behavior analysis method is required to be performed. Moreover, the detection algorithm based on the feature point is highly efficient and can better fulfill the real-time requirements in practical applications.

In the bottom-up system framework, researchers work on various local features such as edges, corners, parts, and spots. In recent years, the local feature descriptors have been greatly developed. The typical algorithm is SIFT proposed by Lowe (2004) in 2004. The algorithm uses the gradient information around the feature points to describe them and use the image pyramid to solve the scale problem. Therefore,

* Corresponding author.

E-mail address: xuanwang_91@126.com (X. Wang).

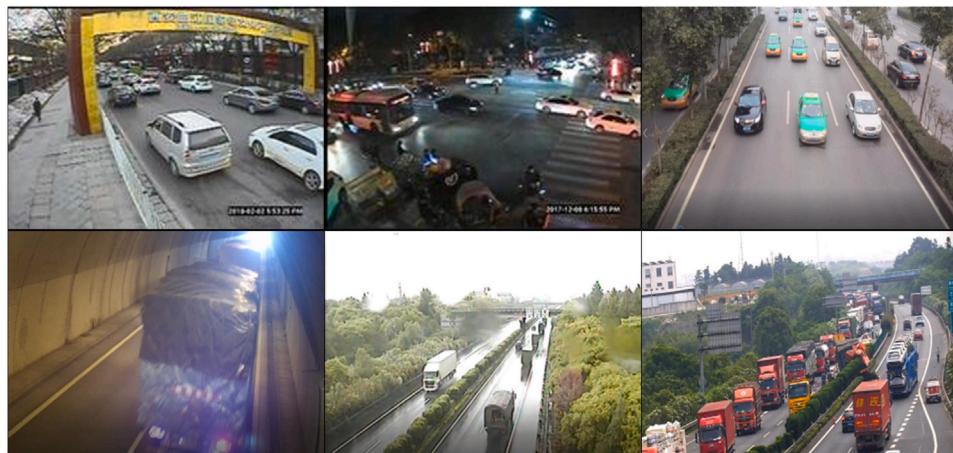


Fig. 1. The traffic video scene.

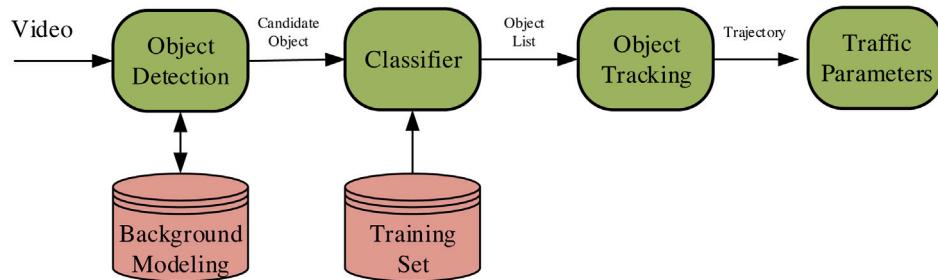


Fig. 2. Top-down traffic video analysis.

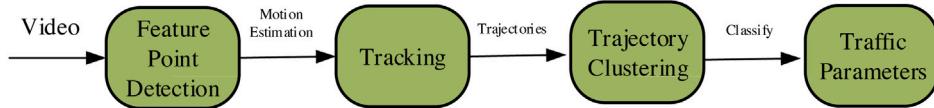


Fig. 3. Bottom-up traffic video analysis.

the SIFT feature descriptors provide good scale invariance and rotation invariance. Improved algorithms, such as PCA-SIFT (Wachs-Lopes et al., 2019), SURF (Bay et al., 2006), PointSIFT (Jiang et al., 2018) were also proposed in the related literature.

After representing some apparent features of the moving objects effectively, it is necessary to use a similarity measure algorithm to perform feature matching on the video sequence and complete the object tracking process (Kamalzadeh et al., 2020). The common similarity measures include Euclidean distance, Gaussian distance, Block distance, Hamming distance, Chessboard distance, Manhattan distance, Weighted distance, Chebyshev distance, Barth Charlie coefficient, Hausdorff distance, etc. In the object tracking process, if we directly search and match the video scene globally to determine its optimal matching position, it will inevitably have to deal with a lot of redundant information. This greatly increases the computing time and reduces the speed of the algorithms. Therefore, it is of great significance to use a specific search algorithm to estimate and calculate the position of the objects in the next moment to narrow down the searching scope. A common approach is to predict the location of the moving object in the next frame and find the best matching position in the vicinity area using, e.g., Kalman filtering (H. Ali and M. Hassan, 2014), extended Kalman filtering (Liu et al., 2020), and particle filtering (Karimian et al., 2020). Another approach is to continuously optimize the search direction to speed up the search and matching, see, e.g., Iswanto et al. (2019) and Wang et al. (2019).

Behavioral understanding of the moving object can be achieved through the trajectory pattern analysis. In the process of discriminating

the trajectory mode, trajectory feature extraction and learning method of trajectory pattern are two essential steps that affect the realization of the trajectory behavior recognition. In terms of trajectory feature selection, many methods perform equal-dimensional processing on the vehicle trajectories to solve the classification problem, such as Zhao (2019) and Ghamdi and Gotoh (2020). To learn the motion trajectory and behavior pattern of the target, researchers also have also put in a lot of effort to realize the detection and discrimination of the abnormal events, such as Zhu (2019) and Mozaffari et al. (2019).

The bottom-up video analysis method can solve the problem of vehicle segmentation in the complex traffic scenes, and it has higher operational efficiency. In this paper we take the local feature points of the image as the research object, using the tracking matching algorithm to obtain the 2D motion trajectory of the vehicle feature points. We then analyze the clustering problem between the trajectories based on the rigid motion constraint. The main contributions of this paper are:

- We propose a feature extraction algorithm based on Binary Robust Invariant Scalable Keypoints (BRISK) (Leutenegger et al., 2011) for complex traffic scenes. For feature detection, we use adaptive FAST algorithm to detect the feature points in the scale space. For feature description, we then construct a hybrid binary feature descriptor based on BRISK. The proposed method guarantees the calculation rate while extracting and locating the feature points effectively.
- We propose a 3D feature analysis method of vehicle trajectory based on rigid motion constraints. Camera calibration is used

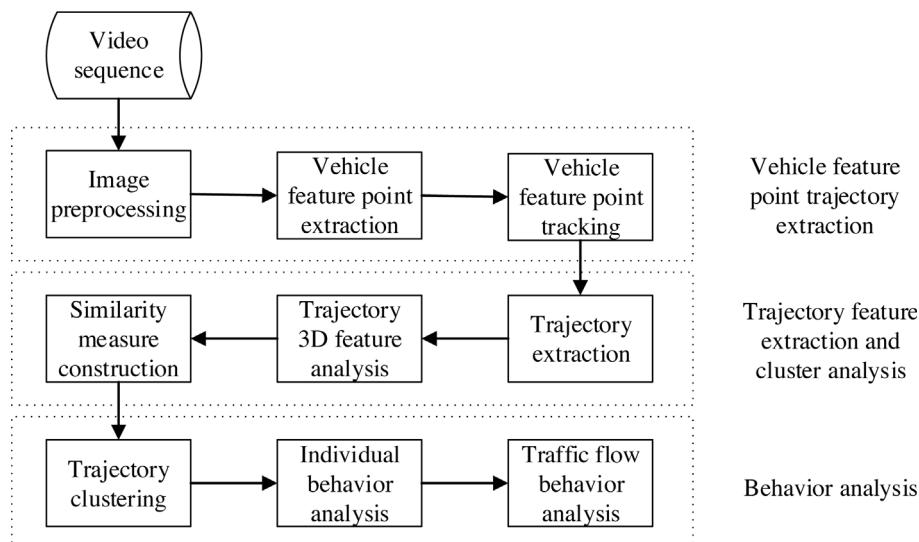


Fig. 4. Overview of the system framework.

to build the back projection data of the image in 3D space. Combined with the idea of back projection, the relative height between different trajectories is obtained based on the rigid motion constraint. The estimated values of relevant traffic information of feature points corresponding to each trajectory in 3D space are further obtained.

- We use the extracted 3D information estimation of feature point trajectories to construct a new similarity measure between trajectories. This measure is then applied to the framework of spectral clustering algorithm to realize the vehicle feature point trajectory clustering in the 3D space.
- Based on the 3D information of vehicle feature point trajectories and its clustering results, the behavior model and semantic analysis of vehicle trajectories in traffic scene are carried out, and the actual road traffic prevalence is analyzed.

The rest of this paper is organized as follows. An overview of the system is presented in Section 2. Section 3 describes the method of feature extraction. Then 3D feature reconstruction is presented in Section 4, followed by Section 5 which presents the vehicle trajectory clustering method. Vehicle behavior analysis is presented in Section 6. The experimental results are reported in Section 7 and finally in Section 8 we draw the conclusion.

2. Overview of the system

According to the specific research process of vehicle trajectory extraction and behavior analysis, the overall technical framework of this paper is illustrated in Fig. 4. The research in this paper is mainly divided into three aspects: vehicle feature point trajectory extraction, trajectory feature extraction and cluster analysis, and vehicle behavior analysis. The vehicle feature point trajectory extraction is the basis of the latter tasks. It mainly studies the feature point detection and stable tracking of the vehicle target in the video sequence, and then obtains the trajectory data of the vehicle feature point in the 2D image plane.

The feature extraction and cluster analysis of the trajectory are based on the camera calibration of the real traffic scene. The motion characteristics of the vehicle trajectory in 3D space are obtained using rigid motion constraint analysis, and the similarity measure is constructed to realize the clustering segmentation. Vehicle behavior analysis is based on the 3D trajectory to obtain the traffic parameters in the actual traffic scene, using the prior knowledge to semantically express the vehicle trajectory, providing a data foundation for further analysis of the individual vehicle behavior and the traffic flow behavior.

3. Feature extraction

In recent years, several methods were proposed for binary feature description for real-time applications, including BRIEFF (Calonder et al., 2010), ORB (Rublee et al., 2011), BRISK (Leutenegger et al., 2011) and FREAK (Alahi et al., 2012). Extracting high quality features and maintaining low computational costs are however very challenging tasks. To address this issue, this paper proposes a feature point detection algorithm based on the improved BRISK algorithm for complex traffic scenes. For the feature point detection, we uses the adaptive FAST algorithm mask to detect the feature points of the scale space. For feature point description, our proposed method constructs a hybrid binary structure feature descriptor based on BRISK algorithm.

3.1. Feature point detector

FAST (Features from accelerated segment test) (Rosten et al., 2008) is a corner detection algorithm proposed by Edward Rosten and Tom Drummond. The most outstanding advantage of this algorithm is its high computational efficiency which is much higher than that of other mainstream algorithms (such as SIFT, SUSAN, Harris). If the machine learning method is applied to the FAST algorithm, it can show even better results. The FAST corner detection algorithm is often used for video processing due to its speed advantage. The principle of the FAST corner point is: if a pixel point and a specific number of pixels in its surrounding area are located in a different area, the pixel point is called a corner point. In the case of gray scale images, if the gray value of the point is smaller or larger than the gray value of the point in its surrounding area, then the pixel may be a corner, as shown in Fig. 5.

3.2. Feature point descriptor

The FAST algorithm only performs feature point detection, but does not further describe the feature points. Therefore, it cannot apply the feature points to the process of image matching and tracking. Several feature descriptors based on the feature points of FAST detection were proposed including ORB, BRISK, and FREAK. Based on the BRISK algorithm, a hybrid binary descriptor is proposed in this paper. The image pyramid is constructed by using the Brisk algorithm in the scale space. Then, the information of the local down sampling. This method improve the robustness of the original BRISK.

The sampling mode of BRISK is shown in Fig. 6. It can be found that the BRISK algorithm only considers the intensity relationship

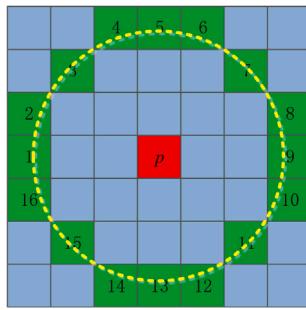


Fig. 5. The principle diagram of the FAST algorithm.

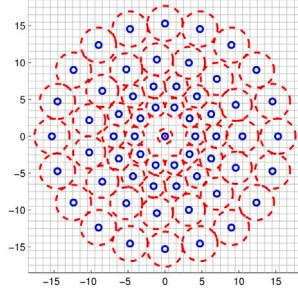


Fig. 6. Sampling mode of BRISK.

between the sampling points, i.e., only the pairwise intensity comparison between the sampling point positions is considered. The local information of the sampling point is therefore lost, which makes the algorithm unstable. To address this issue, the basic idea of this paper is to construct feature descriptors based on the local information of the sampling points and the information between the pairs of sampling points to improve the robustness of the original BRISK.

Let Π be the set of all N sample point positions. For each sample position $\mathbf{p}_i^\alpha = (x_i, y_i) \in \mathbf{P}$, the uniform sampling of four points $S(\mathbf{p}_i^\alpha) = \{s_{i,k}^\alpha, k = 1, 2, 3, 4\}$ is performed on a circle of radius R centered on \mathbf{p}_i^α , where α is the local main direction. This paper uses the Intensity Centroid algorithm (Zhao et al., 2020) to calculate the main direction of the feature points. According to the LBP operator (Liu and Huo, 2019), local information can be encoded by the gray relationship between the sampling position \mathbf{p}_i^α and each local sampling point $s_{i,k}^\alpha$. This encoding is however sensitive to the center point \mathbf{p}_i^α . Therefore it is not applied to binary descriptions. To robustly encode the local information, we propose using the gray relation between local sample points $s_{i,k}^\alpha$ for encoding.

Assume that $I(\mathbf{p}_i^\alpha, \sigma)$ is the smoothed gray value of point \mathbf{p}_i^α , and σ is the Gaussian filter variance. For each rotated sample position \mathbf{p}_i^α , we compare the paired gray values of the local sample points, $s_{i,k}^\alpha, s_{i,t}^\alpha \in S(\mathbf{p}_i^\alpha)$. A local gradient binary descriptor is then constructed by combining all the test results into a binary string, where bit, b , is defined as:

$$b = \begin{cases} 1, & I(s_{i,k}^\alpha, \sigma_i) > I(s_{i,t}^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases} \quad \forall \mathbf{p}_i^\alpha \in \mathbf{P} \wedge s_{i,k}^\alpha, s_{i,t}^\alpha \in S(\mathbf{p}_i^\alpha) \quad (1)$$

Since the local sampling position of each sample point has four values, the dimension of this feature descriptor is $N \times C_4^2 = 6N$ bits. We also note that the gray scale comparison between the local sample points $s_{i,k}^\alpha$ is closely related to the local gradient operator, because they both consider the gray difference between the local sample pairs.

The feature descriptor of the above construction encodes the local information of the sampling points as a binary string. We further supplement it with the global information of the sample points which is encoded by the gray intensity comparison between the sample points.

The set \mathbf{A} represents all the combined results of the sample point pairs:

$$\mathbf{A} = \{(\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) | \mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha \in \mathbf{P} \wedge i \neq j\} \quad (2)$$

Furthermore, a subset \mathbf{B} including M pairs of sample points is selected from \mathbf{A} , so that each bit b of the binary descriptor is constructed by:

$$b = \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases} \quad \forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathbf{B} \quad (3)$$

Here, we construct the feature descriptor consistent with the original BRISK. The short-range pairs of the sample points are used to construct the feature descriptor. The difference is that the M -sample point pairs of the shortest distance are supplementary parts of the previous local gradient-based binary feature descriptor. The mixed BRISK descriptor (M-BRISK) is then constructed following the above two steps of binary string.

4. The 3D feature reconstruction

4.1. Inverse projection transformation

In the process of imaging in the camera the 3D space information is lost and this process is irreversible. At the present, most of the methods for image detection, tracking and behavior analysis are based on 2D image plane. However, due to the perspective transformation of the camera imaging, the geometrical and motion characteristics of the objects no longer exist in the 2D image plane. For example, some geometric features such as symmetry, parallelism, vertical and circular are changed due to perspective projection transformation. The same moving object has obvious scale changes at different positions in the video sequence; and vehicles with uniform motion in 3D world space have non-uniform motion in the 2D video sequence. All of the above situations results in issues with the related algorithms based on 2D image. To solve the segmentation problem of vehicles in complex traffic scenes, it is necessary to extract the 3D information of the vehicle. This paper proposes a method based on rigid motion constraints for vehicle 3D trajectory feature analysis.

Camera calibration is one of the important steps for obtaining the 3D parameters of the objects based on video/image. However, the trajectory clustering and behavior analysis under the monocular camera is based on the vehicle feature point trajectories in 3D space. It is an important precondition for the subsequent algorithm to obtain the transformation between the 2D image and the 3D space. The working process of the camera model is:

$$\lambda p = K [R \ T] P_W = H P_W \quad (4)$$

where, $H = K [R \ T]$, $p = [u, v, 1]^T$, $P_W = [X_W, Y_W, Z_W, 1]^T$, λ is the scale factor, 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 (Amarante and Fujarra, 2020).

Camera imaging is a perspective projection process from 3D space to 2D image. Conversely, the transformation process of mapping 2D image to 3D space is called inverse perspective mapping (IPM). To obtain the reconstructed images with perspective effects through back-projection transformation, the existing constraints and prior knowledge are used to make certain reasoning and estimation (Huansheng et al., 2018). A common method is to first use the transformation relationship and the constraints to achieve the location mapping, and then fill the data. To describe this inverse transformation process, the mapping process is marked as:

$$p_I = F \cdot P_W \quad (5)$$

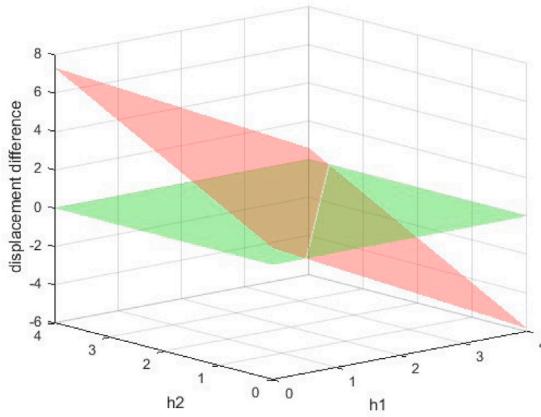


Fig. 7. The rigid motion constraint.

where F represents a transformation matrix of the 3D world coordinate system and the 2D image coordinate system. If a certain image coordinate p_I and one of its coordinate value in the 3D world coordinate system are known, (e.g., the actual height Z_W of the coordinate point), the specific position P_W in the 3D world which corresponding to p_I can be obtained. The process is expressed as:

$$P_W = F^{-1} \cdot (P_I \oplus Z_W) \quad (6)$$

If the pixel points in the 2D image plane are directly back-projected into the 3D world coordinate system, a unique solution cannot be obtained due to uncertain scale parameter. However, if a certain dimensional coordinate parameter in the 3D space is determined, the 3D coordinate corresponding to the 2D pixel coordinate can be uniquely obtained. Therefore, we can preset a back-projection plane in 3D space, to determine the information of a certain dimension so that the data of inverse projection transformation can be obtained on the back-projection plane.

4.2. The rigid motion constraint

This section simulates the motion trajectories of the feature points on a rigid body in 3D space. They are then back-projected onto several back-projection planes paralleling to the road surface. In the actual scene, the height range of the vehicles are constrained. Generally, the height of vehicles is no more than 4 m. Therefore, the 3D information of the vehicle trajectory points can be indirectly obtained. Specifically, we use the enumeration method to test the trajectory height information to reconstruct the trajectory information in 3D space and then use the rigid motion constraints to calculate the height relationship between different reconstructed 3D trajectories.

$$D(P_{fi}^{h_1}, P_{fj}^{h_2}) = D(P_{fi}^{h_1}, P_{li}^{h_1}) - D(P_{fj}^{h_2} - P_{lj}^{h_2}) \quad h_1, h_2 = 0, \dots, 4 \quad (7)$$

where, $D(P_{fi}^{h_1}, P_{li}^{h_1})$ is the displacement of the feature point p_i during F frame in the 3D space, $P_{fi}^{h_1}$ is reconstructed 3D trajectory from the 2D trajectory point p_{fi} with the height information h_1 , and $D(P_{fj}^{h_2}, P_{lj}^{h_2})$ is the displacement difference of the two reconstructed 3D trajectories. Fig. 7 shows the relationship between the displacement differences of two trajectories in an ideal case (without tracking error) using the height enumeration method. It is seen that the heights of the two trajectories and their displacement are on the same plane:

$$ah_1 + bh_2 + c = Diff \quad (8)$$

If the two trajectories belong to the same car, the displacement value is the same, i.e., $Diff = 0$. Therefore, the height relationship of the two trajectories is:

$$ah_1 + bh_2 + c = 0 \quad (9)$$

4.3. The 3D information reconstruction

Based on the idea of back-projection transformation, we set the vehicle moving direction (i.e, the road direction) as the Y_W direction, and constructed multiple back-projection planes parallel to the Y direction. We then reconstruct the 3D trajectory information from the 2D trajectories with the known enumeration value of Z_W . According to (4), (X_W, Y_W) can be calculated as:

$$\begin{aligned} Y_W &= \frac{A - B(H_{31}u - H_{11})}{(H_{32}u - H_{12})(H_{31}v - H_{21}) - (H_{32}v - H_{22})(H_{31}u - H_{11})} \\ X_W &= \frac{A - Y_W(H_{32}u - H_{12})(H_{31}v - H_{21})}{(H_{31}u - H_{11})(H_{31}u - H_{21})} \end{aligned} \quad (10)$$

where, $A = (Z_W H_{13} + H_{14} - v(Z_W H_{33} + H_{34}))(H_{31}v - H_{21})$, and $B = Z_W H_{23} + H_{24} - v(Z_W H_{33} + H_{34})$.

Thereby, it is possible to recover the 3D trajectory of the vehicle target at different height planes. As shown in Fig. 8(a), we simulate the trajectory of the same vehicle in an ideal state. The 2D image projection in the calibration scene is shown in Fig. 8(b). It is seen that the feature point trajectories of different heights have different pixel displacement and pixel speed. Therefore, the 3D trajectory can be estimated by constructing the different projection planes of different heights as shown in Fig. 8(c). Using these trajectory information and the motion characteristics of the rigid objects, we then analyze the real position information and a series of 3D features of vehicles in 3D space, e.g., the actual speed, acceleration, displacement, and driving direction.

The algorithm for 3D feature extraction of vehicle trajectories in traffic scenes is therefor as the following:

- Calibrate the camera in the traffic scene and obtain the transformation matrix H ;
- Use the enumeration method to set the back projection planes with different heights in the range of 0–4 m as shown in Fig. 9, and back-project the 2D trajectories on to the different back projection planes;
- Calculate the velocity $V(i, h)$ of each feature point trajectory on different inverse projection planes, where i represents the i th trajectory and h represents the height of the back projection plane;
- Use the K-mean algorithm for clustering, and classify the trajectory data by velocity difference;
- Obtain the histogram statistics of $V(i, h)$ in each set of trajectories, and use the velocity interval with the highest frequency as an estimated value of the real velocity in 3D space;
- Calculate the height information of each cluster by using the spatial relationship and the estimated velocity;
- Reconstruct the position information of the feature points in the 3D space by combining the height information of each cluster.

5. Trajectory clustering

5.1. Clustering algorithm

The similarity measure is an important basis for data mining techniques such as data classification, clustering and abnormal behavior recognition. We construct a new similar measure relationship between the trajectories using 3D information of feature point trajectories and applies it to the spectral clustering algorithm.

The implementation steps of the algorithm are as follows:

- Construct a similarity matrix W between the trajectories according to the similarity measure between the trajectory data;
- Calculate its normalized Laplacian matrix $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$;
- Calculate the eigenvalues $\{\lambda_i, i = 1, 2, \dots, n\}$ and eigenvectors $\{E_i, i = 1, 2, \dots, n\}$ of L ;
- Obtain an indication feature vector Q_i corresponding to E_i ;
- Perform the K-means clustering using the feature vectors corresponding to the first k minimum eigenvalues of Q .

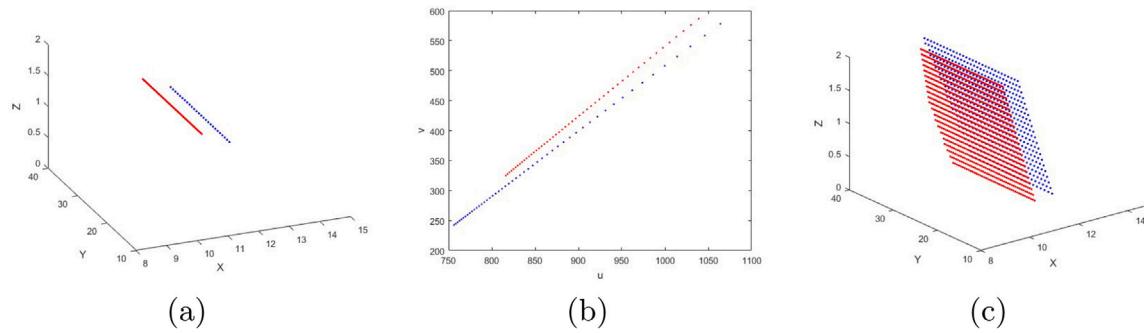


Fig. 8. 3D information reconstruction.

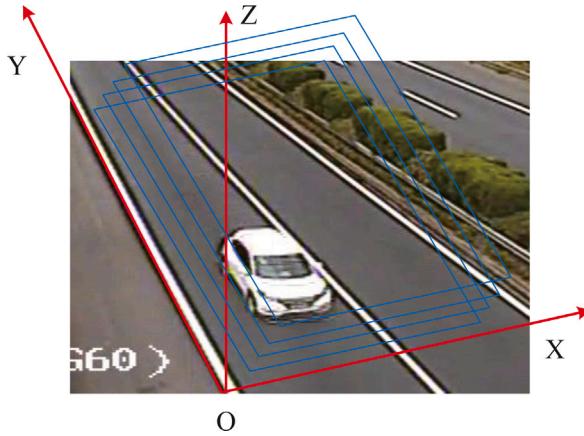


Fig. 9. The setting of the back projection planes.

5.2. Similarity measure

According to the extracted 3D trajectory features, we construct an attribute feature vector $F = (H, V, X, Y)$ that can represent each trajectory information, where H represents the relative height between the trajectory and the reference trajectory, V indicates the 3D velocity of the trajectory reconstructed by the trajectory set T , and (X, Y) represents the 3D coordinate of the trajectory point at a certain time. Furthermore, F covers the feature information inherent of each trajectory and also the relative positional relationship between the trajectories. Therefore, we use the trajectory set T to extract the eigenvector F corresponding to each trajectory. This is then combined with the Gaussian similarity calculation model to construct a new similarity measure S as the following:

$$S(T_i, T_j) = \exp\left(-\frac{d^2(F_{T_i}, F_{T_j})}{2\sigma^2}\right) \quad (11)$$

where, $d(F_{T_i}, F_{T_j})$ is the Euclidean distance of the attribute feature vector extracted by any two trajectories in the trajectory set T , F_{T_i} is a 1×4 feature vector which includes four parameters of the trajectory, and σ is the scale factor.

Since the research object in this paper is a vehicle, the distribution of feature points on a vehicle is limited. It means that X coordinate range of the feature point trajectory in a vehicle cannot exceed the width of the vehicle. Using this property, the similarity matrix W between the vehicle trajectory sets is defined as:

$$W(T_i, T_j) = \begin{cases} \exp\left(-\frac{d^2(F_{T_i}, F_{T_j})}{2\sigma^2}\right) & d(X_{T_i^f}, X_{T_j^f}) \leq \xi \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where, ξ is a threshold parameter according to the actual outer contour size standard of the road vehicle. Since the reconstructed trajectory 3D

information is an estimated value, the estimated value may include an estimation error compared with the real value.

For a given set of trajectories, the construction process of the similar matrix is as follows:

- Calculate the 2D velocity $v = \{v_1, v_2, \dots, v_n\}$ for each 2D trajectory of the trajectory set. Select the trajectory with the minimum 2D velocity v_p as the reference trajectory T_p ;
- Calculate the relative height between each trajectory and the reference trajectory.
- Use the enumeration method to construct different heights of the back projection planes in the range of 0–4 m to recover each trajectory in 3D space;
- Calculate an estimated velocity of each trajectory in the 3D space and the spatial position of the feature point at the current frame;
- Construct the attribute feature vector $F_{T_i} = (H_i, V_i, X_i, Y_i)$ of each trajectory;
- Calculate the similarity matrix W between the trajectory data set T using (12);

6. Vehicle behavior analysis

6.1. Vehicle individual behavior analysis

In this section, the 3D information of the vehicle trajectory is used to further analyze the behavioral pattern of the individual vehicle in the traffic scene to detect abnormal behaviors. Many related threshold information are contained in this section and for a determined traffic scene, the associated threshold information is the same. The threshold information depend on the theoretical calculations and the empirical values.

6.1.1. Over-speed and low-speed driving

According to China's Road Traffic Safety Law, the highway sections should identify the limits of their driving speed clearly. For example, the maximum speed of vehicles on the highway cannot exceed 120 km/h, and the minimum speed cannot be lower than 60 km/h.

If $V_i > V_\alpha$, the vehicle is judged to be over-speed; if $V_i < V_\beta$, the vehicle is determined to be low-speed, where V_i is the estimated value of the real speed of the i th vehicle, V_α and V_β are the maximum speed, and the minimum speed of the road section, respectively.

6.1.2. Retrograde

The camera has a fixed installation position and angle in the traffic scene. Firstly, we manually determine the adjust driving direction of the road based on the driving direction of the vehicle in the traffic video. Then, we use camera calibration technology to obtain the 3D position information of the direction marking line and its direction vector. As shown in Fig. 10, a direction vector can be set in the two-lane road section, and two or more correct direction vectors should be set according to the actual situation.

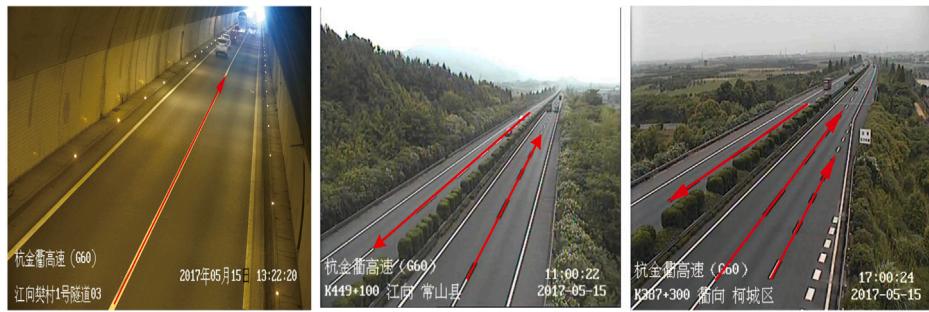


Fig. 10. Setting of the correct driving direction.

According to the 3D trajectory information of the vehicle feature point, we determine the motion vector (X_i^f, Y_i^f) of each frame. The X direction information is then used to select the correct driving direction of the road for retrograde event discrimination. If (X_R, Y_R) is the correct direction of the road, the direction angle of the vehicle is:

$$\theta = \frac{1}{m} \sum_{i=0}^m \arccos\left(\frac{X_R X_i^f + Y_R Y_i^f}{\sqrt{X_R^2 + Y_R^2} \sqrt{(X_i^f)^2 + (Y_i^f)^2}}\right), \quad (13)$$

$$IsRetrograde = \begin{cases} true & \bar{\theta} \geq \alpha \\ false & \bar{\theta} < \alpha \end{cases} \quad (14)$$

where, (X_i^f, Y_i^f) is the direction of motion of the i th trajectory in the same category in frame f , and α is the empirical threshold. In practice, the fault tolerance of the algorithm needs to be considered. The vehicle behavior cannot be solely judged according to the data at a certain moment. Instead, it should be counted whether the direction angle of the vehicle motion satisfies the retrograde condition for a period of time. This paper counts the number l that the vehicle direction angle is greater than the empirical threshold α for a period of time. If $l > \beta$, we then consider the vehicle as a retrograde vehicle.

6.1.3. Parking

In the event of an abnormal parking, the feature point trajectory of the vehicle has obvious characteristics. It is embodied in a state in which the speed of the vehicle is gradually decreased to zero while the position information tends to be constant. Therefore, the discriminating rules are

- If $V_k^f < \xi$, the counter of abnormal speed is incremented by 1.
- If $Isstop > \eta$, the vehicle has an abnormal parking event.

where, V_k^f indicates the instantaneous speed of the k th cluster of the vehicle at the f th frame, ξ is the minimum speed threshold and η is the speed anomaly threshold.

6.1.4. Abnormal lane change

The lane change behavior of the vehicle occurs more frequently in the actual traffic while the road is divided by solid lines (white solid line, yellow solid line, double yellow solid line). These solid lines should not be crossed during the driving.

For a normal driving vehicle, its movement trend is along the direction of the lane line, i.e., its motion trajectory is approximately parallel to the road marking line. However, there is a certain angle between the trajectory of the vehicle in which the lane changing behavior occurs and the road marking line. Therefore, this paper uses the following method to determine the abnormal lane change behavior of the vehicles:

- For a specific traffic road, camera calibration is performed manually to obtain the actual 3D space coordinates of the solid line marker line on the road;

- Calculate the variance in the X direction using the 3D trajectory information of the vehicle feature points in each category:

$$\bar{S} = \frac{1}{mn} \sum_{t=1}^m \sum_{i=1}^n (X_t(i) - \bar{X}_t)^2 \quad (15)$$

- If $\bar{S} > \gamma$, a lane change has happened, and it is necessary to further judge whether the behavior is a violation of the rules. If $|X_t(i) - X_{Road}| < \varepsilon$, the rules are violated.

In the above, $X_t(i)$ represents the X coordinate of the i th point of the t th trajectory in 3D space, \bar{X}_t is the average of the X coordinates of the t th trajectory, X_{Road} indicates the X coordinate of the solid line marker on the road in 3D space and γ, ε are the experience threshold which are determined based on the specific traffic scene.

6.1.5. Traffic flow behavior

In this section, the 3D information of the vehicle motion trajectory and the clustering results are used to calculate the traffic flow and traffic flow speed in a section of the road which can be used to evaluate the real-time traffic status.

6.1.6. Traffic flow

To fully consider the time series of the trajectory points during the motion, previous clustering results are combined to filter the trajectory data of the current frame, so the attribute feature extraction and cluster analysis are only carried out for the newly added trajectory data. It not only reduces computational cost, but also improves the clustering accuracy of the new trajectory data. The proposed strategy is described as the following:

- Set the time interval t of the clustering based on the video rate. That means cluster analysis is performed on the feature point trajectory in the current interest region every interval t frame;
- The trajectory data is filtered twice before each run of the clustering algorithm. One is to screen out the trajectory data with certain length; the other is to filter out the new trajectory data.
- Count the clustering results obtained each time. The traffic flow per hour or day of the road section is then obtained.

6.1.7. Traffic flow speed

In order to facilitate the measurement and calculation, we select the interval average speed as the measurement index of the traffic speed. In the selected observation section, several instantaneous moments are selected in the fixed time intervals and the average value of the instantaneous speeds of all vehicles is calculated by using the 3D information vehicle feature point trajectories. We write:

$$\bar{v}_s = \frac{1}{MN} \sum_{k=1}^N \sum_{i=1}^M \frac{s_k(i)}{\Delta t} \quad (16)$$

where, Δt is the time interval of adjacent frames, $s_k(i)$ is the distance during the time interval between the current frame and the previous frame at the i th feature point of the k th vehicle.



Fig. 11. The test images.

7. Experimental results

In this section, we evaluate the performance of the proposed system. In Section 7.1, the performance analysis of the feature descriptor is evaluated. The trajectory clustering results on different traffic videos are presented in Section 7.2. Section 7.3 performs the practical results of vehicle behavior analysis.

7.1. Performance analysis of feature descriptor

The performance indicators of the feature descriptors are evaluated using the recall and $1 - precision$. We compare the M-BRISK descriptor with the SURF, ORB, BRISK and FREAK descriptors. SURF is a classic fast descriptor, whereas ORB, BRISK and FREAK are recently proposed binary descriptors. For fair comparison, image blocks of the same size (31×31) are set for all test descriptors and different images of Oxford dataset are used for the correlation test. The original picture of the data set is shown in Fig. 11. Each group of images has different changing factors, including fuzzy processing, rotation and scale change, perspective change, illumination change, and image compression. Fig. 12 shows the experimental results of different descriptors for different impact indicator.

For all cases, the M-BRISK descriptor either overperforms or at least comparable with the descriptors of other tests. This is because the discrimination of the descriptor is improved by a combination of the local features and the information between them. As it is seen in Table 1, M-BRISK speed is similar to that of the ORB, BRISK and FREAK algorithms, and they are all much faster than the SURF. In summary, the M-BRISK algorithm achieves higher performance and higher speed, and it is suitable for real-time applications.

7.2. Trajectory clustering results

We collect 1000 sets of vehicle trajectory data from 20 road sections of Hangzhou Jinqu highway for clustering algorithm test, and

Table 1
Running time..

Methods	SURF	ORB	BRISK	FREAK	M-BRISK
Running time (ms)	0.404	0.026	0.038	0.032	0.040

Table 2
Clustering accuracy.

Number of Vehicles	2	3	4	5
3D trajectory clustering accuracy (CP)	94.75%	93.54%	89.17%	87.63%
2D trajectory clustering accuracy (CP)	90.18%	85.21%	78.32%	69.67%

the trajectory datasets contains different numbers of vehicle targets. These scenes are tested on a Windows 10 platform. The image size is 720×288 , and the sampling frequency is 25 FPS. Instances of the experimental clustering trajectory sets are shown in Fig. 13. It is seen that even if the vehicle has a common speed or partial occlusion, the algorithm can effectively cluster the feature point trajectories belonging to different vehicles. We further analyze the clustering results of the 1000 datasets based on the number of vehicle targets, and compare them with the traditional method based on 2D trajectory methods. The results are shown in Table 2.

To evaluate the clustering efficiency of the proposed method, we analyze the relevant experimental results and define the accuracy of the clustering as

$$CP = \frac{1}{N} \sum_{i=1}^N \frac{t_i}{n_i} \times 100\% \quad (17)$$

where, N is the number of trajectory datasets with the same number of vehicles, t_i is the number of trajectory classified correctly for the i th trajectory dataset, n_i is the total number of trajectory included in the i th trajectory dataset.

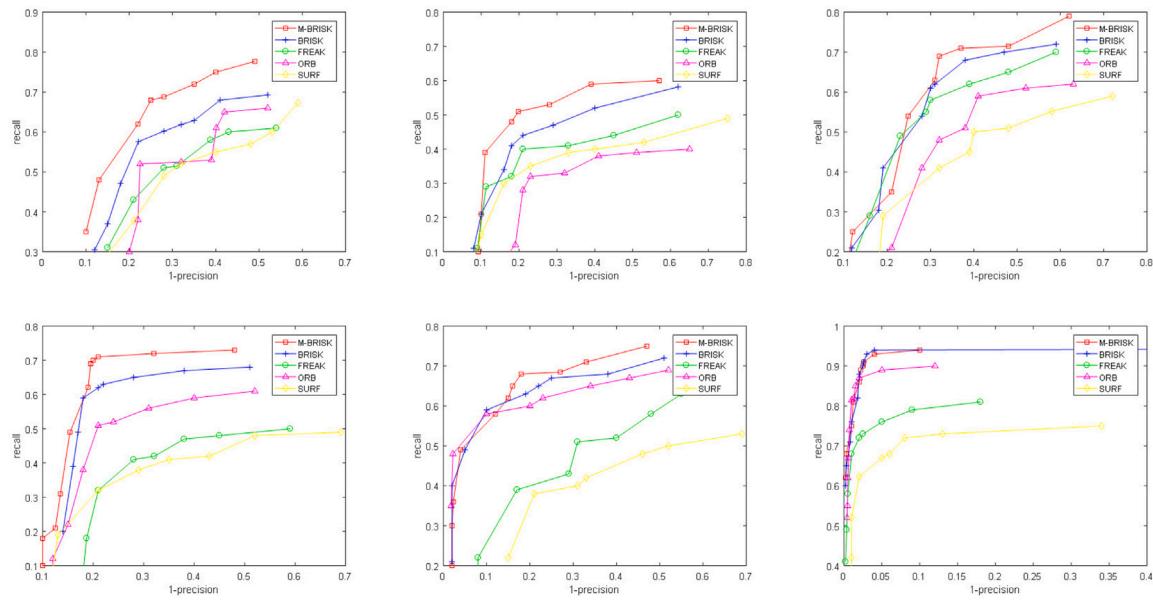


Fig. 12. The experimental results of different descriptors for each set of image pairs.

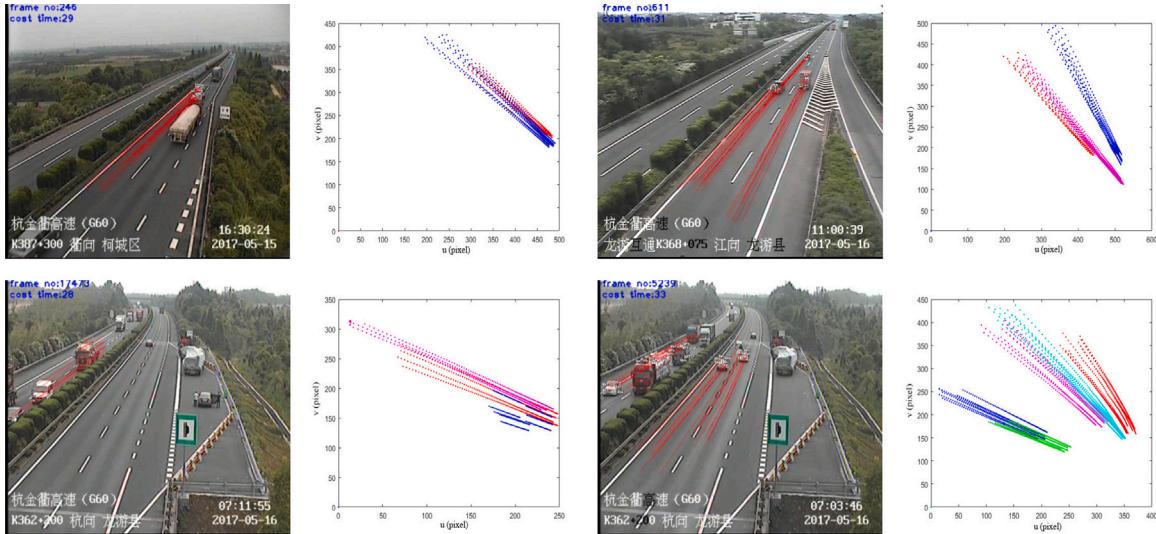


Fig. 13. Vehicle feature point trajectory clustering result.

7.3. Vehicle behavior analysis

7.3.1. Vehicle individual behavior

Reconstructing the 2D trajectory information in the 3D space enables estimation of the real velocity of each trajectory. The velocity of the vehicle objects is estimated in real-time and the velocity curve of the vehicle target is drawn to determine over-speeding, low-speeding or parking events. The following is a specific experimental analysis based on specific trajectory data.

Fig. 14(a) is the 2D trajectory data extracted from the vehicles of a highway section, Fig. 14(b) is the result of cluster analysis, and Fig. 14(c) is the real-time velocity curve. The partial data results of the real-time velocity at the same time are shown in Table 3, wherein the sign indicates the moving direction of the vehicles, the speed of the upstream vehicle is marked as positive, and the speed of the descending vehicle is marked as negative. The estimated real-time velocity of the vehicles can be used as a discriminating indicator to determine whether the vehicle is over-speed or low-speed.

To determine the retrograde behavior, it is necessary to observe its real-time motion vector (see, Table 4). In addition, if an abnormal parking event occurs, the velocity curve of the vehicle is as shown in Fig. 15, where its trajectory velocity continues to approach for a period of time. For the behaviors such as lane change, in addition to the direction angle, the offset in the X direction is also required. As it is seen in the trajectory data of Fig. 16, we can observe the driving direction angle of the real-time. It is seen in Table 5 that the direction angle of the vehicle is increased and the variance of the corresponding trajectory data in the X direction is also larger than the preset. Therefore, the vehicle in Fig. 16 changed lane abnormally.

7.3.2. Traffic flow behavior analysis

here we use the real-time data obtained by the proposed method to analyze the traffic flow and traffic flow's speed in a highway section. Such data is essential for real-time management and monitoring applications. We use the monitoring video of Jinqu highway as the test data and analyze the traffic flow and traffic flow speed of the K362 road sections at 30-minute intervals from 6:30 to 18:00 on May 16, 2017.

Table 3
Vehicle real-time velocity.

Objects	Real-time velocity (m/s)										
Vehicle 1	-33.2	-33.0	-32.9	-32.6	-32.9	-33.2	32.3	-32.1	-32.2	-32.0	-31.7
Vehicle 2	27.9	28.5	27.6	27.8	27.1	27.87	28.5	27.8	27.5	27.2	27.8
Vehicle 3	-28.0	-25.4	-26.2	-26.5	-25.7	-24.6	-26.6	-24.1	-26.3	-28.3	-26.1

Table 4
Direction angle of the vehicle in realtime.

Objects	Direction angle (°)										
Vehicle 1	0.038	0.043	0.045	0.578	0.800	0.484	0.029	0.103	0.625	0.182	0.333
Vehicle 2	0.021	0.016	0.078	0.051	0.338	0.110	0.386	0.245	0.311	0.086	0.028
Vehicle 3	0.315	2.383	1.509	4.215	3.121	0.705	2.065	3.174	3.322	1.419	2.373

Table 5
Direction angle of the vehicle in real time.

	Direction angle (°)											Variance
Trajectory 1	5.01	5.49	6.40	4.69	10.57	6.75	9.45	8.51	9.95	10.1	0.54	
Trajectory 2	6.08	6.34	7.28	5.42	8.57	6.91	7.71	7.17	9.01	9.46	0.52	
Trajectory 3	5.24	5.95	5.40	5.07	9.37	6.93	9.11	7.55	9.73	10.41	0.50	
Trajectory 4	5.02	5.75	7.46	5.12	9.28	6.44	9.48	8.76	10.94	0.892	0.55	
Trajectory 5	5.35	6.24	5.99	6.43	6.27	5.23	7.43	6.47	7.92	7.95	0.54	

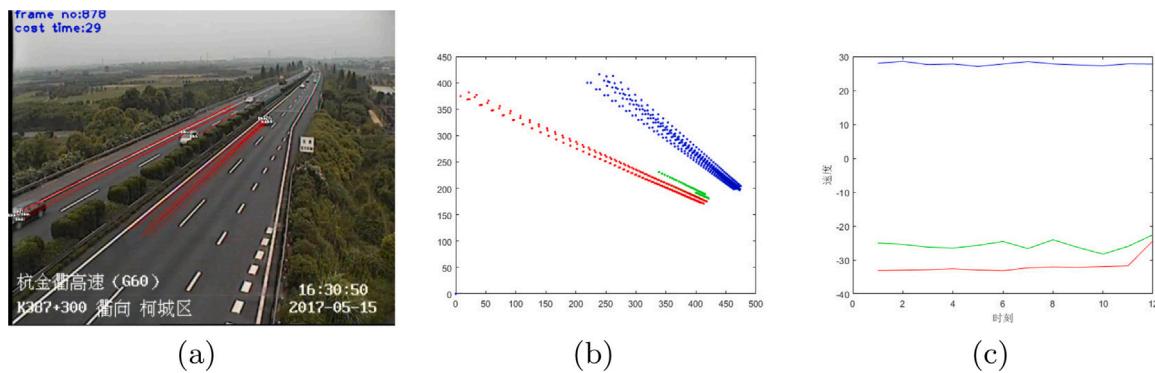


Fig. 14. The real-time vehicle velocity analysis.

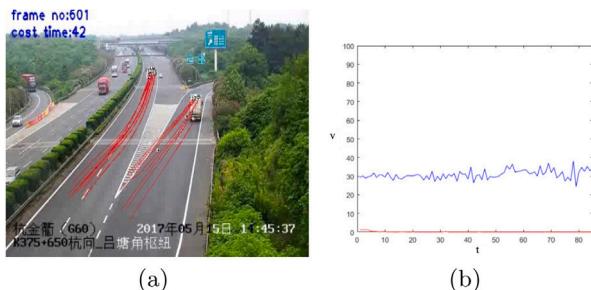


Fig. 15. Vehicle parking analysis.



Fig. 16. Trajectory data of lane change.

According to the obtained real-time traffic flow parameters, we then obtain the real-time parameter curves and observe the time-varying rule of each traffic parameter visually (see, Fig. 17). It is seen that on May 16th, the traffic volume of the K362 section of Jinqu highway was low, and the traffic flow was high in the afternoon. Meanwhile, the traffic flow speed of the whole day is at a reasonable range. Therefore, the traffic condition of this road section is good and smooth.

8. Conclusion

In this paper, we investigated vehicle feature point detection, trajectory extraction, rigid motion constraint, trajectory clustering and vehicle behavior analysis. We then constructed a mixed binary descriptor using the local gradient of the sample point position and the intensity comparison between the sample points. The proposed algorithm is robustness against image blur, rotation, scale, viewing angle and illumination changes, and meets the needs of practical applications in real-time. To improve the solution of the segmentation problem of the moving vehicles in complex traffic scenes, this paper proposed a method based on rigid motion constraints for vehicle 3D trajectory feature analysis, and constructed a new similarity measure between the trajectory sets. The proposed method was then applied to the framework of the spectral clustering algorithm to enable trajectory clustering in 3D space. We further used the obtained 3D information of the vehicle trajectory and its clustering results to analyze the vehicle behavior in the traffic scenes. All the works are to monitor road traffic safety, in order to quickly detect abnormal road traffic incidents, and

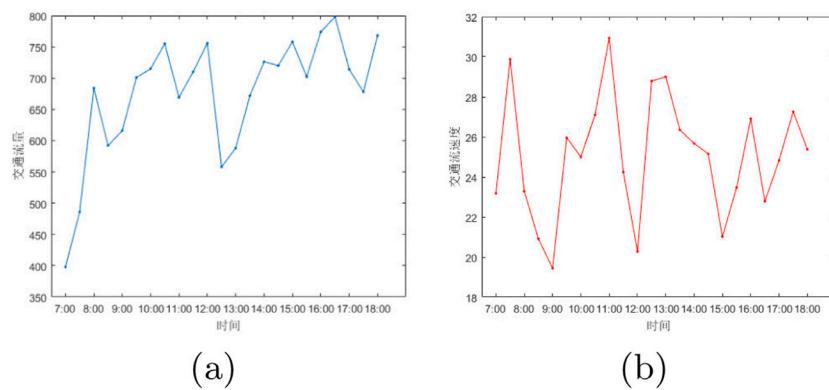


Fig. 17. Traffic flow data of No. 362 section of Hangzhou Jinqu highway on May 16, 2017.

issue early warnings in advance, which can effectively improve road traffic safety.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62072391, No. 61802330, No. 61902338, No. 61801414, No. 62066013), and the Natural Science Foundation of Shandong Province, China (No. ZR2020MF148, No. ZR2020QF108, No. ZR2020QF031, No. ZR2017QF006, No. ZR2019MF060).

References

- Alahi, A., Ortiz, R., Vandergheynst, P., 2012. FREAK: Fast retina keypoint. In: IEEE Conference on Computer Vision and Pattern Recognition.
- Ali, E.M., Ahmed, M.M., Wulff, S.S., 2019. Detection of critical safety events on freeways in clear and rainy weather using SHRP2 naturalistic driving data: Parametric and non-parametric techniques. *Saf. Sci.* 119, 141–149.
- Alonso, M., Mántaras, D.A., Luque, P., 2019. Toward a methodology to assess safety of a vehicle. *Saf. Sci.* 119, 133–140.
- Amarante, R.M., Fujarra, A., 2020. Low-cost experimental apparatus for motion tracking based on image processing and camera calibration techniques. *SN Appl. Sci.* 2 (9).
- Bay, H., Tuytelaars, T., Gool, L.V., 2006. SURF: Speeded up robust features. In: European Conference on Computer Vision.
- Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010. BRIEF: Binary robust independent elementary features. In: European Conference on Computer Vision.
- Dai, Z., Song, H., Wang, X., Fang, Y., Yun, X., Zhang, Z., Li, H., 2019. Video-based vehicle counting framework. *IEEE Access* 7, 64460–64470.
- Fan, Z., Liu, C., Cai, D., Yue, S., 2019. Research on black spot identification of safety in urban traffic accidents based on machine learning method. *Saf. Sci.* 118, 607–616. <http://dx.doi.org/10.1016/j.ssci.2019.05.039>, <https://www.sciencedirect.com/science/article/pii/S0925753519307891>.
- Ghamdi, M.A., Gotoh, Y., 2020. Graph-based topic models for trajectory clustering in crowd videos. *Mach. Vis. Appl.* 31 (5).
- H. Ali, N., M. Hassan, G., 2014. Kalman filter tracking. *Int. J. Comput. Appl.* 89 (9), 15–18.
- Huansheng, Song, Xuan, Wang, Cui, Hua, Weixing, Wang, Qi, Guan, 2018. Vehicle trajectory clustering based on 3D information via a coarse-to-fine strategy. *Soft Comput.: Fusion Found. Methodol. Appl.* 22 (5), 1433–1444.
- Iswanto, I.A., Tan, W.C., Li, B., 2019. Object tracking based on meanshift and particle-Kalman filter algorithm with multi features. *Procedia Comput. Sci.* 157, 521–529.
- Jeon, G., Anisetti, M., Lee, J., Bellandi, V., Jeong, J., 2009. Concept of linguistic variable-based fuzzy ensemble approach: Application to interlaced HDTV sequences. *IEEE Trans. Fuzzy Syst.* 17 (6), 1245–1258.
- Jiang, M., Wu, Y., Zhao, T., Zhao, Z., Lu, C., 2018. PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation.
- Kamalzadeh, H., Ahmadi, A., Mansour, S., 2020. Clustering time-series by a novel slope-based similarity measure considering particle swarm optimization. *Appl. Soft Comput.* 96, 106701.
- Karimian, S.F., Moradi, R., Cofre-Martel, S., Groth, K.M., Modarres, M., 2020. Neural network and particle filtering: A hybrid framework for crack propagation prediction. *arXiv e-prints*.
- Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. BRISK: Binary Robust invariant scalable keypoints. In: International Conference on Computer Vision.
- Liu, M., Gao, S., Li, W., Xie, B., University, S., 2020. State augmentation-based modified iterated extended Kalman filtering. *Appl. Electron. Tech.*
- Liu, X., Huo, Y., 2019. Underground personnel face recognition method based on multi-scale unified LBP operator. *Coal Sci. Technol.*
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60 (2), 91–110.
- Mozaffari, S., Al-Jarrah, O.Y., Dianati, M., Jennings, P., Mouzakitis, A., 2019. Deep learning-based vehicle behaviour prediction for autonomous driving applications: A review. *IEEE Trans. Intell. Transp. Syst.*
- Rosten, E., Porter, R., Drummond, T., 2008. Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (1), 105–119.
- Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R., 2011. ORB: an efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision.
- Wachs-Lopes, G.A., Horvath, M., Giraldo, G.A., Rodrigues, P.S., 2019. A strategy based on non-extensive statistics to improve frame-matching algorithms under large viewpoint changes. *Signal Process., Image Commun.*
- Wang, Z., Duan, N., Fan, L., 2019. Object tracking algorithm fused with YOLO detection and meanshift. *Comput. Eng. Appl.*
- Wang, X., Song, H., Guan, Q., Cui, H., Zhang, Z., Liu, H., 2018. Vehicle motion segmentation using rigid motion constraints in traffic video. *Sustain. Cities Soc.* 42.
- Zhao, S., 2019. Application of a clustering algorithm in sports video image extraction and processing. *J. Supercomput.* (14).
- Zhao, C., Zhu, W., Qiao, G., Zhou, F., 2020. Optimisation method with node selection and centroid algorithm in underwater received signal strength localisation.
- Zhu, L., 2019. Safety detection algorithm in sensor network based on ant colony optimization with improved multiple clustering algorithms. *Saf. Sci.* 118, 96–102.