International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 32, No. 4 (2018) 1855003 (21 pages)

DOI: 10.1142/S0218001418550030

Automatic Measurement of Traffic State Parameters Based on Computer Vision for Intelligent Transportation Surveillance

Jianqiang Ren* and Chunhong Zhang

College of Mathematics and Information Science
Langfang Teachers University
Institute of Pattern Recognition and Intelligent System
Langfang Teachers University
Langfang Key Laboratory of Intelligent Transportation System
Langfang 065000, P. R. China
**renianiang@163.com*

Lingjuan Zhang

College of Physics and Electrical Information
Langfang Teachers University, Langfang 065000, P. R. China
zhanginguan@163.com

Ning Wang† and Yue Feng

College of Mathematics and Information Science
Langfang Teachers University
Institute of Pattern Recognition and Intelligent System
Langfang Teachers University
Langfang Key Laboratory of Intelligent Transportation System
Langfang 065000, P. R. China
†*lfgzdz@163.com*

Received 4 May 2017
Accepted 27 June 2017
Published 6 September 2017

Online automatic measurement of traffic state parameters has important significance for intelligent transportation surveillance. The video-based monitoring technology is widely studied today but the existing methods are not satisfactory at processing speed or accuracy, especially for traffic scenes with traffic congestion or complex road environments. Based on technologies of computer vision and pattern recognition, this paper proposes a novel measurement method that can detect multiple parameters of traffic flow and identify vehicle types from video sequence rapidly and accurately by combining feature points detection with foreground temporal-spatial image (FTSI) analysis. In this method, two virtual detection lines (VDLs) are first set in frame images. During working, vehicular feature points are extracted via the upstream-VDL and

*Corresponding author.

grouped in unit of vehicle based on their movement differences. Then, FTSI is accumulated from video frames via the downstream-VDL, and adhesive blobs of occlusion vehicles in FTSI are separated effectively based on feature point groups and projection histogram of blob pixels. At regular intervals, traffic parameters are calculated via statistical analysis of blobs and vehicles are classified via a K-nearest neighbor (KNN) classifier based on geometrical characteristics of their blobs. For vehicle classification, the distorted blobs of temporary stopped vehicles are corrected accurately based on the vehicular instantaneous speed on the downstream-VDL. Experiments show that the proposed method is efficient and practicable.

Keywords: Computer vision; intelligent transportation monitoring; measurement of traffic parameters; classification of vehicles; K-nearest neighbor classifier; foreground temporal-spatial image.

1. Introduction

Currently, intelligent transportation systems (ITS) have attracted much attention as the efficient way of improving the efficiency of transportation systems, enhancing travel security, and providing more choices to travelers.[1–3] Online automatic measurement of traffic parameters (e.g. flow rate, mean speed and occupancy, etc.) and classification of vehicles have important significance for intelligent transportation surveillance and management.[4,5] Today, a lot of measurement techniques are applied but the video-based methods are more and more widely used in practical engineering as video sequence can supply more plentiful and useful monitoring information than other sensors, such as ground induction loops, bridge sensors, etc.[6–8]

In video based transportation surveillance systems, the cameras mainly include high-altitude cameras installed on aircrafts,[9,10] high-angle cameras[11,12] installed on high buildings and low-angle cameras[13,14] installed on road infrastructures. Comparing with the first two kinds of cameras, low-angle cameras, especially the roadside cameras, have significantly lower installation and maintenance costs and higher maintenance-convenience. The roadside cameras are generally installed on lamp posts or camera backbones at the side of road and this installation method is more economical and practical than placing cameras above the ground at middle of the road. Today, the roadside cameras are widely used in transportation surveillance systems,[14–16] so it has important practical value to study the traffic flow monitoring methods based on the roadside cameras.

The video based detecting methods of traffic parameters mainly include vehicle-tracking-based methods and virtual-lane-based methods.[17] The vehicle-tracking-based methods mainly include optical-flow-based method,[18] frame-difference-based method[19] and background-subtraction-based method.[20] These methods have higher detecting accuracy and can provide a wealth of information, but their processing speed is not satisfactory because the whole (or a big part of) frame image needs to be processed. For virtual-line-based technology, some methods based on frame-information compression have been studied. The methods, which set virtual lanes at crucial places in the frame image and extract traffic parameters, can reduce computational cost obviously. Reference 21 studied a typical video temporal-spatial

accumulation technique. When vehicle goes across the video sequence, their information can be captured by a manual virtual detection lines (VDLs). Therefore, the processing speed is increased by reducing redundant usage of non-vehicle information, but the accuracy of the method is not satisfactory. After that a lot of improved VDLs-based detection methods have been proposed.[22–26] These methods generate temporal-spatial image (TSI) by accumulating luminance value of pixels on the VDL of every frame images in time order and then calculate traffic parameters. For example, method in Ref. 22 counts vehicles based on TSI in urban road environment. Although the interference factors coming from complex environments, traffic congestions and night headlights of vehicle are discussed, VDLs cannot be set crossing lane lines or unwanted shadow regions, otherwise vehicle blobs in the TSI will be merged seriously to each other. Method in Ref. 23 uses two VDLs to calculate the mean speed and vehicle volume with less computer resources and less processing time, but the influence from moving shadows is not considered, so the resulting distortion and occlusion of blobs negatively affect the detection accuracy. Additionally, the above methods cannot recognize the type of vehicles that is important for transport operators.

Recently, some novel methods of TSIs-based vehicle detection and classification have been studied. The methods in Refs. 24–26 are three typical methods. Reference 24 utilizes segmented blobs in a single TSI and some appropriate frames of the original video to extract the features of the moving objects. Further, Ref. 25 employs multiple VDLs to detect traffic volume and classify vehicles. The detection of moving vehicles is based on blobs in the TSIs. Then, based on such blobs, some key vehicular frames (KVFs) are extracted from video sequence. For a KVF, the middle of a vehicle is just on the VDL. Then the features for vehicle classification are extracted from the KVFs. Although the method has higher classification accuracy, the extraction of classification features relies heavily on the original video frames. The extraction, storage and process of the KVFs exacerbate the time complexity and space complexity of the method. Moreover, Refs. 24 and 25 do not take account of the influence of moving shadows on the vehicle detecting accuracy, but these shadows always result in unwanted distortion of object-areas and serious occlusion to each other. In this case, we provided a feasible approach to detect vehicles volume, mean speed and vehicle types based on blobs in two foreground temporal-spatial images (FTSIs) directly.[26] It minimizes the dependence on the original frames and eliminates unwanted interferences of moving shadows. However, the method has some limits as follows: (i) it separates occlusive blobs in FTSIs based on projection histograms in four different directions, which is more likely to make over-segmentation for irregular blob corresponding to single vehicle since there is no information of number of vehicles in a blob; (ii) a blob will be distorted along time axis in FTSI if the vehicle slow drives or temporary stops on the VDL, which seriously influence the pairing of blobs in the two FTSIs, the calculating of vehicular length and the accuracies of vehicles classification; (iii) it still cannot correct the distortion of blobs because they cannot judge whether the vehicle stops.

In this paper, we solve the above problems by combining VDL-based FTSIs with fast extracting and grouping of vehicular feature points further. We extract information of vehicular objects based on computer vision technology, then calculate traffic parameters and recognize vehicle types based on pattern recognition technology (K-nearest neighbor (KNN) classifier). Comparing with the existing technologies, the proposed method has novelty and major advantages as follows: (i) it can be applied to accurately measure multiple parameters of traffic flow state and recognize vehicle types concurrently online via a roadside video; (ii) the problem due to the occlusions of neighbor vehicles is better solved by combining fast extracting and grouping of feature points with generating and analyzing of FTSI, the accuracies of traffic parameters measurement and vehicles classification are improved; (iii) for vehicles classification, the influence that resulted from slow driving or temporary stopping of vehicles is eliminated accurately; (iv) the constructing method of the vehicular feature vector improves the accuracy of vehicles classification further. Experiments have proved the efficiency and practicability of the proposed method.

The remaining sections are organized as follows. Section 2 describes the problem and presents our method framework. Sections 3–5 elaborate the method in detail. Section 6 presents experiments and analyzes the results. Section 7 concludes the paper with some discussions.

2. Problem Description and Method Framework

Figure 1(a) shows the traffic camera installed at roadside and Fig. 1(b) shows its frame image and the two VDLs set by the proposed method. Rapid and accurate measurement of traffic parameters based on the camera is challenging due to the occlusions of neighbor vehicles in adjacent lanes and the stopping of vehicles on the VDL. In the proposed method, these challenging problems are solved effectively by combining VDL-based FTSI with extracting and grouping of vehicular feature points.

The framework of the proposed method is shown in Fig. 2. First, the calibration of traffic scene is performed during the installation of the surveillance system, where the positions of the road lanes are calibrated manually and the camera parameters are calibrated via the method in Ref. 14. At the same time, two VDLs are set in upstream and downstream of the road and they are perpendicular to the lane-lines according to

(a) (b)

Fig. 1. (a) Installation of camera. (b) A sample of video frame and VDLs setting.

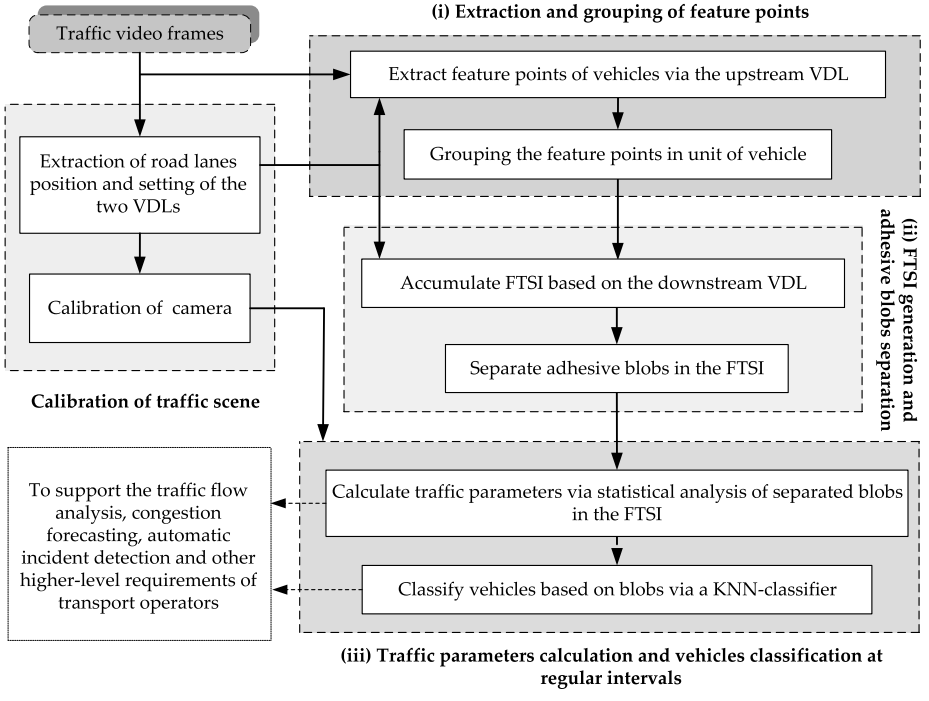


Fig. 2. Flowchart of the proposed method.

the projection relationship of camera as shown in Fig. 1(b), where the distance between the two VDLs is as far as possible to better support the elimination of vehicles occlusion. The widths along the lane direction of the two VDLs are set to seven pixels based on empirical data of experiments, which provides effective support for the feature-points extracting and grouping, the FTSI generation and the adhesive blobs separation.

During the work, based on experiments and performance analysis, first we adopt median filtering method to filter the noise in the original video images.[17,23] Then feature points of vehicles are extracted via the upstream VDL and are grouped in unit of vehicle based on their movement differences. FTSI is accumulated from video frames based on the downstream VDL and the adhesive blobs in the FTSI are separated effectively based on the groups of feature points and a projection histogram of blob pixels (see Sec. 3 for details). At regular intervals, the traffic parameters are calculated and vehicle types are identified based on the corrected blobs in the FTSI. For vehicles classification, the influence of blob distort along time axis, which results from the stopping of the vehicle on the VDL, are eliminated accurately based on the vehicular instantaneous speed on the VDL, in which the speed is provided by feature point trackers.

In the following three sections, we will discuss the proposed method in detail from three aspects: (i) extracting and grouping of vehicular feature points; (ii) FTSI

generation and adhesive blobs separation; (iii) traffic parameters calculation and vehicles classification.

3. Extracting and Grouping of Vehicular Feature Points

This step is aimed to (i) provide the number and the location of vehicles to accurately separate the adhesive blobs in the downstream-VDL-based FTSI and (ii) obtain vehicle speed more accurately that not only is a output parameter but also is used to eliminate the blobs distort of vehicles stopped on the downstream VDL and improves the accuracy of vehicles classification.

For the extraction of vehicular feature points, we do not adopt the traditional extraction methods because they have two main disadvantages: (i) they extract feature points in whole image, complicated calculations result in slow speed; (ii) the extracting accuracy and the grouping accuracy of feature points are influenced by unwanted background pixels. To solve these problems, the proposed method is based on the upstream VDL to extract vehicular feature points quickly and accurately. For the upstream VDL, we firstly build its background image via the GMM modeling method to obtain vehicular pixels, and then extract feature points of vehicles by using the automatic extracting method of good feature point.[27] Our experiments show that the seven-pixel height of the VDL ensures the effect of background modeling and feature-points extracting.[26]

Though the relationship of the extracted feature points to vehicles has been recorded preliminarily during their extraction, some points may be grouped incorrectly since neighboring vehicles occlusion in the upstream VDL, especially in heavy traffic. So our method tracks the extracted feature points immediately via the pyramidal Kanade–Lucas–Tomasi (KLT) trackers[28] and then corrects the incorrect groups based on the movement differences of feature points. The key points of the part are described as follows.

First, as mentioned above, the tracking of feature points is based on pyramidal KLT tracker. However, if we use the tracker alone, the feature points often drift when the local luminance of vehicles changes suddenly in the video or the vehicles run faster than the frame rate. Lots of experiments prove that the uniform local binary pattern (ULBP)[29] texture descriptors have better robustness on the above negative factors, so the proposed method adopts the ULBP texture descriptors to eliminate the drifting points and enhance the matching accuracy of feature points between frames during the tracking process. For a feature point A in a frame image, if the tracked position A' in a subsequence frame image shifts obviously, their ULBP texture descriptors will be calculated in their neighboring windows in the frames, respectively. Taking A for example, its ULBP texture descriptor is as follows

$$ULBP_{P,R}(A) = \begin{cases} \sum_{i=0}^{P-1} (S(g_A, g_i) \times 2^i), & U(LBP_{P,R}(A)) \leq 2 \\ \xi, & \text{otherwise} \end{cases}, \tag{1}$$

where g_A and g_i are the pixel values of the current point A and its ith neighboring pixel point, P is the number of the neighboring points, R is the neighboring radius set 3 in this paper, ξ is a constant selected as $P+1$ in this paper based on experiments, $S(g_A, g_i)$ and $U(LBP_{P,R})$ are the binarization function and the jump number function that are constructed as follows

$$S(g_A, g_i) = \begin{cases} 1, & (g_A - g_i) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$U(LBP_{P,R}(A)) = |S(g_A, g_{p-1}) - S(g_A, g_0)| + \sum_{i=1}^{P-1} |S(g_A, g_i) - S(g_A, g_{i-1})|. \tag{3}$$

If the difference between the ULBP texture descriptors of A and A' meets the following criteria, the matching judged to be successful. Otherwise, the feature point will be deleted as an invalid tracking point.

$$|ULBP_{P,R}(A) - ULBP_{P,R}(A')| \leq \delta, \tag{4}$$

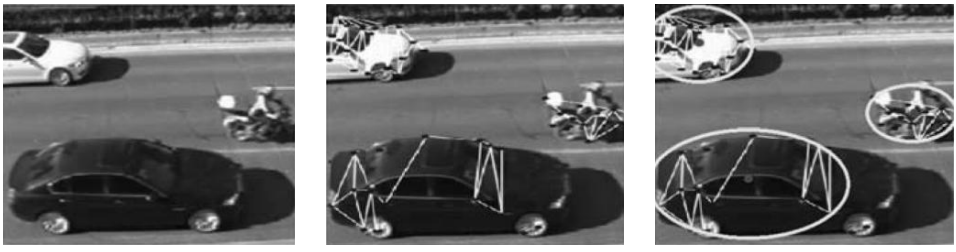
where δ is a threshold, whose value is obtained by the Otsu adaptive threshold method.[30]

Second, in the correction of incorrect groups of feature points, a vehicular point-group is described as an edge-weighted dynamic graph (EWDG) expressed as $G = \{V, E, W\}$. The "V" is the set of the vehicular feature points, "E" expresses the set of optimal edges that link feature points in V and are generated via the Delaunay-triangulation method,[31] and "W" is the set of weights that correspond to the edges and are set as the movement differences of the two feature-points connected by the corresponding edge. Since moving vehicle has rigid body features, feature points on a vehicle have high movement similarities. However, feature points on different vehicles have high movement differences. For two feature points, if they belong to a same vehicle, their spatial distance may change due to the change of vehicle position and the perspective transformation, but the change is smooth and slight. However, if the two points belong to different vehicles, their spatial distance often changes more obviously because vehicles often change their speed or make lateral moves to adapt the changes of traffic stream and so the relative positions of the adjacent vehicles often change in actual traffic scenes, especially in heavy traffic that the original incorrectly groupings are more likely to occur. Based on this characteristic, we proposed a correction algorithm via a state transition model to modify the point groups effectively during the tracking of feature points. For any two feature points i and j in the t_0th frame image, their state vector $E_{i,j}^{(t_0)}$ is expressed as follows:

$$E_{i,j}^{(t_0)} = [d_x^{(t_0)}, d_y^{(t_0)}]^T, \tag{5}$$

where

$$\begin{cases} d_x^{(t_0)} = \max\limits_{t\in[0,t_0]} |x_i^{(t)} - x_j^{(t)}| - \min\limits_{t\in[0,t_0]} |x_i^{(t)} - x_j^{(t)}| \\ d_y^{(t_0)} = \max\limits_{t\in[0,t_0]} |y_i^{(t)} - y_j^{(t)}| - \min\limits_{t\in[0,t_0]} |y_i^{(t)} - y_j^{(t)}| \end{cases}, \tag{6}$$



(a) (b) (c)

Fig. 3. Extracting and grouping of vehicular feature points. (a) An original image. (b) Extracting and grouping results. (c) Vehicles description based on their centers of feature points.

where $(x_i^{(t)}, y_i^{(t)})$ and $(x_j^{(t)}, y_j^{(t)})$ are the spatial coordinates of feature points i and j in the tth frame respectively. Then the predictive value $\hat{E}_{i,j}^{(t_0)}$ and the modification value $\tilde{E}_{i,j}^{(t_0)}$ of the state vector can be calculated based on the information of the (t_0-1)th frame via the following recursion formulae

$$\begin{aligned} \hat{E}_{i,j}^{(t_0)} &= \mathbf{A}\tilde{E}_{i,j}^{(t_0-1)} \\ \tilde{E}_{i,j}^{(t_0)} &= \hat{E}_{i,j}^{(t_0)} + K(Z_{i,j}^{(t_0)} - \mathbf{H}\hat{E}_{i,j}^{(t_0)}) \end{aligned}, \tag{7}$$

where $\mathbf{A}$ is transition matrix, $\mathbf{H}$ is observation matrix and K is Kalman gain,[32] $Z_{i,j}^{(t_0)}$ is measurement vector in t_0 frame and $Z_{i,j}^{(t_0)} - \mathbf{H}\hat{E}_{i,j}^{(t_0)}$ is movement differences vector. When any component of $Z_{i,j}^{(t_0)} - \mathbf{H}\hat{E}_{i,j}^{(t_0)}$ exceed an established threshold three times in succession, the two points will be judged to belong to different vehicles and the corresponding edge will be deleted. Then the incorrect groups of feature points can be corrected effectively. An example of extracting and grouping results of feature points is shown in Fig. 3(b). The tracking for any feature point will maintain until the point cross and leave the downstream VDL completely. Then the vehicles can be descripted based on the centers of feature points as shown in Fig. 3(c), which will be used to separate adhesive blobs and eliminate blobs distortion in the downstream FTSI.

4. FTSI Generation and Adhesive Blobs Separation

FTSI is generated based on the center line of the downstream VDL. Then adhesive blobs of occlusion vehicles in the FTSI are separated effectively based on the groups of feature points and a projection histogram of pixels in the blobs.

4.1. *FTSI generation*

For eliminating the influence of background pixels and vehicle shadows, the foreground image in VDL is extracted first via a modified background subtraction

method, and then the undesired moving shadows are eliminated based on texture similarity, finally the FTSI is generated based on the undisturbed foreground image.

First, Gaussian mixture model method is employed to obtain the VDL background images. Considering that vehicles that stops temporarily or runs slowly on the VDL under congestion might be misidentified as background, we employ a modified update rate[32–34] to update the background model.

Based on the obtained background image, the foreground in VDL can be extracted by subtraction method, but there are undesired moving shadows with the vehicle pixels in the foreground image. Analyses show[25,35–37] that if a pixel belongs to shadow region, its local texture feature in foreground image is always similar with that in background image, but if the pixel belongs to a vehicle, the above similarity does not exist, so we adopt an algorithm as shown in Fig. 4 to eliminate the shadows and generate the FTSI.

Define *Mask* **as the size of** F_t **(viz.** B_t**)**
  for per *pixel* (x, y) **in** *Mask* **do**
    if $F_t(x, y) \neq 0$ **then**
      $Mask\ (x, y) \leftarrow 255$
    else
      $Mask\ (x, y) \leftarrow 0$
    end if
  end for
  for per *pixel* (x, y) **in** *one ROI* **do**
    if $F_t(x, y) \neq 0 \wedge F_t(x, y) - B_t(x, y) < 0$ **then**

$$Texture_t^F = \sum_{j=1}^{24}((g_j^{F_t(x,y)} - F_t(x,y)) / \sum_{k=1}^{24} k)$$

$$Texture_t^B = \sum_{j=1}^{24}((g_j^{B_t(x,y)} - B_t(x,y)) / \sum_{k=1}^{24} k)$$

      if $|Texture_t^F - Texture_t^B| \leq \sigma$ **then**
        $Mask(x, y) \leftarrow 100$
      end if
    end if
  end for
  for per *pixel* (x, y) **in** *Mask* **do**
    $Mask\ (x, y) \leftarrow 0$
  end if
  select one *pixel* (x, y) **as** *seed* **in** *Mask* **under condition**
    $Mask\ (x, y) = 100 \wedge$ *num of its zero neighbors* ≥ 3
  end select
  use *region growing* **with** *seed* **to detect** *more-accurate shadow*
  eliminate *shadow* **for** *foreground image* F_t
  Filter F_t **using** *size filter* **to eliminate** *leftover little- noise*
  accumulate F_t *pixels that are on the center line of the downstream-VDL* **to generate** *FTSI*

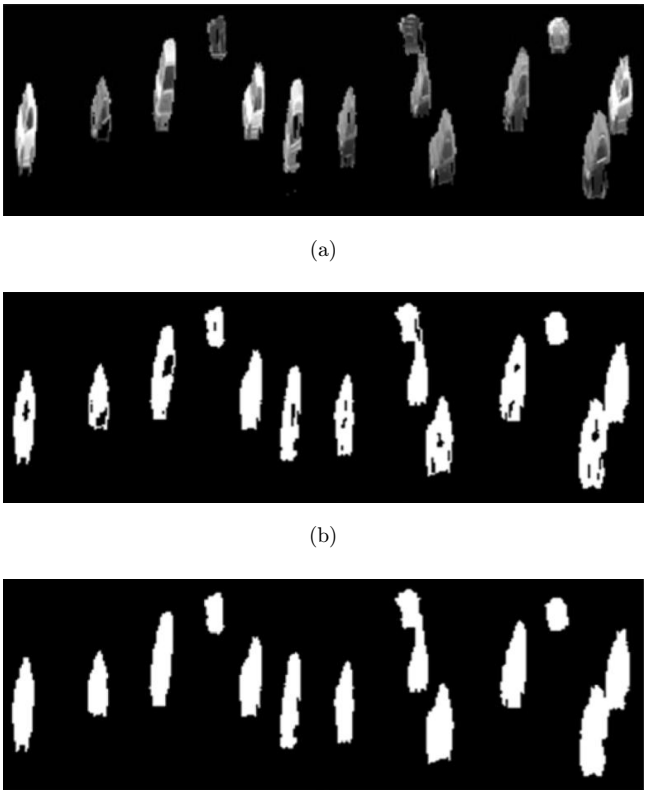
Fig. 4. Algorithm of shadow elimination and FTSI generation.

In Fig. 4, lines 1 to 24 are used to detect moving shadows, where the *Mask* is designed for saving the shadow location. F_t is the foreground image at moment t, $F_t(x, y)$ is the current pixel in F_t, $g_j^{F_t(x,y)}$ is the jth neighbor in the texture window of the current pixel in F_t. B_t is the background image at moment t, $B_t(x, y)$ is the current pixel in B_t, $g_j^{B_t(x,y)}$ is the jth neighbor in the texture window of the current pixel in B_t. After detecting shadows, line 25 is used to eliminate shadows from the foreground image. Line 26 is used to eliminate the leftover little-noise by using the size filter, and the last line is used to obtain the FTSI.

The FTSI is generated by concatenating together the treated F_t pixels, which are on the center line of the downstream VDL, over time. The column pixels vector at the moment t of the FTSI is as follows

$$\mathrm{FTSI}_t = [P_{t,0}, P_{t,1}, \ldots, P_{t,N}]^T, \tag{8}$$

where $P_{t,i}(0 \leq i \leq N)$ is the ith foreground pixel on the center line of the downstream VDL, N is the length of the downstream VDL. A sample of the FTSI is shown in Fig. 5(a).



(a)

(b)

(c)

Fig. 5. (a) An original FTSI. (b) The binaryzation FTSI. (c) The processed FTSI.

4.2. *Separation of adhesive blobs and correction of distorted blobs*

In the obtained FTSI, there is some interference such as holes, blobs adhesion and blobs distort. These interference factors influence the traffic parameters calculation and the vehicles classification. Here we firstly perform binarization processing for FTSI as shown in Fig. 5(b) and adopt a method based on contour detection and holes fill[26] to eliminate the holes in blobs as shown in Fig. 5(c).

Then we design a method based on the groups of feature points and a projection histogram of pixels in the blobs to effectively separate adhesive blobs in the FTSI due to vehicles occlusion. For each feature-points group obtained from Sec. 3, we describe it with its center as shown in Fig. 3(c). Taking the ith group as an example, its final EWDG is $G_i = \{V_i, E_i, W_i\}$. We take out all the feature points from V_i and calculate the center C_i of the ith group as follows:

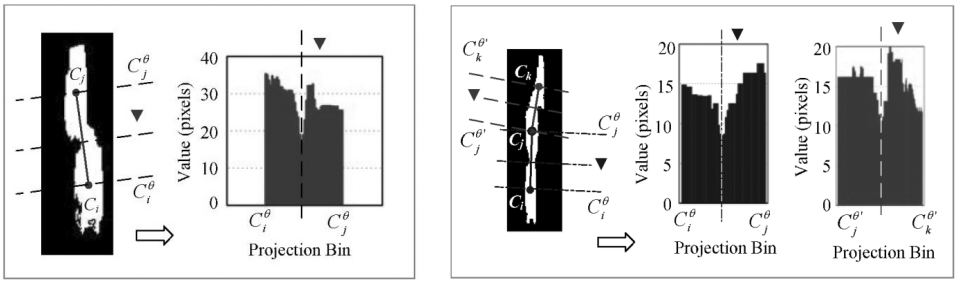
$$C_i = (x_{C_i}, y_{C_i}) = \left(\sum_{j=1}^{M} x_j / M, \sum_{j=1}^{M} y_j / M \right), \tag{9}$$

where (x_j, y_j) is the coordinate of the jth feature point in V_i, M is the feature points number in V_i.

For each blob in the FTSI, if two groups of feature points (e.g. group i and group j) fall in it as shown in Fig. 6(a), we calculate the projection histogram of pixels in the blob between the edges C_i^θ and C_i^θ of the two group centers (C_i and C_j) in the direction that has the included-angle θ with the positive direction of x-axis of the image, and then separate the blob from the location with the min value, where the θ is calculated as follows:

$$\theta = \theta_{\overrightarrow{c_i c_j}} + \frac{\pi}{2}, \tag{10}$$

where $\theta_{\overrightarrow{c_i c_j}}$ is included-angle of the vector $\overrightarrow{c_i c_j}$ with the positive direction of x-axis of the image.



(a) (b)

Fig. 6. Adhesive blobs separation. (a) Separation of two adhesive blobs. (b) Separation of multi adhesive blobs.

If more than two groups fall in a blob, we use the above method with formula (10) to separate them one by one. Taking three groups as an example, we express their centers as C_i, C_j and C_k respectively, and then the process of separation is shown in Fig. 6(b).

5. Traffic Parameters Calculation and Vehicles Classification

At regular intervals, traffic parameters are calculated in unit of lane via statistical analysis of blobs and vehicle types are classified based on geometrical features of the blob in the FTSI. For a blob, the lane it belongs to is judged by its bottom pixels, which can reduce mistakes resulting from the perspective projection errors. The calculation of traffic parameters and the classification of vehicles are introduced below.

5.1. *Traffic parameters calculation*

For each time interval, we can easily calculate many traffic parameters based on the FTSI obtained above. In this paper, we focus on three main traffic parameters: (i) flow rate q which is defined as the number of vehicles that pass the downstream VDL per unit time; (ii) mean speed v which is defined as the average value of speeds of all vehicle that pass the downstream VDL during the time interval; (iii) occupancy o which is defined as the ratio of the sum of all the times that vehicles pass the downstream VDL and the length of the time interval. Taking the kth lane in the tth time interval as an example, the three parameters defined above are calculated as follows:

$$\begin{aligned} q_t^k &= N_t^k/\Delta t \\ v_t^k &= \sum_{i=1}^{N_t^k} v_i/N_t^k \\ o_t^k &= \sum_{i=1}^{N_t^k} w_i/\Delta t \times 100\% \end{aligned} \tag{11}$$

where N_t^k is the number of vehicles that pass the upstream VDL in the kth lane during the tth time interval, Δt is the length of time interval, v_i is the speed of the ith vehicle calculated as the mean speed of all the feature points in the vehicle group when they cross the VDL, w_i is the blob width of the ith vehicle along the time axis.

5.2. *Vehicles classification*

From the vehicular blobs in the FTIS, we can obtain legible side geometrical characteristic of the vehicles. At the same time, the fact that different vehicle types

have different side outline provides strong support to recognize different types of vehicles.

First, we eliminate the influence of blob distort due to stopping of the vehicle on the VDL. For each blob, we obtain the vehicular instantaneous speeds during it passes the VDL based on the group and the trackers of feature points. For any moment that the vehicular instantaneous speed is zero, we eliminate the corresponding column pixels of the blob in the FTSI.

The corrected blobs provide strong support to shape-based classification of vehicles. However, if we use the whole shape information of a blob as the classification feature, the feature vectors are not aligned and the computation is heavy. Here we construct a geometrical feature vector to solve the above problems and represent the blob effectively. For the ith vehicle blob, the feature vector constructed as follows:

$$F_i = [h_i, l_i, Vs_i, Hs_i]^T, \tag{12}$$

where h_i and l_i are the actual height and length of the ith vehicle. They are calculated based on the vehicular height and the vehicular width in pixel respectively. The vehicular height in pixel is the blob height in pixel. The vehicular width in pixel can be calculated as follows

$$w_i = w_i' \cdot v_i / f_s, \tag{13}$$

where f_s is the frame rate of the video, w_i' is the blob width in pixel, v_i is the instantaneous speed of the vehicle when it pass the VDL and its calculation is as same as the v_i in formula (11). In addition, Vs_i (or Hs_i) in formula (12) is a vertical (or horizontal) shape vector constructed as follows

$$\begin{aligned} Vs_i &= [Vs_i^1, Vs_i^2, Vs_i^3, Vs_i^4, Vs_i^5]^T \\ Hs_i &= [Hs_i^1, Hs_i^2, Hs_i^3, Hs_i^4, Hs_i^5, Hs_i^6, Hs_i^7, Hs_i^8]^T, \end{aligned} \tag{14}$$

where $Vs_i^j (1 \le j \le 5)$ is a spline of vertical uniform sampling and $Hs_i^k (1 \le k \le 8)$ is a spline of horizontal uniform sampling. They are generated as shown in Fig. 7. From this figure and the formula (14), we can see that the sampling quality is correlated with the vehicle speed and the frame rate of the camera. Analysis shows that frame rate of 30 fps can guarantee the sampling quality well for urban transportation surveillance.[26] In addition, the high-frequency cameras are widely applied in the transportation surveillance system today, which provides strong support to ensure sampling quality of over-speed vehicles.

Then, a KNN classifier is trained to classify vehicles based on the above feature vector, where the vehicle types include sedan, hatchback, microbus, bus, SUV, van and pick-up truck.

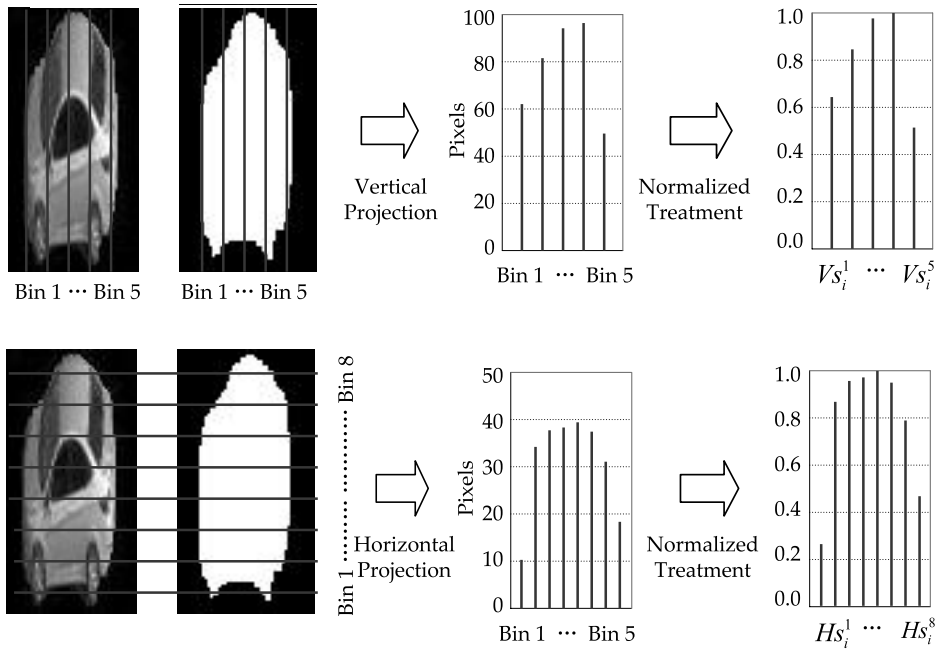


Fig. 7. Generation of the vertical and the horizontal shape vectors for vehicles classification.

6. Experiment Result Analysis

6.1. *Testing samples and experiment settings*

In order to test the performance of the proposed method, experiments are performed based on a real material set of 540 lane-minutes traffic videos, which includes six videos with 30 min length that are collected by a traffic surveillance camera in Beijing. These videos are collected under three different weather conditions, including sunny, cloudy and snowy, and under two traffic conditions, including peak hours and off-peak hours, respectively. In the traffic scene, the number of lanes is 3 and all the video images have 640*480 pixels@30 fps. In experiments, all the codes of the methods are implemented in VC++ 2010. The experimental results are analyzed in the next subsection.

6.2. *Experimental result analysis*

6.2.1. *Result analysis of traffic parameters calculation*

We first select a 15 min video material as a typical example to analyze the results of q, v and o in detail. There are notable traffic flow fluctuations in the scene, so its results and analysis has typical significance. In the experiments, the lengths of all time intervals are set as 1 min. Figure 8 shows the results of the three parameters, where the ground truth data are extracted manually with the aid of a toolkit-software

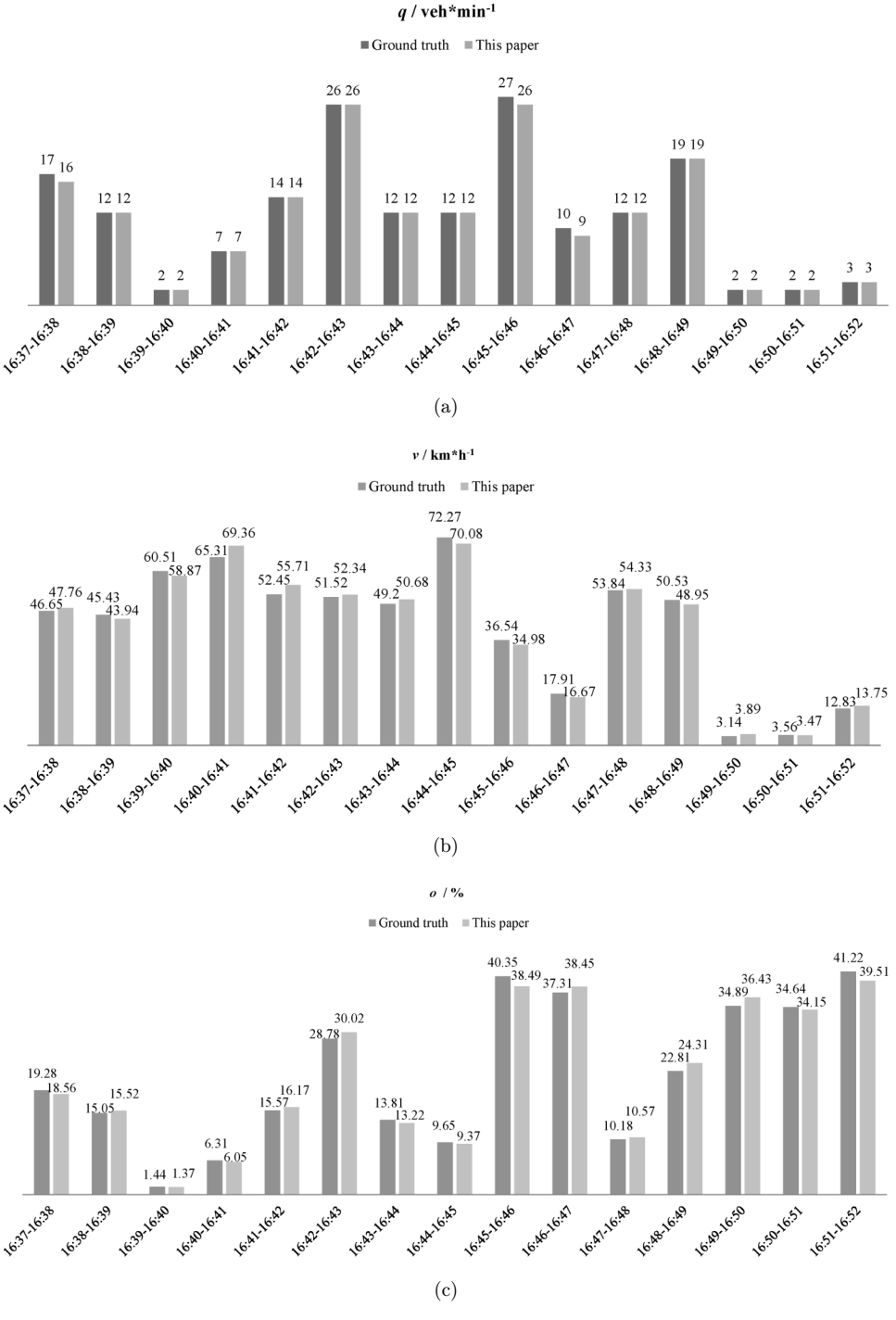


Fig. 8. Measurement results of traffic parameters. (a) Results of q. (b) Results of v. (c) Results of o.

"ViPER"[38] that is widely used for evaluating performance of vision-based detection methods. From the subfigures, we can see that our method can detect almost of all the parameters accurately for each time interval, including those congestion time-periods. Analysis shows that the mean errors of the measurement results for q, v and o can provide solid foundation for higher-level requirements of transport operators.

In order to evaluate the proposed method more fully, we performed a statistical analysis on all experimental results for all the video materials. Table 1 shows the statistical results of the measurement errors of traffic parameters, where L1–L3 are the lane numbers and the lane 1 (L1) is the nearest lane from the camera. In the table, the classical virtual-line-based method is the traditional technology based on two detection lines without considering the influence of shadows. The improved virtual-line-based method is based on two TSIs.[26] We can see that the average measurement errors for all the parameters are influenced by weather condition, traffic conditions and lane locations to some extent.

For the influence of weather condition, we can see that all the methods have the best accuracy in cloudy day and have the worst accuracy in snowy day. Analysis shows that this is mainly because there are moving shadows in sunny day and moving reflections in snowy day, which influences the measurement accuracies of the methods. In this case, the improved virtual-line-based method adopts shadow elimination measures so its results better than that of the classical methods, and the proposed method adopts not only shadow elimination measures but also more efficient separation measures so it received the best results. For the influence of traffic conditions, we can see that the result accuracies of any method in off-peak hours are better than that in peak hours, which is mainly due to the influence of vehicle occlusion. However, the impact of this factor on the proposed method is slighter than the other two methods obviously, which is benefit from the efficient separation of the blobs in FTSI.

For the influence of the lane locations, the situations and analysis are similar to that of traffic conditions. The influence of vehicle occlusion to method accuracy is most serious for lane 3 (the farthest lane from the camera) but is most slight for lane 1 (the nearest lane) result from the perspective transformation of the camera. Because the proposed method adopts the efficient measure of blobs separation, its error results for lane 3 are obviously smaller than those of the other methods, especially in peak hours.

6.2.2. *Results analysis of vehicle classification*

The results of vehicle classification are shown in Table 2, where we contrast the accuracy data of the proposed method with those of the method in Ref. 26. From the table, we can see that the average accuracy of the proposed method is higher than that of the method in Ref. 26 under the cloudy condition. For the sunny condition, the proposed method classified 94.73% of the vehicles accurately which is about 4% points higher than that of the method in Ref. 26. For the snowy condition, although

Table 1. Statistical analysis of measurement errors of the traffic parameters (%).

			Sunny			Cloudy			Snowy			
Method	Parameter	In Peak Hours	L1	L2	L3	L1	L2	L3	L1	L2	L3	Mean
Classical Virtual-line-based Method	q	yes	5.64	21.71	24.90	3.64	13.71	19.92	7.37	25.22	26.31	16.49
		no	3.72	10.17	12.39	2.35	6.83	9.04	4.22	12.64	13.51	8.32
	v	yes	14.55	20.51	24.28	11.52	15.01	18.16	13.68	17.24	22.45	17.48
		no	7.11	10.49	13.26	5.37	8.79	12.97	8.25	12.31	15.72	10.47
	o	yes	13.69	20.98	25.62	9.71	15.96	18.24	15.42	21.96	27.71	18.81
		no	9.03	12.54	16.16	6.03	9.43	13.12	12.87	15.05	17.31	12.39
Improved Virtual-line-based Method	q	yes	4.42	11.67	14.86	3.75	10.27	14.11	8.51	16.66	19.28	11.50
		no	2.06	4.31	5.26	1.34	2.87	3.69	4.53	8.15	7.89	4.46
	v	yes	6.12	13.98	14.71	3.18	9.59	11.96	7.74	15.72	19.67	11.41
		no	4.78	5.84	6.87	2.65	4.34	5.26	5.13	8.47	9.22	5.84
	o	yes	4.18	12.96	15.15	3.64	11.95	13.37	5.55	13.75	16.32	10.76
		no	2.99	4.22	4.92	2.26	4.66	5.18	5.16	5.10	6.02	4.50
Method in This Paper	q	yes	1.98	4.15	5.29	1.86	3.94	4.79	3.43	5.83	7.31	4.29
		no	1.66	2.57	2.83	1.09	1.99	2.56	2.18	4.69	4.32	2.65
	v	yes	3.02	4.31	4.81	2.22	3.74	4.23	3.97	5.61	5.88	4.20
		no	2.27	2.87	4.27	1.24	2.57	3.26	3.36	3.62	5.79	3.25
	o	yes	3.64	4.45	5.73	2.28	3.91	4.50	4.47	5.39	6.04	4.49
		no	2.87	3.68	4.67	1.99	3.07	3.73	4.18	4.50	5.88	3.84

Table 2. Statistical analysis of the accuracy of vehicles classification.

Method	Weather Conditions	Accuracy of Vehicles Classification (%)							
		Hatchback	Sedan	SUV	Microbus	Bus	Van	Pick-Up Truck	Mean
Method in Ref. 26	Sunny	90.69	91.97	89.29	90.62	94.86	90.19	88.02	90.81
	Cloudy	90.97	92.32	90.03	92.59	95.14	91.54	89.14	91.68
	Snowy	82.69	84.55	87.84	89.01	91.07	86.17	85.68	86.72
	Mean	88.17	89.61	89.05	90.74	93.69	89.30	87.61	89.74
This Paper	Sunny	93.74	95.56	95.83	92.96	98.44	94.87	91.73	94.73
	Cloudy	95.34	96.08	97.72	93.70	98.60	95.68	92.72	95.69
	Snowy	91.28	92.89	89.54	89.88	93.49	87.28	89.26	90.52
	Mean	93.45	94.84	94.36	92.18	96.84	92.61	91.24	93.65

the correct recognition rate of the proposed method is less than those under the other two weather conditions, it is higher than that of the method in Ref. 26.

In addition, the correct recognition rates of bus are generally higher than other vehicle types. Analysis shows that this is mainly because buses more appear in the nearest lane where the occlusions is slight and their geometrical characteristic is obviously different from other vehicle types. In a word, the testing results of the proposed method are all greater than 90% that significantly outperform those of the contrastive literature. Analysis shows that the improvement of the classification accuracies mainly benefits from the efficient separation measure of blobs and the construction of the vehicular feature vector.

7. Conclusion

This paper focuses on the online automatic measurement of traffic parameters and classification of vehicles types based on video. A novel high-efficient measurement and classification method is proposed to provide powerful support to traffic flow analysis, congestion forecasting, automatic incident detection and other higher-level requirements of traffic surveillance. In the proposed method, the number of vehicles in a blob in the FTSI and the accurate instantaneous speed of vehicles at the location of a downstream-VDL are provided by the trackers of vehicular feature points, which fully guarantees the performance of traffic parameters measurement and vehicles classification. The extraction of vehicular feature points is based on an upstream VDL and a local background modeling, which effectively reduces the computational complexity of whole image processing and the influence of background pixels to quality of feature points. The proposed method can detect many traffic parameters and recognize vehicle types based on FTSI directly and high-efficiently.

The proposed method can work automatically, but its accuracy will reduce when one lane is congested but others are moving freely. As future works, we will attempt to utilize the adaptive segmental technology of VDLs to improve the method performance.

Acknowledgments

The authors would like to thank the Natural Science Foundation for Doctors of Langfang Teachers University (Grant No. LSLB201409), the Scientific Research Foundation of Hebei Higher Education (Grant No. QN2015209), the Soft Science Foundation of Hebei Province (study on planning strategy of intelligent road traffic of Jing-Jin-Xiong Triangle under the coordinate development background) and the scientific research & originality team of Computational Intelligence and Intelligence System in Langfang Teachers University (Grant No. B-04). In addition, we appreciate the guidance from Dr. Halid Mahama.

References

1. M. J. Cobo *et al.*, A bibliometric analysis of the intelligent transportation systems research based on science mapping, *IEEE Trans. Intell. Transport. Syst.* **15** (2014) 901–908.
2. X. Y. Hu *et al.*, Vehicle detection technology based on cascading classifiers of multi-feature integration, *Int. J. Pattern Recogn. Artif. Intell.* **31** (2017) 1750032.1–1750032.15.
3. M. Kafai and B. Bhanu, Dynamic bayesian networks for vehicle classification in video, *IEEE Trans. Indus. Inf.* **8** (2012) 100–109.
4. Y. S. Frank and X. Zhong, Automated counting and tracking of vehicles, *Int. J. Pattern Recogn. Artif. Intell.* **31** (2017) in press.
5. B. T. Morris and M. M. Trivedi, Learning, modeling, and classification of vehicle track patterns from live video, *IEEE Trans. Intell. Transport. Syst.* **9** (2008) 425–437.
6. L. J. Qin *et al.*, Improved position and attitude determination method for monocular vision in vehicle collision warning system, *Int. J. Pattern Recogn. Artif. Intell.* **30** (2016) 1655019.1–1655019.15.
7. Y. Shan, H. S. Sawhney and R. Kumar, Unsupervised learning of discriminative edge measures for vehicle matching between nonoverlapping cameras, *IEEE Trans. Pattern Anal. Mach. Intell.* **30** (2008) 700–711.
8. K. F. Wang and Y. J. Yao, Video-based vehicle detection approach with data-driven adaptive neuro-fuzzy networks, *Int. J. Pattern Recogn. Artif. Intell.* **29** (2015) 1555015.1–1555015.32.
9. C. L. Azevedo *et al.*, Automatic vehicle trajectory extraction by aerial remote sensing, *Procedia-Soc. Behav. Sci.* **111** (2014) 849–858.
10. R. Y. Du and Z. R. Peng, Time effectiveness analysis of UAV vehicle detection data in reverse direction, *J. Transport. Syst. Eng. Inf. Technol.* **14** (2014) 34–40.
11. S. Kamijo *et al.*, Traffic monitoring and accident detection at intersections, *IEEE Trans. Intell. Transport. Syst.* **2** (2000) 108–118.
12. Y. Y. Fu *et al.*, Traffic incident detection technology based on highly-shot video, *J. Highway Transport. Res. Develop.* **31** (2014) 128–134.
13. L. Unzueta *et al.*, Adaptive multi-cue background subtraction for robust vehicle counting and classification, *IEEE Trans. Intell. Transport. Syst.* **13** (2012) 527–540.
14. O. Bulan *et al.*, Video-based real-time on-street parking occupancy detection system, *J. Electron. Imag.* **22** (2013) 6931–6946.
15. M. Dubskaá *et al.*, Fully automatic roadside camera calibration for traffic surveillance, *IEEE Trans. Intell. Transport. Syst.* **16** (2015) 1162–1171.
16. Y. Zheng and S. Peng, A practical roadside camera calibration method based on least squares optimization, *IEEE Trans. Intell. Transport. Syst.* **15** (2014) 831–843.

17. J. Q. Ren *et al.*, Detection and analysis of traffic flow characteristic parameters based on time-space trajectory tracking, *J. Transport. Syst. Eng. Inf. Technol.* **15** (2015) 62–68.
18. P. Pawlik, Z. Bublinński and A. Glowacz, Color analysis supporting a traffic flow measurement based on optical flow, *Image Process. Commun.* **19** (2015) 45–49.
19. Q. F. Zhang, Q. Chen and D. Y. Wang, Image brightness correction for frame-difference-based intelligent video monitoring systems, *J. Appl. Opt.* **34** (2013) 778–783.
20. L. Unzueta *et al.*, Adaptive multi-cue background subtraction for robust vehicle counting and classification, *IEEE Trans. Intell. Transport. Syst.* **13** (2012) 527–540.
21. C. Setchell, Applications of computer vision to road-traffic monitoring, Ph.D. dissertation, University of Bristol, 1997.
22. Y. H. Hue, A traffic-flow parameters evaluation approach based on urban road video, *Int. J. Eng. Intell. Syst.* **2** (2009) 33–39.
23. C. Francisco and F. Alejandro, Low complexity algorithm for the extraction of vehicular traffic variables, in *Proc. Int. IEEE Conf. Intelligent Transportation Systems*, Washington, DC, USA, October (2011), pp. 1021–1026.
24. N. U. Rashid, N. C. Mithun and B. R. Joy, Detection and classification of vehicles from a video using time-spatial image, in *Proc. IEEE Int. Conf. Electrical and Computer Engineering*, Dhaka, Bangladesh, January (2011), pp. 502–505.
25. N. C. Mithun, N. U. Rashid and S. M. Rahman, Detection and classification of vehicles from video using multiple time-spatial images, *IEEE Trans. Intell. Transport. Syst.* **13** (2012) 1215–1225.
26. J. Q. Ren, L. Xin and Y. Z. Chen, High-efficient detection of traffic parameters by using two foreground temporal-spatial images, in *Proc. Int. IEEE Conf. Intelligent Transportation Systems*, The Hague, Netherlands, October (2014), pp. 1965–1970.
27. J. Shi and C. Tomasi, Good features to track, in *Proc. 1994 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, June (1994), pp. 593–600.
28. J. Suhr, Kanade-lucas-tomasi (KLT) feature tracker computer vision lab, Computer Vision, 2009.
29. T. Ojala, M. Pietikainen and T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* **24** (2000) 971–987.
30. J. Guo *et al.*, Preceding vehicle detection and tracking adaptive to illumination variation in night traffic scenes based on relevance analysis, *Sensors* **14** (2014) 15325–15347.
31. L. J. Guibas, D. E. Knuth and M. Shrir, Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica* **7** (1992) 381–413.
32. J. Q. Ren *et al.*, Lane detection in video-based intelligent transportation monitoring via fast extracting and clustering of vehicle motion trajectories, *Math. Problems Eng.* **14** (2014) 1–12.
33. H. H. Lin, J. H. Chuang and T. L. Liu, Regularized background adaptation: A novel learning rate control scheme for Gaussian mixture modeling, *IEEE Trans. Image Process., A Pub. IEEE Signal Process. Soc.* **20** (2011) 822–836.
34. Y. L. Tian, A. Senior and M. Lu, Robust and efficient foreground analysis in complex surveillance videos, *Mach. Vis. Appl.* **23** (2012) 967–983.
35. J. Q. Ren, Novel adaptive algorithm of shadow elimination for video moving objects, *Comput. Eng. Appl.* **46** (2010) 188–191.
36. Y. S. Lee and W. Y. Chung, Video based abnormal event detection with moving shadow removal in home healthcare application, *Sensors* **12** (2012) 573–584.
37. B. C. Yin, Y. Liu and Z. F. Wang, Moving shadow detection by combining chromaticity and texture invariance, *J. Image Graph.* **19** (2014) 896–905.

38. B. Axel, B. Marco and E. Julia, A review and comparison of measures for automatic video surveillance systems, *EURASIP J. Image Video Process* **5** (2008) 1–30.

Jianqiang Ren received his Ph.D. in Control Science and Engineering and M. S. degree in Pattern Recognition and Intelligent System from Beijing University of Technology, Beijing, China, in 2016 and 2010. He is currently an Associate Professor at the Institute of Pattern Recognition and Intelligent System at the Langfang Teachers University. His research interests cover pattern recognition, computer vision and intelligent system.

Chunhong Zhang received her M. S. degree in Computer Science and Technology from the Beijing University of Technology, Beijing, China, in 2010. She is currently a lecturer at the College of Mathematics and Information Science at Langfang Teachers University. Her research interests include artificial intelligence and computer application.

Lingjuan Zhang received her M. S. degree in Mechanical Design and Theory from the Yanshan University, Qinhuangdao, China, in 2002. She is currently an Associate Professor at the College of Physics and Electrical Information at the Langfang Teachers University. Her research interests include pattern recognition, intelligent system, etc.

Ning Wang received her M. S. degree in Integrated Circuit Engineering from Beihang University, Beijing, China, in 2017. She is currently an assistant at the College of Mathematics and Information Science at Langfang Teachers University. Her research interests include computer vision, image processing and artificial intelligence.

Yue Feng was born in 1992 and received her M. S. degree in Control Science and Engineering from Tianjin University. Currently, she is an assistant at the Institute of Pattern Recognition and Intelligent System at the Langfang Teachers University. Her research direction includes intelligent detection and intelligent system.