IET Intelligent Transport Systems

Research Article

IET Journals
The Institution of Engineering and Technology

Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition

ISSN 1751-956X
Received on 16th February 2017
Revised 24th August 2017
Accepted on 22nd October 2017
E-First on 6th December 2017
doi: 10.1049/iet-its.2017.0047
www.ietdl.org

Honghong Yang[1] ✉, *Shiru Qu*[1]

[1]*Department of Automation, Northwestern Polytechnical University, Xi'an 710072, People's Republic of China*

✉ *E-mail: yanghonghong0615@163.com*

Abstract: Real-time vehicle counting can efficiently improve traffic control and management. Aiming to efficiently collect the real-time traffic information, the authors propose an effective vehicle counting system for detecting and tracking vehicles in complex traffic scenes. The proposed algorithm detects moving vehicles based on background subtraction method with 'low-rank + sparse' decomposition. For accurately counting vehicles, an online Kalman filter algorithm is used to track the multiple moving objects and avoid counting one vehicle repeatedly. The proposed method is evaluated on three publicly available datasets, which include seven video sequences with various challenging scenes for detection performance evaluation, and another two video sequences for vehicle counting evaluation. The experimental results demonstrate a good performance of the proposed method in terms of both qualitative and quantitative evaluations.

1 Introduction

Real-time estimation the number of vehicles in traffic video sequences is an important task in intelligent transportation system (ITS), which can offer reliable information for traffic management and control. The number of vehicles on-road reflects the traffic status, such as road-traffic intensity, lane occupancy and congestion level. This kind of information can be used for early incident detection, road congestion prevention and automated route planning [1].

In a traditional ITS, vehicle counting is always solved by employing special sensors such as magnetic loop, microwave or ultrasound detectors. However, those sensors have some limitations due to their simple formats or high installation cost [2]. With the development of digital video processing, a vision-based vehicle counting system, coupled with video camera and image processing techniques, offers an attractive appeal due to its powerful ability in detecting vehicle type, density, velocity and even traffic accident. In addition, the vision-based vehicle counting system has the advantages over traditional sensor methods in terms of flexibility, low implementation cost, easy installation and maintenance [3].

Video-based vehicle detection methods can be categorised into three different classes: frame difference method [4], optical flow method [5] and background subtraction (BS) method [6]. The frame difference method is fast and simple, but only detects parts of the moving objects. This is because this method compares the difference between moving objects and the background in successive frames only. Since the optical flow method needs to calculate the optical information of the whole image, it is often difficulty to satisfy the real-time applications. BS method is one popular technique for traffic detection. It first implements the background modelling and then finds the moving objects by comparing the difference between the input image and the background. One major drawback of BS is that modelling the background in real scenarios is not easy. A common practice of a number of BS methods is to initialise the background model by an 'empty scene'. However, even initialised with the 'empty scene', it is still difficult to maintain the background model throughout the entire sequence in real applications.

To efficiently overcome the main weaknesses of BS, in this paper, we propose a new video-based vehicle detection and counting method for complex traffic scenes. The proposed method detects vehicles by BS with sparse and low-rank decomposition, which is robust to illumination or weather changes. Then, an online Kalman filter algorithm is used to track each vehicle in several frames in order to obtain a reliable vehicle counting result. The main contributions of this paper are as follows:

(i) We online detect moving objects by the 'low-rank + sparse' decomposition of an input matrix $\boldsymbol{Z} = \boldsymbol{L} + \boldsymbol{E}$, where sparse outlier $\boldsymbol{E}$ consists of two parts, moving objects $\boldsymbol{S}$ and noise $\boldsymbol{G}$, $\boldsymbol{E} = \boldsymbol{S} + \boldsymbol{G}$. A typical noise $\boldsymbol{G}$ may be arisen from various illumination conditions or waving trees.
(ii) Group sparsity constraint is used to further improve the accuracy of foreground detections in complex dynamic scene. The group sparsity constraint, taking the spatial-temporal structure of neighbourhood pixels into consideration, makes the algorithm more robust to random noise.
(iii) Traffic objects detection is implemented in an online manner without requiring the entire image sequences. So our method can online update the background model and is robust to dynamic sceneries.

This paper is organised as follows: Related works are summarised in Section 2. Section 3 describes the details of the proposed method for moving vehicle detection and background estimation in complex traffic scenes, followed by our vehicle counting method in Section 4. Experimental results are discussed in Section 5, and conclusion is drawn in Section 6.

2 Related works

Vehicle detection is the first step to collect traffic parameters. Over the past few decades, a large number of vehicle detection and counting approaches have been developed. Existing vehicle detection approaches can be divided into appearance-based and motion-based classes [7]. The appearance-based methods locate vehicles using visual features, such as colour, texture, edge or shape. In [8], Held *et al.* presented a probabilistic framework for vehicle detection by vehicle context and scale. Tsai *et al.* proposed a statistic linear model for vehicle detection by colour and edge map [9]. In another work, Jia *et al.* detected vehicles by a generated model, which is based on edge information and Markov chain Monte-Carlo method [10]. Recently, in [6], Luis *et al.* presented a multi-cue vehicle segmentation architecture to fuse different image

© The Institution of Engineering and Technology 2017

cues. Li *et al.* utilised a multi-scale model to detect vehicles by fusing multiple features [11]. Fusing different visual features has been proven an effective approach to obtain moving objects. Visual features, as shown in previous works, have the ability to distinguish vehicles from the background. However, these appearance-based methods are always time consuming. Hence, those methods have some limitations to handle real-time data.

The motion-based methods always assume that everything is static except for vehicles in a traffic scene. This is impractical in real traffic scene, with many examples such as lighting fluctuations, pedestrians or waving branches. Even illumination changes may cause moving shadows in traffic scene. A number of algorithms have been proposed to overcome the above issues. Kim *et al.* [12] presented a codebook model by using statistical filters to eliminate continuous slight movements on the background. Other researchers have proposed improved BS methods [13–18]. The most popular algorithms for background extraction include the mixture of Gaussians model (MOG) [14], kernel density estimation [15], block correlation [16], codebook model [12], Hidden Markov model [17] and linear autoregressive models [18]. These methods, however, have two weaknesses. In initialisation stage, the BS method assumes that the object does not appear at the very first frame, or an 'empty scene' the first frame is, which is impractical in real applications. Another important disadvantage is that it is difficult to maintain and update the background model throughout one image sequence in realistic traffic circumstance.

Vehicle detection provides useful information for video-based vehicle counting. As known, the counting result can offer reliable information for traffic control [19]. The task of vehicle counting is to estimate the number of vehicles presented in a given image [20]. The existing vision-based vehicle counting approaches can be divided into three categories: counting by detection, counting by clustering and counting by regression [21]. In general, the clustering [22] or regression [23] methods need to explicitly extract the object feature in order to build an accurate appearance model. This requires a large number of training data from traffic scene to train the regression model or build a special vehicle appearance model in an offline manner. The detection-based methods [24] need to explicitly segment the objects from the background. Thus, they need to build a robust background model and then find the moving objects by detecting difference significant enough between one input video sequence and their background. Therefore, building a proper background model is the key step for an accurate vehicle-counting method. Recently, Zhang *et al.* [25] proposed a vehicle counting method based on foreground time-spatial image. The main steps in their work are background initialisation, foreground detection and background updating. They use a self-adaptive sample consensus BS method to model background, aiming to resolve the deficiencies in traditional counting method, such as computationally expensive and failure-prone in realistic traffic scenarios. In [26], Barcellos *et al.* presented a novel video-based detection and vehicle counting algorithm which combines BS, particle filter tracking and clustering algorithm. They exploit the motion coherence of objects to achieve foreground targets. In addition, Gaussian mixture models are used to perform background modelling. Vehicle tracking is implemented by using particle filter with spatial adjacency of the targets. Then, vehicles are counted by detecting the intersections of the tracked targets with user-defined virtual loops. Quesada *et al.* [27] proposed an automatic vehicle counting method by using principal component pursuit (PCP) to implement background modelling, which is similar with our work. The authors in [27] and ours use the low-rank approximation to achieve the background model and foreground detections. The PCP method, considered to be the state-of-the-art video background modelling method, is a kind of low-rank approximation algorithm. Their works consist of two stages: training and counting. They pay more attention on training stage, including motion segmentation, feature extraction and parameter estimation. After training, vehicle counting is performed by implementing motion segmentation and feature extraction with the parameters estimated in the training stage.

In our work, the background modelling is performed by low-rank approximation, which is similar with [27]. However, in our 'low-rank + sparse' decomposition framework, input data are regarded as a combination of background $\boldsymbol{L}$, moving objects $\boldsymbol{S}$ and noise G. Meanwhile, group sparsity constraint is used in our work to obtain improved foreground detection results. In addition, vehicle counting is performed by multi-object tracking method. According to the tracking results, we can build reliable target motions and achieve a reliable counting result. There is no pre-trained, no feature extraction in our works, which is different from the study in [27].

3 Proposed method

In this section, we present our scheme for real-time vehicle detection and counting in detail. Our methodology is based on the well-known BS method. It consists of three stages: background modelling, vehicles detection and vehicle counting.

3.1 Online background modelling

Video background modelling is the first step for vehicle detection. The sparse representation and low-rank decomposition methods based on robust principal component analysis (RPCA) or PCP are considered to be the state-of-the-art methods in background modelling [28]. Methods under this framework follow the idea that background sequence is modelled by a low-rank matrix and the moving objects correspond to the sparse outliers. There are several methods proposed to decompose the input matrix $\boldsymbol{Z}$ into a low-rank matrix $\boldsymbol{L}$ and a sparse outlier $\boldsymbol{S}$, including RPCA [29], PCP [30], online robust PCA [31], go decomposition (GoDec) [32], detecting contiguous outliers in the low-rank representation (Decolor) [33], contiguous outliers representation via online low-rank approximation (Corola) [34], incremental principal component pursuit (incPCP) [28] and so on.

Although several methods have been proposed to solve the low-rank approximation problem and provided good performance in background modelling [35], a major drawback of existing algorithms is the high computational cost. Those methods always operate in a batch manner and assume that the observed video is noise free. However, the real traffic scene is more challenging due to high traffic density, occlusions and complex dynamic environments. Therefore, it is unable to directly apply existing algorithms in real-time traffic detection. A reliable object detection and robust background modelling method, as a result, is still an open issue in BS [36].

In this paper, the proposed BS method is based on two constraints. The first one is the background areas in video sequence are linearly correlated with each other, i.e. areas will arise a low-rank matrix. The second one is that we have the prior knowledge that, one moving object is observed by a sparse and contiguous piece in consecutive frames, and usually occupies a relatively small portion of an image. In addition, locations of a moving object in successive frames are likely to group together, instead of scattered randomly. This indicates that the moving objects satisfy the group sparsity constraint [37]. Furthermore, the proposed BS method is applied in real traffic scene. Vehicle pixels are always corrupted by noises, such as illumination changes, shadow moving, waving leaves and branches. By considering the low-rank and group sparsity constraints, we formulate the BS problem by decomposing the input observed image $\boldsymbol{Z}$ into a low-rank matrix $\boldsymbol{L}$ and a binary sparse matrix $\boldsymbol{S}$, along with a noise term G

$$\boldsymbol{Z} = \boldsymbol{L} + \boldsymbol{S} + G \tag{1}$$

Then, we formulate our objective function as follows:

$$\min_{X,E} \frac{1}{2}\| \boldsymbol{Z} - \boldsymbol{L} - E \|_{\mathrm{F}}^2 + \lambda_1 \| \boldsymbol{L} \|_* + \lambda_2 \Omega(E) \tag{2}$$

where $\boldsymbol{Z} = [z_1, z_2, \ldots, z_n] \in \mathbb{R}^{m \times n}$ is a matrix representing n images, $z \in \mathbb{R}^m$ is a vectorised image. $E = \boldsymbol{S} + G$, $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm, $\| \cdot \|_*$ is the nuclear norm and $\Omega(E) = \sum_{j=1}^{n} \sum_{g \in v} \| E_g^i \|_\infty$ is the group sparsity constraint. $E \in \mathbb{R}^{m \times n}$, $E^i \in \mathbb{R}^n$ is the ith

© The Institution of Engineering and Technology 2017

column in E with n indices $\{1, 2, \ldots, n\}$. $g \in v$ contains a subset of these indices. Here, we set 3×3 group neighbours.

Since (2) is a non-convex problem, we follow the method in [31] to reformulate $\boldsymbol{L}$ and relax the nuclear norm to the Frobenius norm. Then (2) can be presented by the following optimisation problem for each observed image:

$$\min_{U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{r \times n}, E} \frac{1}{2} \| \boldsymbol{Z} - \boldsymbol{U}\boldsymbol{V} - E \|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}(\| \boldsymbol{U} \|_{\mathrm{F}}^2 + \| \boldsymbol{V} \|_{\mathrm{F}}^2) + \lambda_2 \Omega(E) \tag{3}$$

where $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$, $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ is the basis matrix of the low-dimensional subspace and the corresponding coefficient is $\boldsymbol{V} \in \mathbb{R}^{r \times n}$. r is the upper bound for the rank of $\boldsymbol{U}$ and $\boldsymbol{V}$. λ_1 controls the basis and coefficients for low-rank matrix and λ_2 controls the group sparsity constraint for the foreground.

In particular, the optimisation problem in (3) can be solved by two-step alternating minimisation. The first step is low-rank approximation by fixing E to minimisation $\boldsymbol{L}$. The second step is the sparse optimisation by fixing $\boldsymbol{L}$ to minimisation E.

3.2 Online background estimation

Initialisation: In order to meet the time complexity, the number of low-dimensional subspace basis is initialised by the first N frames. Since the OR-PCA method is used in sparse optimisation, the initialisation step for basis $\boldsymbol{U}$ is in a batch manner. In this case, the rank r is roughly estimated for the rest of images. Since only the leading N video frames are used to finish the initialisation for $\boldsymbol{U}$ and considering the small value of $N = 3$, the complexity of implementing the batch manner is limited. After the initialisation, we have achieved $\boldsymbol{L}_{t-1}$ and E_{t-1}, where $\boldsymbol{Z}_{t-1} = \boldsymbol{L}_{t-1} + E_{t-1}$, $\boldsymbol{Z}_{t-1} = \boldsymbol{Z}(:, 1:t-1)$ and $\boldsymbol{L}_{t-1} = \boldsymbol{U}_r \boldsymbol{V}_r^{\mathrm{T}}$, $\boldsymbol{U} \in \mathbb{R}^{m \times r}$, $\boldsymbol{V} \in \mathbb{R}^{r \times m}$. For every new frame z_t, the optimisation problem in (3) can be rewritten as

$$\min_{\boldsymbol{L}, E} \frac{1}{2} \| \boldsymbol{Z}_t - \boldsymbol{L}_t - E_t \|_F^2 + \lambda_1 \| \boldsymbol{L}_t \|_{\mathrm{F}}^2 + \lambda_2 \Omega(E_t), \quad s.t.\ rank(\boldsymbol{L}_t) \le r \tag{4}$$

Then an incremental approach is used to solve (4) by implementing the two-step alternating minimisation as follows:

$$\boldsymbol{L}_t^{k+1} = \arg\min_{\boldsymbol{L}} (\| \boldsymbol{Z}_t - \boldsymbol{L}_t - E_t^k \|_{\mathrm{F}}^2), \quad s.t.\ rank(\boldsymbol{L}_t) \le r \tag{5}$$

$$E_t^{k+1} = \arg\min_{E} \| \boldsymbol{Z}_t - \boldsymbol{L}_t^{k+1} - E_t \|_{\mathrm{F}}^2 + \lambda_2 \Omega(E_t) \tag{6}$$

where $\boldsymbol{Z}_t = \boldsymbol{Z}_{t-1} + z_t$, $\boldsymbol{L}_t = \boldsymbol{L}_{t-1} + l_t$ and $E_t = E_{t-1} + e_t$. $\boldsymbol{Z}_t, \boldsymbol{L}_t$ and E_t represent the input image set, background set and noise set up to the tth frame, respectively. z_t, l_t and e_t are the input image, background and noise term in the tth frame, respectively. The superscript k is the iteration number and the subscript t can be any frame in an image sequence.

3.3 Online foreground detection

In the second step of foreground detection, the sparse outlier E is a combination of the true foreground mask $\boldsymbol{S}$ and noise data G. The noise data have limited influence on background model because they are assumed belonging to moving parts only. However, they can cause false alarm during foreground detection.

For current frame $z_t \in \mathbb{R}^m$, we compute the foreground mask $s_t \in \mathbb{R}^m$ from the sparse outlier e_t, where $e_t = z_t - l_t$. To handle noisy data, in this step, we employ the sparse and contiguous pieces [37]. Then, we take the group sparsity constraint into consideration by implementing MRF algorithm to obtain the foreground mask [38]. The energy function of the foreground mask s_t is defined as follows:

$$\sum_{j=1}^{n} \sum_{i,j \in v} \| s_t(i,j) \|_{\infty} + \sum_{j}^{v} \sum_{t}^{\tau} w^2(i,t) N_x^2(i,j,t) \tag{7}$$

where $v \in m \times n$ is a set of vertices that corresponding to the image pixels, τ is the number of neighbourhood, $N_x \in \mathbb{R}^{n \times \tau}$ is the value of $S's$ neighbour, $w \in \mathbb{R}^{n \times \tau}$ is the weight for neighbourhood. For a 2D image, we set $\tau = 4$. Then the weight is $w = \mathrm{e}^{-((z_t^i - z_t^j)/2\sigma)}$. z_t^i and z_t^j are, respectively, the grey value for the ith and jth pixel in the tth frame image. σ is the variance for an image. Here, the gco-v3.0 library [39] is used to optimise (7).

Then the second step of the optimisation problem that detects moving objects from the sparse outlier in (7) can be rewritten as minimise the energy function as follows:

$$\min_{\boldsymbol{L}, \boldsymbol{S}} \frac{1}{2} \sum_{i,j} (\boldsymbol{Z}_t - \boldsymbol{L}_t)^2 + \sum_{j=1}^{n} \sum_{i,j \in v} \| s_t(i,j) \|_{\infty} + \sum_{j}^{v} \sum_{t}^{\tau} w^2(i,t) N_x^2(i,j,t), \quad s.t.\ rank(\boldsymbol{L}_t) \le r \tag{8}$$

Equation (8) can be reformulated as the first-order MRF as in [38]. The graph-cut algorithm in [40] is used to solve s_t. The result of s_t in (8) is the foreground component in $\boldsymbol{Z}_t$.

Here, we need to mention that the group sparsity constraint used in our work, aiming to improve the foreground detection, is similar in [37]. Although our work and [37] both rely on low-rank approximation with the group sparsity constraint, the major differences are: (i) The low-rank approximation is online formulated in our work, which enables the proposed method to process the input video one frame at a time. The authors in [37] employ a batch formulation, which needs to pre-train from the image sequences. (ii) They [37] formulate the input data into the background and the sparse outlier. In our work, the low-rank approximation for the foreground object detection takes the noise term into consideration, and formulates the sparse outlier E into two parts, moving objects $\boldsymbol{S}$ and noise G, where $E = \boldsymbol{S} + G$. (iii) To effectively separate foreground moving objects $\boldsymbol{S}$ from noise G, we use the group sparsity constraint for the moving objects, and solve the constrained problem by the first-order MRF and graph-cut. In [37], they assume the foreground trajectories of the objects should be short and the targets should move slowly. Their background modelling and foreground detection method are performed by decomposing the motion trajectory matrix after obtaining the foreground and background trajectories. As claimed in [37], their work heavily depends on trajectory tracking technique, and if the tracking technique fails, their method may not work well.

4 Real-time vehicle counting

In order to improve the accuracy of vehicle counting, multi-object tracking algorithm is applied to refine the result of the foreground detection. The foreground moving objects are achieved by 'low-rank + sparse' method in Section 3. In this section, we first use Hungarian algorithm to assign detections for trackers and form short contiguous tracklets [41]. Then, we estimate the object states in new frames by implementing online Kalman filter [42]. The estimation is determined by the previous object states as follows:

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{A}\boldsymbol{x}_{t-1} + \boldsymbol{B}u_t + Q_t \\ y_t &= \boldsymbol{H}\boldsymbol{x}_t + R_t \end{aligned} \tag{9}$$

where $\boldsymbol{x}_t = [p_x(t), p_y(t), \dot{p}_x(t), \dot{p}_y(t), w, h]^{\mathrm{T}}$ is the state vector. $[p_x(t), p_y(t)]^{\mathrm{T}}$ is the vehicle position in the tth frame. $[\dot{p}_x(t), \dot{p}_y(t)]^{\mathrm{T}}$ is the vehicle velocity, $[w, h]$ is the vehicle size. $\boldsymbol{y} = [p_x, p_y, w, h]^{\mathrm{T}}$ is the observation vector. u_t is the control input. $\boldsymbol{A}$ is the state matrix. $\boldsymbol{B}$ is the input matrix. $\boldsymbol{H}$ is the observation matrix, Q_t and R_t are Gaussian noises.

We first use Kalman filter to predict object states for existing trackers in current frame. Then, we associate the foreground

© The Institution of Engineering and Technology 2017

detection results with the trackers by performing the Hungarian algorithm. The cost function for association is defined by the Euclidean distance between the predicted position of object states and the position of foreground detections. Finally, according the association results, each tracker is updated by the associated detection. The object state, predicted by Kalman filter, is also corrected by the new foreground detection result. Simultaneously, the number of visible tracker increases Section 1. The unassociated trackers are marked as invisible and its counter increases accordingly. If a tracker is invisible for many consecutive frames, Section 4 in this paper, we assume that the object moves out and delete its tracker. We also create a new tracker for any unassociated foreground detection. The noisy detections tend to result in short-lived trackers. For this reason, in our work, the foreground detection is regarded as a single vehicle object only if it is tracked in consecutive frames and the visible counter exceeds a predefined threshold. This means that we count a vehicle until its total visible number exceeds a specified threshold (Section 5 in experiments).

5 Experiments

5.1 Evaluation the vehicle detection

In order to evaluate the foreground detection results of our algorithm, experiments are done on several traffic video sequences. A summary of the sequences used in experiments is shown in Table 1. The video sequences are available in'ChangeDtetection.net' (CDnet2014) [43] and Youtube. The proposed method is tested on these traffic video datasets and compared with the state-of-the-art algorithms, including MOG [14], GoDec [32], Decolor [33], Corola [34] and incPCP [28]. To perform a fair comparison, all parameters in compared methods are provided by the authors with default parameters. For quantitative evaluation, the metrics of precision, recall and the F-measure are used to show the overall accuracy, defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$

$$F\_\text{measure} = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where TP, FP and FN are the number of true positives, false positives and false negatives, respectively. Figs. 1 and 2 show the results of the proposed method on these traffic video datasets, Figs. 3–9 show the qualitative evaluation among different compared methods by visual assessment of some detected binary objects mask and the background model for all test sequences. Table 2 and Fig. 10 show the quantitative evaluation results by precision, recall and F-measure on four test videos where the ground truth is available.

Figs. 1*a* and 3 are the highway sequence in daytime. The main challenge of this sequence is that the background (highway and trees) is constantly occluded by shadows (of moving cars and waving trees) in a sunny day. In Fig. 1, column 1 shows the original input images. Columns 2 and 3 show the estimated background $\boldsymbol{L}$ and the sparse outlier E. Columns 4 and 5 are the estimated noise G and the detected foreground objects $\boldsymbol{S}$, respectively. Column 6 shows the tracking results in different frames. Fig. 3 shows the comparative results of the six background estimation models. From Figs. 3*b*, *f, i* and *k*, we can observe that the foregrounds detected by MOG, Decolor, Corola and incPCP are slightly contaminated by noise. The results of GoDec and the proposed method are seldom affected by noise because they take the noise term into sparse approximation, which is beneficial to keep the foreground detection from being corrupted by noise. In terms of the GoDec results in Fig. 3*d*, the shadows caused by moving vehicles and waving trees show a powerful effect on foreground detection. The moving vehicles have been regarded as noise because the GoDec algorithm overestimates noise. Only the proposed method, showing in Figs. 3*m* and *n*, can effectively build background model and keep the foreground objects from being corrupted.

Figs. 1*b* and 4 are the intermittentPan sequence in daytime. The main challenge for this sequence is that the background is constantly occluded by waving trees in a sunny day. Fig. 1*b* shows the results of the proposed method in this sequence. Fig. 4 shows the comparative results of the six background estimation models. We can observe that the foregrounds detected by MOG in Fig. 4*b*, Decolor in Fig. 4*f*, Corola in Fig. 4*i* and incPCP in Fig. 4*k* are contaminated by noise in different degree. The results of GoDec and the proposed method are merely affected by noise and both give a good result.

Figs. 1*c* and 5 are the Streetcorneratnight sequence at nighttime. The main problem for this sequence is that the vehicle headlights have a strong effect on foreground detection. In addition, the illumination in nighttime is very low, which makes the foreground detection more difficult. Fig. 1*c* shows the results of the proposed method at frame 864, 882, 970 and 986. Fig. 5 shows the comparative results of the six background estimation models. We can see from Figs. 5*b*, *d* and *f* that the vehicle headlight has a strong effect on foreground detections. MOG, GoDec and Decolar all regard the reflection as foreground objects. Corola in Fig. 5*i* and incPCP in Fig. 5*k* show better performance to those illumination changes. However, due to the absence of noise term in their works, the results of their foreground detection are easily affected by noises. The proposed method gives the best results in this test sequence.

Figs. 1*d* and 6 show the tramStation sequence in nighttime. The main challenge for this sequence is the light reflection. Fig. 1*d* shows the results of the proposed method at frame 511, 619, 710 and 780. Fig. 6 shows the comparative results of the competing methods. As shown in Fig. 6, the foreground detection results by MOG, Decolor, Corola and incPCP are easily affected by noises. Due to light changes, GoDec regards several vehicles as noises. The proposed method, as seen in Figs. 1*d* and 6*m* and *n*, shows its superiority over the competing methods.

Some experiments have been made using the streetlight sequence from CDnet2014 dataset, highwayII and Rheinhafen sequences from Youtube. The streetlight sequence in Fig. 2*a* and 7 refers to a sunny day with waving trees. The foreground detection results by MOG, Decolar and Corola are heavily affected by noises. GoDec and incPCP have a poor performance on foreground detection as shown in Fig. 7*d* and *j*. The proposed method can accurately get the foreground and the background model as shown in Figs. 7*l* and *m*. The highwayII sequence in Figs. 2*b* and 8 consists of a crowed scene with many vehicles on highway. We can see that the proposed method gives a good foreground detection and background estimation. As known, Decolar is a batch method, resulting in a global optimal solution on foreground detection. The results for MOG, incPCP and Corola are heavily affected by noises. Due to the overestimated noise, GoDec only detects parts of the vehicles. Rheinhanfen in Figs. 2*c* and 9 is a daytime sequence with intermittent object moving. We can see from Fig. 2*c* that, the vehicle in frame 61 at the intersection was stopped, and other vehicles were slowing down. The stopped vehicle in frame 133 started moving again and left the scene, while the others still waited at intersection. As shown in Fig. 2*c*, the proposed method updates the background model accurately for each frame. The batch manner-based foreground detection methods, like Decolor in Fig. 9*g*, have not completely 'forgotten' the moving vehicles presented in previous frames, and the foreground detection result thus contains previous moving objects. The MOG, Corola and incPCP methods only detect parts of the moving vehicles. The foreground detection by GoDec misses vehicles because it pays more attention on noise term.

Table 2 and Fig. 10 show the quantitative evaluation results by precision, recall and F-measure on the four test video sequences where the ground truth is available. The higher values of these three metrics are, the better performance one algorithm has. Five state-of-the-art BS methods, MOG and four other BS-based low-rank and sparse decomposition methods (GoDec, Decolor, Corola and incPCP), are used for performance comparison with the proposed method. As illustrated in Table 2, the precision, recall and F-measure values of the proposed method are higher than that of the other five methods. According to our experimental results, the

© The Institution of Engineering and Technology 2017

proposed method is superior to other compared methods in terms of both qualitative and quantitative valuations.

5.2 Evaluation the vehicle counting result

To demonstrate the efficiency of the proposed method on vehicle counting, we compare the vehicle counting results of the proposed method with the state-of-the-art works like [27, 44]. Since some codes for the state-of-the-art works are not publicly available or the test videos used in their papers are difference, in our work, we only choose [27, 44] as the competing methods. Works in [27] also use low-rank approximation to build video background model, which is similar in our work. In addition, the authors in [27] are an extended work of incPCP method [28]. The incPCP method, as a baseline method in Section 5.1, is used to compare with the proposed method in terms of detection results. Since the reported counting results in [44] have the highest success ratio in many test sequences, we also compare our method with [44].

In experiment, the test videos in Section 5.1 come from CDnet2014 and Youtube (for detection evaluation), which aim to compare the background model and the foreground detection results of the proposed method with related works, as shown in Figs. 3–9. In experiment Section 5.2, we focus on comparing the vehicle counting results. So we select two sequences from the

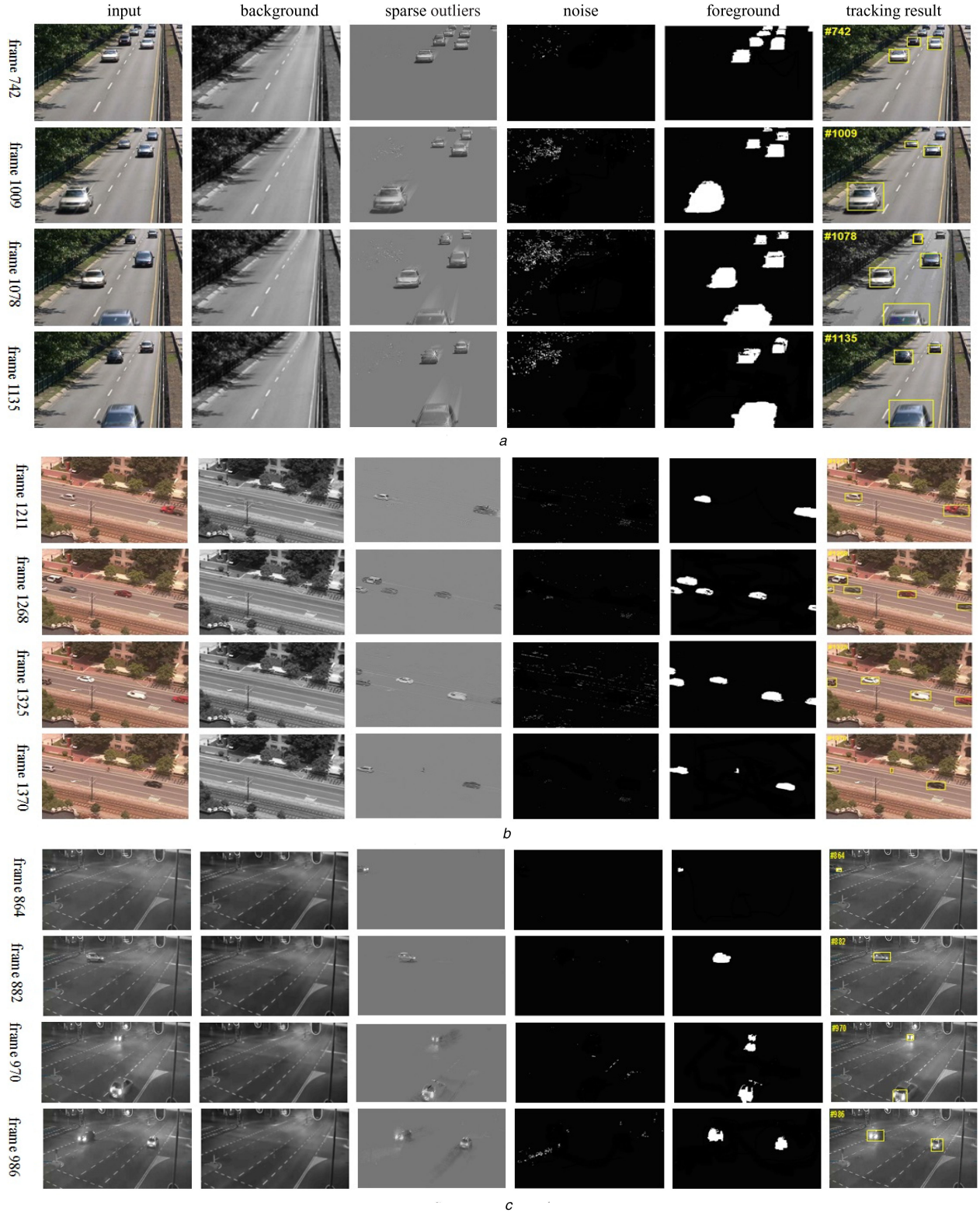

Fig. 1 *Continued*

© The Institution of Engineering and Technology 2017

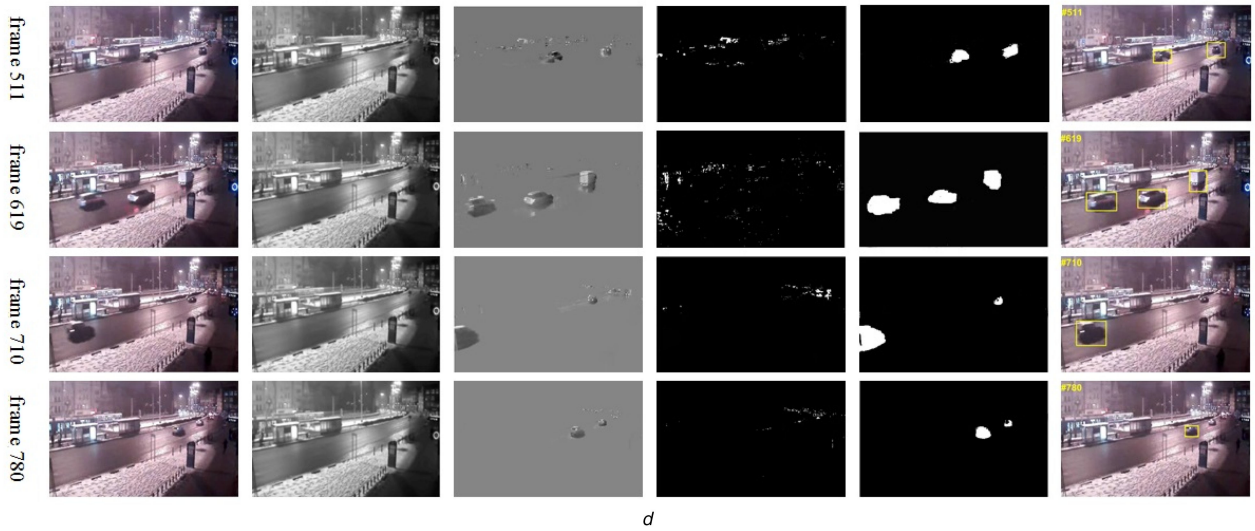

Fig. 1 *Results of the proposed method on four ground truth available sequence. From left to right: Column 1 shows the original input images. Columns 2 and 3 show the results of the estimated background* ***L*** *and sparse outlier E. Columns 4 and 5 show the results of the estimated noise G and detected foreground objects* ***S****, respectively. Column 6 is the tracking results*

(a) Highway sequence, ***(b)*** IntermittentPan sequence, ***(c)*** Streetcorneratnight sequence, ***(d)*** TramStation sequence

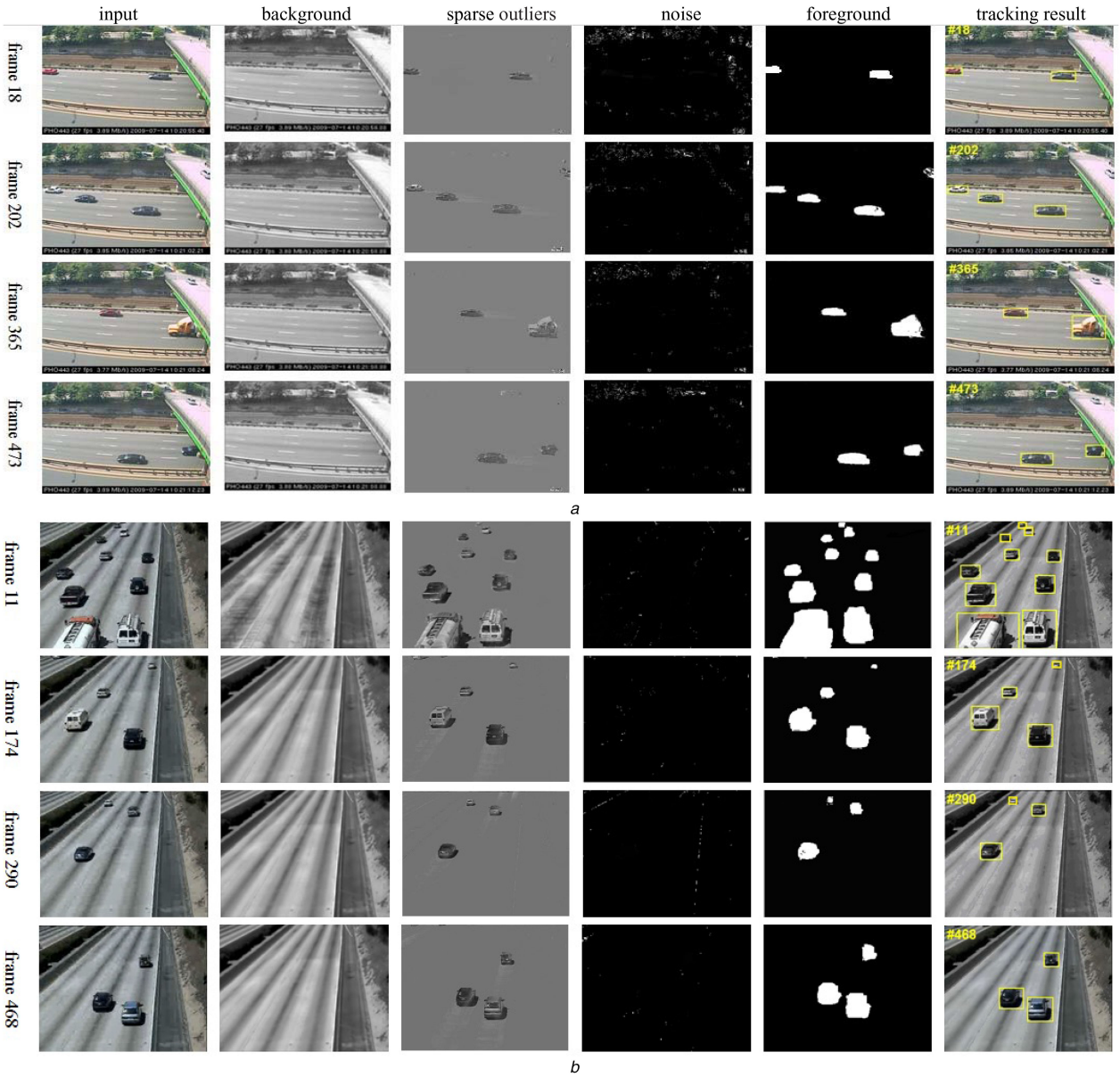

Fig. 2 *Continued*

© The Institution of Engineering and Technology 2017

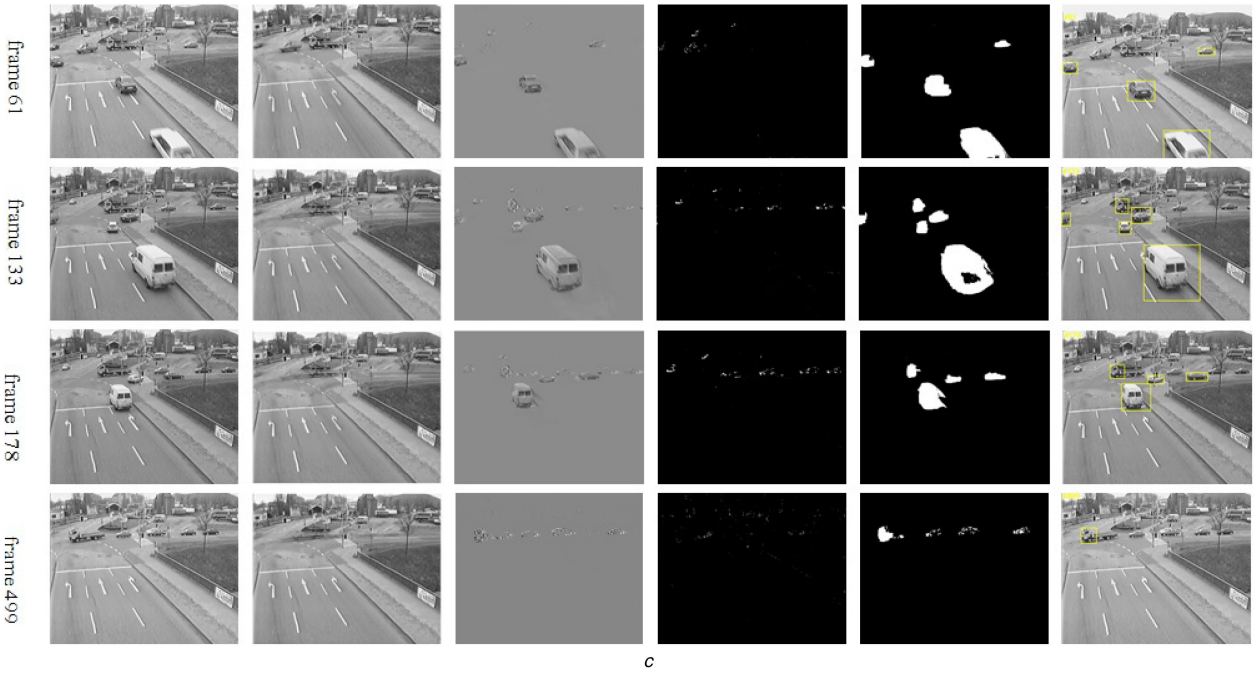

Fig. 2 *Results of the proposed method on three test sequences*
(a) Streetlight sequence, ***(b)*** HighwayII sequence, ***(c)*** Rheinhafen sequence

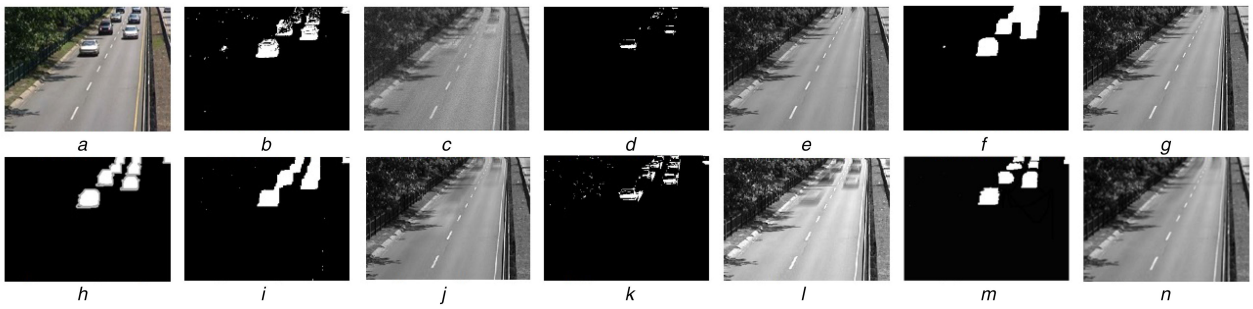

Fig. 3 *Qualitative comparison results of the six background estimation models on highway sequence. From left to right:*
(a) Input, ***(b)*** and ***(c)*** Foreground (FG) and background (BG) of MOG, ***(d)*** and ***(e)*** Foreground (FG) and background (BG) of GoDec, ***(f)*** and ***(g)*** Foreground (FG) and background (BG) of Decolor, ***(h)*** Ground truth (GT) for the foreground, ***(i)*** and ***(j)*** Foreground (FG) and background (BG) of Corola, ***(k)*** and ***(l)*** Foreground (FG) and background (BG) of incPCP, ***(m)*** and ***(n)*** Foreground (FG) and background (BG) of our method

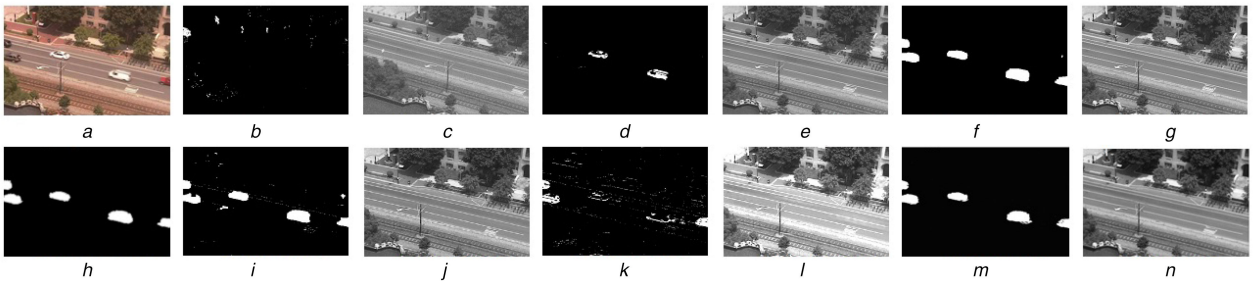

Fig. 4 *Comparative results of the six background estimation models on intermittentPan sequence*
(a) Input, ***(b)*** MOG FG, ***(c)*** MOG BG, ***(d)*** GoDec FG, ***(e)*** GoDec BG, ***(f)*** Decolor FG, ***(g)*** Decolor BG, ***(h)*** Ground truth, ***(i)*** Corola FG, ***(j)*** Corola BG, ***(k)*** incPCP FG, ***(l)*** incPCP BG, ***(m)*** Our FG, ***(n)*** Our BG

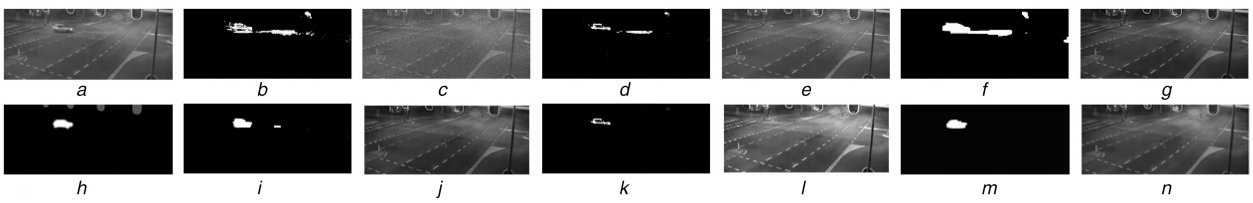

Fig. 5 *Comparative results of the six background estimation models on Streetcorneratnight sequence*
(a) Input, ***(b)*** MOG FG, ***(c)*** MOG BG, ***(d)*** GoDec FG, ***(e)*** GoDec BG, ***(f)*** Decolor FG, ***(g)*** Decolor BG, ***(h)*** Ground truth, ***(i)*** Corola FG, ***(j)*** Corola BG, ***(k)*** incPCP FG, ***(l)*** incPCP BG, ***(m)*** Our FG, ***(n)*** Our BG

© The Institution of Engineering and Technology 2017

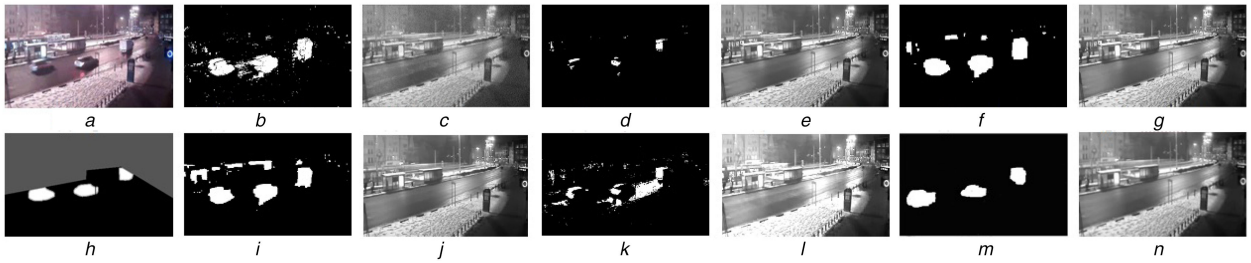

Fig. 6 *Comparative results of the six background estimation models on tramStation sequence*
(a) Input, ***(b)*** MOG FG, ***(c)*** MOG BG, ***(d)*** GoDec FG, ***(e)*** GoDec BG, ***(f)*** Decolor FG, ***(g)*** Decolor BG, ***(h)*** Ground truth, ***(i)*** Corola FG, ***(j)*** Corola BG, ***(k)*** incPCP FG, ***(l)*** incPCP BG, ***(m)*** Our FG, ***(n)*** Our BG

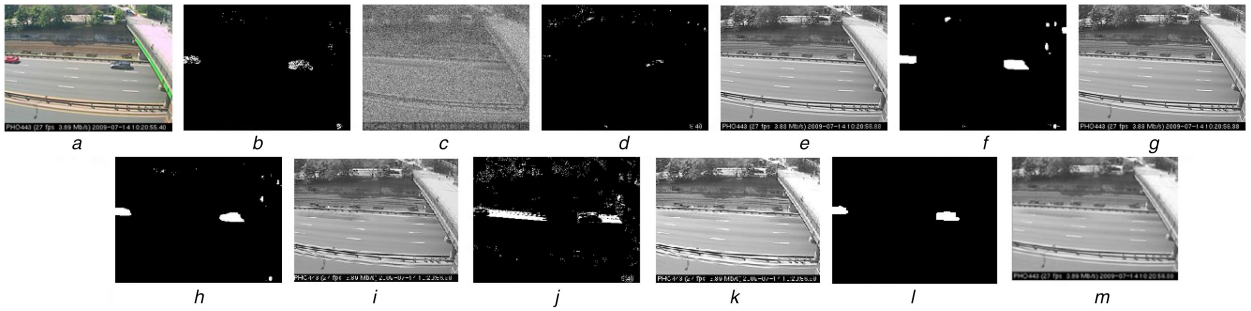

Fig. 7 *Comparative results of the six background estimation models on Streetlight sequence*
(a) Original image, ***(b, c)*** Results of MOG for detected foreground (FG) and estimated background (BG), ***(d, e)*** Results of GoDec for detected foreground (FG) and estimated background (BG), ***(f, g)*** Results of Decolor for detected foreground (FG) and estimated background (BG), ***(h, i)*** Results of Corola for detected foreground (FG) and estimated background (BG). Columns ***(j, k)*** Show the detected foreground (FG) and estimated background (BG) of incPCP, ***(l, m)*** Results of the proposed method for detected foreground (FG) and estimated background (BG)

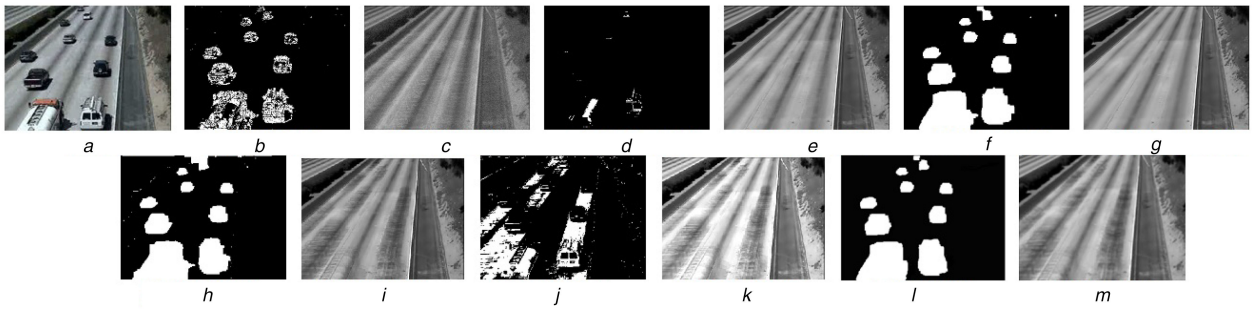

Fig. 8 *Comparative results of the six background estimation models on highwayII sequence*
(a) Input, ***(b)*** MOG FG, ***(c)*** MOG BG, ***(d)*** GoDec FG, ***(e)*** GoDec BG, ***(f)*** Decolor FG, ***(g)*** Decolor BG, ***(h)*** Corola FG, ***(i)*** Corola BG, ***(j)*** incPCP FG, ***(k)*** incPCP BG, ***(l)*** Our FG, ***(m)*** Our BG

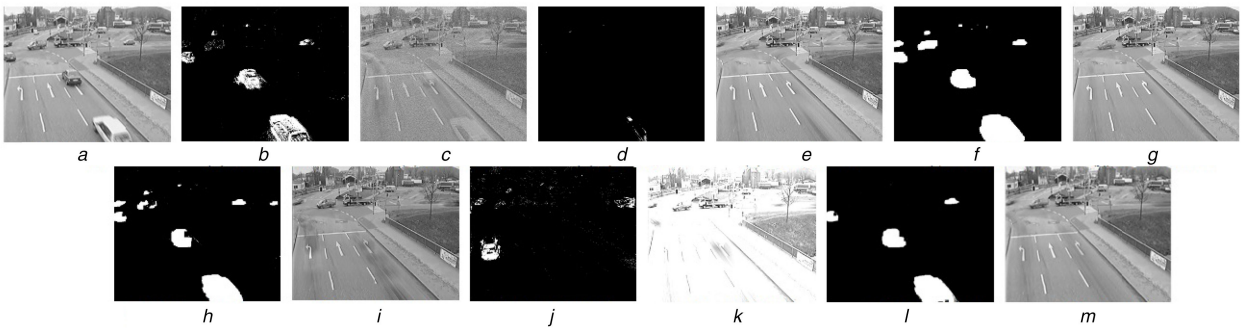

Fig. 9 *Comparative results of the six background estimation models on rheinhafen sequence*
(a) Input, ***(b)*** MOG FG, ***(c)*** MOG BG, ***(d)*** GoDec FG, ***(e)*** GoDec BG, ***(f)*** Decolor FG, ***(g)*** Decolor BG, ***(h)*** Corola FG, ***(i)*** Corola BG, ***(j)*** incPCP FG, ***(k)*** incPCP BG, ***(l)*** Our FG, ***(m)*** Our BG

GRAM road-traffic monitoring dataset [45], M-30 (800×480) and M-30 HD (1200×720). Both are colour videos of a highway under different conditions (sunny or cloudy) to compare the counting performance of our work with the state-of-the-art methods, as shown in Fig. 11 and Table 3. In addition, we also present the vehicle counting results of our method in test videos from CDnet2014 and Youtube (Table 1), as shown in Table 4.

From Fig. 11 and Table 3, we can see that the proposed method can effectively obtain the background model and foreground detection results, and achieve a comparable vehicle counting results with the state-of-the-art methods. The precision of our counting results in M-30 and M-30 HD datasets is 5.2 and 4.7% lower than that of [27], and 2.6 and 9.5% higher than that of [44]. However, the vehicle counting precisions of the proposed method

© The Institution of Engineering and Technology 2017

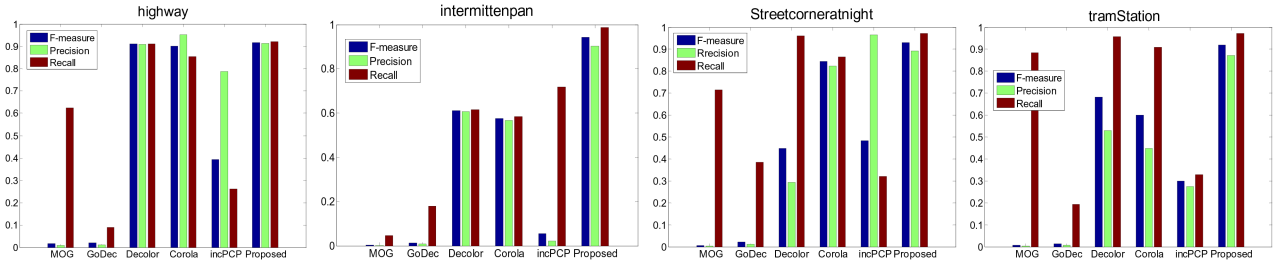

Fig. 10 *Evaluation results for test sequences*

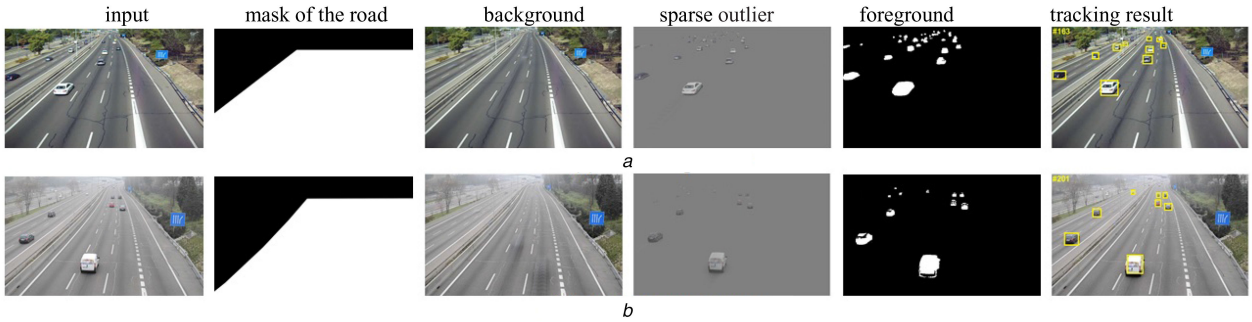

Fig. 11 *Background model and vehicle detection results of the proposed method on GRAM dataset*
(a) M-30 sequence, ***(b)*** M-30-HD sequence

in GRAM dataset exceed 88% in all sequences. Our vehicles counting precision is larger than 84% in CDnet2014 and Youtube video sequences, as shown in Table 4. Those experimental results prove the good performance of the proposed method on vehicle counting.

Table 4 gives the counting results of the proposed method on CDnet2014 and Youtube video sequences. From the vehicle counting results, as shown in Tables 3 and 4, we can see that the proposed method may miss-count some vehicles. It is worth mentioning that our method does not count one single vehicle twice since it uses the multi-object tracking algorithm. However, it may miss-count some vehicles because of the failure of Kalman tracking in Section 4. We count foreground detection as a vehicle only after it is tracked for several consecutive frames (five consecutive frames in this paper). The foundation for this assumption is that the noisy detections may generate several short-lived trackers. With this assumption, we can effectively reduce the noise in some extent in complex traffic scene.

Vehicle counting is a key step for traffic parameters estimation. We test the proposed method on daytime, nighttime, crowd and intermittent object motion scenes. Although the proposed method can online process one frame at a time, it still has a handful of batch initialisation, which tends to have some delay in real applications. As known, vehicle occlusion is a common concern in real traffic scenarios. In our work, we mainly focus on how to obtain the robust background model and achieve more accurate foreground detections. In addition, we rely on the multi-object tracking to track the foreground moving targets and counting

Table 1 Information of the sequences used in detection evaluation

Dataset	Sequence	Size×Frames	Description
CDnet2014	highway	$[320 \times 240] \times 600$	sunny day, shadows and waving trees
	intermittentPan	$[560 \times 368] \times 200$	sunny day, waving trees
	Streetcorneratnight	$[595 \times 245] \times 200$	night scene, light changes
	tramStation	$[480 \times 295] \times 300$	night scene, light changes
	streetlight	$[320 \times 240] \times 500$	sunny day, waving trees
youtube	highwayII	$[320 \times 240] \times 500$	crowed scene, occlusion
	Rheinhafen	$[320 \times 240] \times 500$	intermittent object motion

Table 2 Comparison of the metrics for test sequences

Sequence	Highway			Intermittenpan			Streetcorneratnight			TramStation		
methods	F_measure	Precision	Recall	F_measure	Precision	Recall	F_measure	Precision	Recall	F_measure	Precision	Recall
MOG	0.0175	0.0089	0.6247	0.0025	0.0013	0.0466	0.0057	0.0029	0.7157	0.0082	0.0041	0.885
GoDec	0.0205	0.0115	0.0898	0.0129	0.0067	0.1786	0.0214	0.011	0.3844	0.0140	0.0073	0.1937
Decolor	0.9099	0.9092	0.9106	0.6106	0.6061	0.6152	0.4483	0.2923	0.9611	0.6818	0.5292	0.9578
Corola	0.9008	0.9519	0.8549	0.5757	0.566	0.5850	0.8442	0.8244	0.8650	0.6003	0.4479	0.9101
incPCP	0.3934	0.7879	0.2620	0.0535	0.02174	0.7174	0.4824	0.96656	0.3215	0.2996	0.2751	0.3290
Our	0.9171	0.9131	0.9211	0.9437	0.9028	0.9884	0.9303	0.8917	0.9725	0.9191	0.8719	0.9717

Table 3 Vehicle counting results for competing methods on GRAM dataset

Sequence	True number of vehicle	Proposed method			Quesada *et al.* [27]			Bouvie *et al.* [44]		
		Count	Diff	Precision, %	Count	Diff	Precision, %	Count	Diff	Precision, %
M-30	77	71	10	92.2	75	2	97.41	69	8	89.62
M-30 HD	42	37	5	88.1	39	3	92.86	33	9	78.57

© The Institution of Engineering and Technology 2017

Table 4 Vehicle counting results on CDnet2014 and Youtube video sequences

Sequence	Actual number of vehicle	Number of detection vehicle	Number of counting vehicle	Number of miss detection vehicle	Percentage of correctly counted vehicle	Percentage of correctly detection vehicle, %
highway	16	15	14	1	93.3%	93.75
intermittentPan	15	15	15	1	93.33%	93.3
streetcorneratnight	21	21	19	0	90.48	100
tramStation	13	14	11	1	84.6%	92.3
streetlight	19	18	17	1	89.47%	94.74
highwayII	91	88	84	3	92.31%	96.7
Rheinhafen	17	15	14	2	82.35%	88.26

vehicles based on the tracking results. The tracking algorithms may cause miss-counting in some sequences, especially in the case of similar appearances or heavy occlusions. The heavy traffic and congestion situation in urban traffic scenarios often pose more challenges for the proposed method to achieve accurately vehicle detection, tracking and counting results. Therefore, one of our further researches is to learn how to improve the proposed method to eliminate the time delay, in particular removing or accelerating the batch initialisation step, as well as to pay more attention on occlusion analysis.

6 Conclusion

In this paper, a real-time vehicle detection and counting method based on sparse and low-rank approximation is proposed. Sparse representation and low-rank background modelling provide a powerful tool for object detection. According to our experimental results, the proposed sparse and low-rank-based vehicle detection method achieves a satisfied result. The key to this success relies on the accurate foreground detection and background modelling. Foreground detection results in sequential frames are passed to the online Kalman filter, which can build a reliable tracklet for each vehicle. This is beneficial to accurately count the number of vehicles and avoid double counting. In addition, it is useful to reduce the noise since noisy detections tend to result in short-lived trackers. Experimental results on Change Detection benchmark 2014 and Youtube sequences, as well as GRAM dataset, demonstrate the good performance of the proposed method in vehicles detection and counting.

7 Acknowledgments

This work was supported by the China Astronautic Science and Technology Innovation Foundation under grant no. CASC201104, China Aviation Science Fund Project under grant no. 2012ZC53043. The authors thank the valuable comments from the reviewers and editors.

8 Reference

[1] Zhang, J.P., Wang, K.F., Lin, W.H., *et al.*: 'Data-driven intelligent transportation systems: a survey', *IEEE Trans. Intell. Transp. Syst.*, 2011, **12**, (4), pp. 1624–1639

[2] Atiq, H.M., Farooq, U., Ibrahim, R., *et al.*: 'Vehicle detection and shape recognition using optical sensors: a review'. Second Int. Conf. Machine Learning and Computing (ICMLC), Bangalore, India, February 2010, pp. 223–227

[3] Liu, Y., Tian, B., Chen, S., *et al.*: 'A survey of vision-based vehicle detection and tracking techniques in ITS'. IEEE Int. Conf. Vehicular Electronics and Safety (ICVES), Dongguan, China, July 2013, pp. 72–77

[4] Ren, J.Q., Chen, Y.Z.: 'Multiple objects parameter detection in urban mixed traffic scene', *J. Transp. Inf. Safe*, 2009, **27**, pp. 47–54

[5] Abdagic, A., Tanovic, O., Aksamovic, A., *et al.*: 'Counting traffic using optical flow algorithm on video footage of a complex crossroad'. Conf. Record of IEEE Int. Conf. Electronics in Marine (ELMAR), Zadar, Croatia, October 2010, pp. 41–45

[6] Unzueta, L., Nieto, M., Cortes, A., *et al.*: 'Adaptive multi-cue background subtraction for robust vehicle counting and classification', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (2), pp. 527–540

[7] Wu, K., Xu, T., Zhang, H.: 'Overview of video-based vehicle detection technologies'. 6th Int. Conf. Computer Science & Education (ICCSE), Singapore, Singapore, August 2011, pp. 821–825

[8] Held, D., Levinson, J., Thrun, S.: 'A probabilistic framework for car detection in images using context and scale'. IEEE Int. Conf. Robotics and Automation (ICRA), Saint Paul, USA, June 2012, pp. 1628–1634

[9] Tsai, L.W., Hsieh, J.W., Fan, K.C.: 'Vehicle detection using normalized color and edge map', *IEEE Trans. Image Process.*, 2007, **16**, (3), pp. 850–864

[10] Jia, Y., Zhang, C.: 'Front-view vehicle detection by Markov chain Monte Carlo method', *Pattern Recognit.*, 2009, **42**, (3), pp. 313–321

[11] Li, Y., Wang, F.Y., Li, B., *et al.*: 'A multi-scale model integrating multiple features for vehicle detection'. Int. IEEE Conf. Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, May 2013, pp. 399–403

[12] Kim, K., Chalidabhongse, T., Harwood, D., *et al.*: 'Background modeling and subtraction by codebook construction'. Int. Conf. Image Processing (ICIP), Singapore, April 2004, pp. 3061–3064

[13] Zhang, Y., Zhao, C., He, J., *et al.*: 'Vehicles detection in complex urban traffic scenes using Gaussian mixture model with confidence measurement', *IET Intell. Transp. Syst.*, 2016, **10**, (6), pp. 445–452

[14] Stauffer, C., Grimson, W.: 'Adaptive background mixture models for real-time tracking'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Fort Collins, USA, June 1999, pp. 246–252

[15] Elgammal, A., Harwood, D., Davis, L.: 'Non-parametric model for background subtraction'. 6th European Conf. Computer Vision (ECCV), Dublin, Ireland, June 2000, pp. 751–767

[16] Matsuyama, T., Ohya, T., Habe, H.: 'Background subtraction for non-Stationary scenes'. Proc. Asian Conf. Computer Vision (ACCV), Taipei, Taiwan, January 2000, pp. 662–667

[17] Rittscher, J., Kato, J., Joga, S., *et al.*: 'A probabilistic background model for tracking'. Proc. European Conf. Computer Vision (ECCV), Dublin, Ireland, June 2000, pp. 336–350

[18] Monnet, A., Mittal, A., Paragios, N., *et al.*: 'Background modeling and subtraction of dynamic scenes'. Int. IEEE Conf. Computer Vision (ICCV), Nice, France, October 2003, pp. 1305–1312

[19] Kamkar, S., Safabakhsh, R.: 'Vehicle detection, counting and classification in various conditions', *IET Intell. Transp. Syst.*, 2016, **10**, (6), pp. 406–413

[20] Lempitsky, V., Zisserman, A.: 'Learning to count objects in images'. 24th Annual Conf. Neural Information Processing Systems (NIPS), Vancouver, Canada, December 2010, pp. 1324–1332

[21] Liu, X., Wang, Z., Feng, J., *et al.*: 'Highway vehicle counting in compressed domain'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, June 2016, pp. 3016–3024

[22] Rabaud, V., Belongie, S.: 'Counting crowded moving objects'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), New York, USA, June 2006, pp. 705–711

[23] Zhang, C., Li, H., Wang, X., *et al.*: 'Cross-scene crowd counting via deep convolutional neural networks'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Boston, USA, June 2015, pp. 833–841

[24] Dollar, P., Wojek, C., Schiele, B., *et al.*: 'Pedestrian detection: an evaluation of the state of the art', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (4), pp. 743–761

© The Institution of Engineering and Technology 2017

[25] Zhang, Y., Zhao, C., Zhang, Q.: 'Counting vehicles in urban traffic scenes using foreground time-spatial images', *IET Intell. Transp. Syst.*, 2017, **11**, (2), pp. 61–67
[26] Barcellos, P., Bouvié, C., Escouto, F., *et al.*: 'A novel video based system for detecting and counting vehicles at user-defined virtual loops', *Expert Syst. Appl.*, 2015, **42**, (4), pp. 1845–1856
[27] Quesada, J., Rodriguez, P.: 'Automatic vehicle counting method based on principal component pursuit background modeling'. IEEE Int. Conf. Image Processing (ICIP), September 2016, Phoenix, USA, pp. 3822–3826
[28] Rodriguez, P., Wohlberg, B.: 'Incremental principal component pursuit for video background modeling', *J. Math. Imaging Vis.*, 2016, **55**, (1), pp. 1–18
[29] Candes, E., Li, X., Ma, Y., *et al.*: 'Robust principal component analysis?', *J. ACM*, 2009, **58**, (3), pp. 1–39
[30] Wright, J., Peng, Y., Ma, Y., *et al.*: 'Robust principal component analysis: exact recovery of corrupted low-rank matrices by convex optimization'. Proc. Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, December 2009, pp. 3–20:56
[31] Feng, J., Xu, H., Yan, S.: 'Online robust PCA via stochastic optimization'. Proc. Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, December 2013, pp. 404–412
[32] Tianyi, Z., Tao, D.: 'Godec: randomized low-rank and sparse matrix decomposition in noisy case'. Proc. 28th Int. Conf. Machine Learning (ICML), Bellevue, Washington, USA, July 2011, pp. 33–40
[33] Zhou, X., Yang, C., Yu, W.: 'Moving object detection by detecting contiguous outliers in the low-rank representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (3), pp. 597–610
[34] Shakeri, M., Zhang, H.: 'COROLA: a sequential solution to moving object detection using low-rank approximation', *Comput. Vis. Image Underst.*, 2016, **146**, (1), pp. 27–39
[35] Bouwmans, T., Zahzah, E.H.: 'Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance', *Comput. Vis. Image Underst.*, 2014, **122**, (4), pp. 22–34
[36] Zhang, Y., Zhao, C., He, J., *et al.*: 'Vehicles detection in complex urban traffic scenes using a nonparametric approach with confidence measurement'. Int. Conf. Workshop on Computing and Communication, Vancouver, Canada, October 2015, pp. 1–7
[37] Cui, X., Huang, J., Zhang, S., *et al.*: 'Background subtraction using low rank and group sparsity constraints'. European Conf. Computer Vision (ECCV), Florence, Italy, October 2012, pp. 612–625
[38] Li, S.: '*Markov random field modeling in image analysis*' (Springer-Verlag New York Inc., Secaucus, NJ, USA, 2001)
[39] http://vision.csd.uwo.ca/code/
[40] Boykov, Y., Veksler, O., Zabih, R.: 'Fast approximate energy minimization via graph cuts', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **11**, pp. 1222–1239
[41] Ahuja, R., Magnanti, T., Orlin, J.: '*Network flows*' (Prentice-Hall, 1988)
[42] Zou, X., Li, D., Liu, J.: 'Real-time vehicles tracking based on Kalman filter in an ITS'. Proc. Int. Symp. Photoelectron Detection Image, Beijing, China, September 2007, pp. 662–666
[43] Wang, Y., Jodoin, P.M., Porikli, F., *et al.*: 'CDnet 2014: an expanded change detection benchmark dataset'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Columbus, USA, June 2014, pp. 393–400
[44] Bouvie, C., Scharcanski, J., Barcellos, P., *et al.*: 'Tracking and counting vehicles in traffic video sequences using particle filtering'. IEEE Int. Instrumentation and Measurement Technology Conf., May 2013, pp. 812–815
[45] Guerrero-Gomez-Olmedo, R., Lopez-Sastre, R.J., Maldonado-Bascon, S., *et al.*: 'Vehicle tracking by simultaneous detection and viewpoint estimation'. Int. Work Conf. Interplay between Natural and Artificial Computation, 2013, Part II (LNCS, **7931**), pp. 306–316

© The Institution of Engineering and Technology 2017