Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3515512, 14 pages
https://doi.org/10.1155/2021/3515512

Research Article

Traffic Foreground Detection at Complex Urban Intersections Using a Novel Background Dictionary Learning Model

Qianxia Cao,[1] **Zhengwu Wang**,[2] **and Kejun Long**[3]

[1]*Key Laboratory of Highway Engineering of Ministry of Education, Changsha University of Science and Technology, Changsha 410114, China*
[2]*School of Traffic & Transportation Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[3]*Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems, Changsha University of Science and Technology, Changsha 410114, China*

Correspondence should be addressed to Qianxia Cao; qianxiacao@gmail.com

Received 26 August 2021; Accepted 30 October 2021; Published 11 November 2021

Academic Editor: Xinqiang Chen

In complex urban intersection scenarios, due to heavy traffic and signal control, there are many slow-moving or temporarily stopped vehicles behind the stop lines. At these intersections, it is difficult to extract traffic parameters, such as delay and queue length, based on vehicle detection and tracking due to the dense and severe occlusion of vehicles. In this study, a novel background subtraction algorithm based on sparse representation is proposed to detect the traffic foreground at complex intersections to obtain traffic parameters. By establishing a novel background dictionary update model, the proposed method solves the problem that the background is easily contaminated by slow-moving or temporarily stopped vehicles and therefore cannot obtain the complete traffic foreground. Using the real-world urban traffic videos and the PV video sequences of *i*-LIDS, we first compare the proposed method with other detection methods based on sparse representation. Then, the proposed method is compared with other commonly used traffic foreground detection models in different urban intersection traffic scenarios. The experimental results show that the proposed method performs well in keeping the background model being unpolluted from slow-moving or temporarily stopped vehicles and has a good performance in both qualitative and quantitative evaluations.

1. Introduction

Real-time traffic data collected at intersections are essential information for intelligent traffic control. Compared with the continuous traffic flow at road sections, the traffic flow at intersections is interrupted due to the influence of traffic signals or traffic signs. The traffic data such as traffic volume, vehicle speed, and time occupancy cannot reflect the inherent traffic state characteristics at intersections. Some more suitable parameters can better reflect the traffic operation state of the intersection, such as control delay, queue length, saturation degree, and number of stops, but the parameters are difficult to be directly captured by the traditional sensors.

At present, many urban intersections are equipped with the traffic video surveillance systems to obtain traffic parameters that can directly reflect the traffic state information. Existing methods based on traffic videos for obtaining traffic parameters are mainly divided into two categories including the virtual coil or virtual line method and the image zone method. The virtual coil or virtual line method is to set a virtual coil or virtual line at a specified position on the road. When vehicles pass through the virtual coil area or virtual line, traffic parameters can be obtained by analyzing image feature changes. The image zone method is to obtain traffic parameters by vehicle detecting and tracking [1, 2]. Recently, deep learning has been successfully applied in object detection and tracking, such as the faster region-based convolutional neural network (R-CNN) [3], You Only Look Once (YOLO) [4], and single shot multibox detector (SSD) [5]. However, with the increasing traffic volume at the urban intersection, vehicle detective and tracking methods are increasingly affected by some factors, such as occlusion, slow-moving, or temporarily stopped vehicles. In order to

reduce the complexity of algorithms caused by overcoming environmental disturbances, methods based on visual features for obtaining traffic state parameters are receiving increasing attention.

Traffic foreground detection is the first step in the traffic parameter extraction based on videos. However, due to the influence of traffic signals, the traffic flow at the intersection periodically queues and dissipates behind the stop line, resulting in many slow-moving or temporarily stopped vehicles. The robust foreground detection at urban intersections faces enormous challenges. Existing foreground detection methods mainly include the background subtraction method, optical flow method, and interframe subtraction method. Both the optical flow method and the interframe subtraction method are suitable for detecting the moving foreground. However, there are both moving vehicles and queuing vehicles at urban intersections. If the queuing vehicles are not detected, this may result in incomplete foreground detection. Background subtraction is an effective technique for detecting queuing vehicles and moving vehicles at urban intersections. For the BS methods, the background models are first established, and then, the foregrounds are obtained by comparing the subtraction between the current frame and the background frame [6]. The existed BS methods can be roughly divided into the parametric background model and nonparametric background model. The simplest parametric background modeling method is based on a statistical analysis of the histogram values for each pixel of the past K frames, and the mean and median or the maximum frequency is used to estimate the background [7]. Another parametric background method is to establish a single-peak probability density function for background pixels, such as running Gaussian average [8]. Since a single Gaussian density function cannot handle dynamic background scenarios, the Gaussian mixture model (GMM) composed of N Gaussian component is established for each pixel to estimate the background [9]. Many improved GMM have been proposed for detecting foreground objects in traffic scenarios. To simplify the calculation to increase the operation speed, an image block-based GMM background model is constructed [10]. In addition, the expectation-maximization (EM) algorithm is fused with the Gaussian mixture model for improving the segmentation quality of moving vehicles [11]. The Gaussian mixed model and the previous methods are very effective for continuous variable scenarios, but dynamic scenarios with fast nonstationary changes cannot be accurately described by a set of Gaussian functions. The nonparametric background modeling method is more suitable for the case where the density function is more complicated or cannot be parametrically modeled. The kernel density estimation is a nonparametric method by estimating the background probability of each pixel from the most recent multiframe sequence [12]. Another nonparametric method is the codebook model, which is a background modeling method based on pixel color. The method uses a quantization technique to create a codebook based on the color distance and brightness of each pixel in the images sequence. An improved codebook algorithm is proposed to achieve better results than the original codebook model and the Gaussian mixed model; however, this method could not handle slow-moving or temporarily stopped vehicles at intersections [13]. Visual background extractor (ViBe) is an algorithm for the motion detection by background subtraction. It is a very fast algorithm, based on samples and several innovative processes, including time subsampling, random substitution, and spatial diffusion [14]. The sigma-delta filtering algorithm uses the sigma-delta filter for background estimation and foreground detection to achieve computational efficiency and low memory consumption [15].

Recently, the sparse representation [16] has been successfully applied in background detection. The methods under this framework follow that the background in the video sequence is modeled by a low-rank matrix and the moving object corresponds to a sparse outlier. In [17], the k-means classifier is used to train the dictionary, and the matching pursuit algorithm is used to obtain the sparse coefficients; then, the sparse linear combination is used to estimate the background. The method is simple and can obtain good preliminary detection results. To effectively deal with the dynamic changes of background illumination and environment, background modeling is performed in combination with K-SVD (singular value decomposition) training dictionary and average sparse coefficients [18]. Fixed dictionaries are used for background modeling but do not reflect background changes in dynamic scenarios [19]. In [20], the dynamic adaptive update dictionary is used for background modeling, and the background subtraction is performed by the sparse reconstruction error of the current image and the background image. Yang and Qu [21] proposed a real-time vehicle detection and counting in complex traffic scenarios using the background subtraction model with low-rank decomposition. These methods provide good performance for background modeling. However, the background is easily contaminated by slow-moving or temporarily stopped vehicles at intersections, resulting in the missing foreground detection. To solve this problem, Toral et al. and Manzanera and Richefeu developed a sigma-delta model with confidence measurement (sigma-delta with CM) to detect vehicles in urban traffic scenarios [22, 23]. Instead of the sigma-delta model, Zhang et al. used GMM with confidence measurement for each pixel to efficiently resolve deficiencies in the background subtraction model [24, 25]. The methods have achieved good results at simple intersections with small traffic volumes.

In this study, aiming at the traffic foreground detection at complex urban intersections, we do the following work:

(1) We have established a novel background dictionary update model, which automatically removes foreground information from the background when updating the background dictionary, preventing the background from being contaminated by slow-moving or temporarily stopped vehicles at complex intersections

(2) The independence of sparse representation leads to large differences in the sparse representation of

similar feature blocks of the same foreground object. We introduce a manifold regularity term to build an adaptive sparse coding model for obtaining a continuous and consistent representation of the foreground object.

(3) The traffic foreground detection is obtained based on the feature reconstruction error

This study is organized as follows. Related works are summarized in Section 2. Section 3 describes the details of the proposed method for background estimation and traffic foreground detection at the intersection. Experimental results are discussed in Section 4, and conclusion is drawn in Section 4.

2. Materials and Methods

The overview of our proposed method is shown in Figure 1. First, we extract some image frames from video sequences of the urban intersection to form a training sample set. Each image is divided into blocks, and some blocks of each image are randomly extracted to form a subset for training the background dictionary. Then, to avoid the background pollution by the slow-moving or temporarily stopped vehicles at the intersection, we establish a background dictionary update model to limit the foreground update to the background dictionary. Due to the independence of the sparse representation, the sparse representation coefficients of different image blocks of the same object may be very different, resulting in discontinuities in the foreground object representation. To obtain a more accurate background update dictionary, we introduce a manifold regularity term to build an adaptive sparse coding model for obtaining a continuous and consistent representation of the foreground object. Finally, for any frame of the video set, the sparse reconstruction error of the current frame and the background is used for foreground detection.

The proposed BS method is based on two assumptions. The first one is that the backgrounds of an arbitrary scenario are linearly correlated with each other and can be sparsely and linearly represented by the atoms of the dictionary. The foreground is observed by a sparse and contiguous piece in consecutive frames, leading to the changing of the background, and greatly transforms the projection over the dictionary [16]. The second one is that locations of the foreground in successive frames are likely to be grouping together instead of randomly scattering. This indicates that the foreground objects satisfy the structured sparsity constraint [26].

2.1. Initial Background Dictionary Learning. According to assumption one, we formulate the BS problem by linearly decomposing the input image X into a low-rank matrix X_B and a binary sparse matrix X_F:

$$X = X_B + X_F, \tag{1}$$

where X is the video image frame, X_F is the foreground candidate, and X_B is the background model.

The background model is generally linearly represented by the background dictionary D_b and the sparse coefficient α [27]:

$$X_B = D_b \times \alpha. \tag{2}$$

The initial background dictionary D_b needs to be trained first. Most of the existing methods are to manually extract clean background frames from video sequences or use the first N frames in the video sequence as the training set. However, in complex intersection scenes, especially at intersections with large traffic volumes, it is difficult to obtain clean background frames without foreground. So, we use actual video frames directly to train the dictionary. To avoid the background dictionary being contaminated by the foreground in video images, we randomly extract training images from video sequences when the intersection traffic volume is small. Then, each image is segmented into blocks, and blocks are randomly extracted from each image to form a set of sample blocks for background training. Caused by the temporarily stopped or slow-moving vehicles problem at intersections, we select training images at a specific time interval instead of selecting every frame in the image sequence. The specific interval is determined according to the actual situation. When the moving speed of the foreground target at the intersection is slower, the interval between selected video frames is longer.

Given a video set, each of the collected images is divided into n nonoverlapping blocks. The blocks in the video frame are selected at a certain interval to form a training set X, and the dictionary D_b satisfies the following formula:

$$D_b = \arg\min_{D_b} \sum_{m=1}^{M} \left(x_m - D_b \alpha_{m2}^2 + \lambda_1 \alpha_{m1}\right), \tag{3}$$

where M is the number of sample blocks in the training set, x_m is the m^{th} block vector in the training set, α_m is the m^{th} sparse coefficient, and λ_1 is the regularization parameter. The trained dictionary is shown in Figure 2.

2.2. Background Dictionary Update Model Based on Foreground Uncorrelation. A fixed initial background dictionary does not adapt to the background changes in the video sequences. To effectively extract the foreground in video sequences, the background dictionary needs to be adaptively updated over time. However, in complex scenarios, such as temporarily stopped or slow-moving vehicles at intersections, the foreground will gradually merge into the background and become an element in the background dictionary. Therefore, the reconstruction error may be difficult to accurately describe the difference between the current image and the background, resulting in missing foreground detection.

To solve this problem, we propose a background dictionary update model with minimal correlation with the foreground. In the process of background dictionary updating, the background dictionary can learn the atoms by minimizing the correlation between the background dictionary and the foreground dictionary. In this way, when the

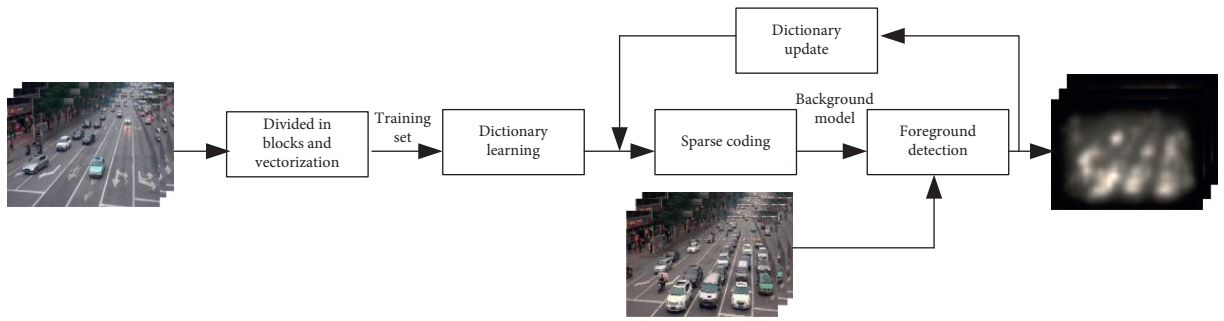

Figure 1: Overview of the proposed method.

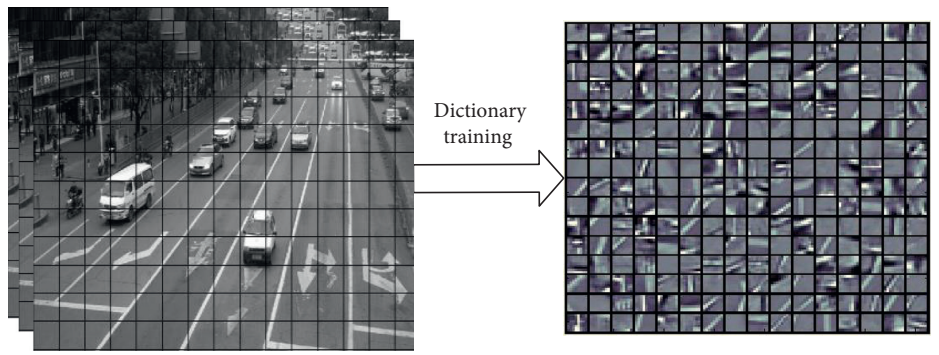

Figure 2: Initial background dictionary training. (a) Training set. (b) Training dictionary.

background is expressed in a sparse linear combination, the foreground is not merged into the background, resulting in a better foreground detection result.

2.2.1. Model Construction. Given a video frame $X = [x_1, x_2, \ldots, x_n] \in R^{m\times n}$, x_i is the ith block vector of the frame X. The background dictionary is $D_b = [b_1, b_2, \ldots, b_q] \in R^{m\times q}$, and the foreground dictionary is $D_f = [f_1, f_2, \ldots, f_p] \in R^{m\times p}$. The correlation between the background dictionary D_b and the foreground dictionary D_f is required to be as small as possible. That is, $D_f^T D_{bF}^2$ is as small as possible. Then, we formulate our optimization model as follows:

$$\begin{aligned} &\min_{D_b,\alpha} X - D_b\alpha_F^2 + \lambda_1\alpha_1 + \lambda_2 D_f^T D_{bF}^2 \\ &\text{s.t. } b_j^T b_j = 1, \quad j = 1, 2, \ldots, q, \end{aligned} \tag{4}$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N] \in R^{q\times N}$ is the sparse coefficient of the image block on the dictionary D_b, $\lambda_1$and λ_2 are the regularization parameter, and the regularization term $D_f^T D_{bF}^2$ is the cross correlation of D_b and D_f.

2.2.2. Model Solving. The model solution of (4) is the joint optimization problem between the background dictionary D_b and the representation coefficient α. If they both are variables, the optimization problem is nonconvex; if one is fixed, the optimization problem is transformed into a convex optimization problem. Learning from the alternate rotation optimization of the K-SVD algorithm, the solution steps are as follows:

Step 1: fixing dictionary D_b, (4) is

$$\min_{\alpha} X - D_b\alpha_F^2 + \lambda_1\alpha_1. \tag{5}$$

This is a standard lasso problem that can be solved using existing methods.

Step 2: fixing α, atoms in D_b are updated one by one. β_j represents the row vector of α, $j = 1, 2, \ldots, q$. Thus (4) can be reformulated as follows:

$$\begin{aligned} &\min_{b_l} X - \sum_{j\neq l}^{q} b_j\beta_j - b_l\beta_{lF}^2 + \lambda_2 \sum_{j\neq l}^{q} D_f^T b_{j2}^2 \\ &+ \lambda_2 D_f^T b_{l2}^2, \text{s.t. } b_l^T b_l \\ &= 1. \end{aligned} \tag{6}$$

Letting $Z = X - \sum_{j\neq l}^{q} b_j\beta_j$, and throwing away items that are not related to b_l, we can rewrite (6) as

$$\begin{aligned} &\min_{b_l} Z - b_l\beta_{lF}^2 + \lambda_2 D_f^T b_{l2}^2, \\ &\text{s.t. } b_l^T b_l = 1. \end{aligned} \tag{7}$$

Applying the Lagrange multiplier method to (7), then (7) can be presented by the following equivalent problem:

$$\min_{b_l} Z - b_l\beta_{lF}^2 + \lambda_2 D_f^T b_{l2}^2 + \gamma\left(1 - b_l^T b_l\right), \tag{8}$$

where γ is the Lagrange multiplier. Then, (8) can be rewritten by relaxing the Frobenius norm:

$$\begin{aligned}
&= \min_{b_l} \mathrm{Tr}\left\{ (Z - b_l\beta_l)^T (Z - b_l\beta_l) \right\} \\
&\quad + \lambda_2 \left\{ \left(D_f^T b_l\right)^T \left(D_f^T b_l\right) \right\} + \gamma\left(1 - b_l^T b_l\right) \\
&= \min_{b_l} \mathrm{Tr}\left\{ \left(Z^T Z - Z^T b_l\beta_l - \beta_l^T b_l^T Z + \beta_l^T b_l^T b_l\beta_l\right) \right\} \\
&\quad + \lambda_2\left(b_l^T D_f D_f^T b_l\right) - \gamma b_l^T b_l + \gamma.
\end{aligned} \tag{9}$$

Ignoring items that are not related to b_l, and using the correlation properties of symmetric matrices, we simplify (9) into the following formula:

$$\min_{b_l} \mathrm{Tr}\left\{-2Z\beta_l^T b_l^T\right\} + b_l^T\left\{\left(\beta_l\beta_l^T - \gamma\right)\mathrm{I} + \lambda_2 D_f D_f^T\right\} b_l + \gamma. \tag{10}$$

Deriving the above formula with respect to b_l, the outcome is

$$2\left[\left(\beta_l\beta_l^T - \gamma\right)\mathrm{I} + \lambda_2 D_f D_f^T\right] b_l - 2Z\beta_l^T. \tag{11}$$

Let the above formula be equal 0; then, the optimal solution is

$$b_l = \left[\left(\beta_l\beta_l^T - \gamma\right)\mathrm{I} + \lambda_2 D_f D_f^T\right]^{-1} Z\beta_l^T. \tag{12}$$

We can normalize b_l as

$$b_l = \frac{b_l}{b_{l2}}. \tag{13}$$

Updating each atom as described above, subsequent atoms are updated using the previously updated atom until the stop criterion is reached. For the determination of γ, it can be proved that $g(\gamma) = b_l(\gamma)^T b_l(\gamma)$ is a monotonic function about γ, and $g(\gamma) = 1$ has a unique solution. The dichotomy is used to solve γ, and the optimal solution D_b is obtained by substituting γ into formula (12).

2.2.3. Algorithm Convergence Analysis. The objective function formula (4) is monotonous under the update rule (12).

Let $\hat{b}_j$, $j = 1, 2, \ldots, l\text{-}1$ be the update value of the previous l-1 step; the update of the step l can be reformulated as

$$\begin{aligned}
F(b_l) &= Z_l - b_l\beta_{lF}^2 + \lambda_2 D_f^T b_{l2}^2 \\
&\quad + \lambda_2 \left\{ \sum_{j=1}^{l-1} D_f^T \hat{b}_{j2}^2 + \sum_{j=l+1}^{q} D_f^T b_{j2}^2 \right\},
\end{aligned} \tag{14}$$

where $Z = X - \sum_{j=l}^{l-1} \hat{b}_j\beta_j - \sum_{j=l+1}^{q} b_j\beta_j$. When $b_l = [(\beta_l^T\beta_l - \gamma)\mathrm{I} + \lambda_2 D_f D_f^T]^{-1} Z_l\beta_l^T$, $F(\hat{b}_l)$ is the minimum value. That is, $F(\hat{b}_l) \le F(b_l)$. Similarly, in the updating process of step $l+1$, $F(b_{l+1}) \le F(\hat{b}_l)$, and $F(\hat{b}_{l+1}) \le F(b_{l+1})$. Then, we can obtain

$$F\left(\hat{b}_{l+1}\right) \le F\left(b_{l+1}\right) \le F\left(\hat{b}_l\right) \le F\left(b_l\right). \tag{15}$$

Therefore, the objective function is monotonous. In the dictionary update phase, when the atom is updated one by one, the objective function is degraded.

2.3. Sparse Coding Based on Foreground Consistency Representation. For the standard sparse representation, when the dictionary is given, the sparse coefficient α_i is only affected by the input data x_i, regardless of other input data $x_{k(k\neq i)}$. That is, the sparse coefficients are independent of each other, and the correlation between the input data sequences is cut off. According to assumption two, image blocks belonging to the same foreground object have similar features in foreground object detection, and the corresponding sparse representations of the image blocks should be similar. However, the independence of the sparse representation causes a large difference in the sparse representation of similar image blocks. This difference leads to discontinuities and inconsistencies in the extracted foreground, further affecting the accuracy of the background dictionary update. Manifold learning focuses on constructing data relationships and displaying the intrinsic local structure between image blocks. The relationship is described by the local geometry between the data, and the Laplacian matrix is obtained and added as a constraint to the standard sparse representation matrix. In this way, the neighborhood geometry relationship in the original space remains unchanged in the sparse space, and the obtained sparse representation better reflects the original geometry of the data.

2.3.1. Model Construction. According to above, the following structural sparse representation model is established:

$$\alpha = \min_{\alpha} \sum_{i=1}^{N} \left(x_i - D_b\alpha_{i2}^2 + \lambda_1\alpha_{i1}\right) + \frac{\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i - \alpha_{j2}^2 W_{ij}, \tag{16}$$

where parameters β and λ_1 are the regularization parameters and W_{ij} is a spatial adjacency between blocks, $W_{ij} = \begin{cases} e^{-((\alpha_i - \alpha_j^2)/t)} & \text{if } \alpha_i \text{ is adjacent to } \alpha_j \\ 0 & \text{else} \end{cases}$, where t is a constant, and $W_{ij} = W_{ji}$. When $\alpha_i - \alpha_j^2 \le \varepsilon$, α_i and α_j are adjacent, when α_i is a point in the k neighborhood of α_j or α_j is a point in the k neighborhood of α_i.

2.3.2. Model Solving. When D_b is fixed in (16), the equation becomes a convex function. Because it is time-consuming to directly solve this convex function, we optimize α_i one by one until the entire α converges instead of optimizing all the columns in α at the same time. At each step of optimizing α_i, we fix the other columns of α ($j \neq i$) and rewrite (16) as

$$\min_{\alpha_i} J(\alpha_i) + \lambda_1\alpha_{i1}, \tag{17}$$

where

$$J(\alpha_i) = x_i - D_b\alpha_{i2}^2 + \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i - \alpha_{j2}^2 W_{ij}. \tag{18}$$

Equation (18) can be rewritten by relaxing the Frobenius norm:

$$\begin{aligned} J(\alpha_i) &= (x_i - D_b\alpha_i)^T(x_i - D_b\alpha_i) + \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_j)W_{ij}(\alpha_i - \alpha_j)^T \\ &= \left(x_i^T x_i - x_i^T D_b\alpha_i - \alpha_i^T D_b^T x_i + \alpha_i^T D_b^T D_b\alpha_i\right) + \beta\left(\sum_{i=1}^{N}\alpha_i\left(\sum_{j=1}^{N}W_{ij}\right)\alpha_i^T - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i W_{ij}\alpha_j^T\right), \end{aligned} \tag{19}$$

and removing the item unrelated to α_i, we can rewrite $J(\alpha_i)$ as

$$J(\alpha_i) = \left(-2\alpha_i^T D_b^T x_i + \alpha_i^T D_b^T D_b\alpha_i\right) + \beta\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i L_{ij}\alpha_j^T\right), \tag{20}$$

where $L_{ij} = D_{ii} - W_{ij}$, $D_{ii} = \sum_{j=1}^{N} W_{ij}$ are a diagonal matrix. We continue to separate α_k $(k \neq i)$ from α_i:

$$J(\alpha_i) = \left(-2\alpha_i^T D_b^T x_i + \alpha_i^T D_b^T D_b\alpha_i\right) + \beta\left(\alpha_i L_{ii}\alpha_i^T + \sum_{k\neq i}^{N}\alpha_i L_{ik}\alpha_k^T\right). \tag{21}$$

Then, formula (17) can be reformulated as

$$\min_{\alpha_i}\left(-2\alpha_i^T D_b^T x_i + \alpha_i^T D_b^T D_b\alpha_i\right) + \beta\left(\alpha_i L_{ii}\alpha_i^T + \sum_{k\neq i}^{N}\alpha_i L_{ik}\alpha_k^T\right) + \lambda_1\theta^T\alpha_i, \tag{22}$$

where $\theta \in \{-1, 0, 1\}$ is the sign of α_i. Deriving the above formula with respect to α_i, the outcome is

$$\frac{\partial J(\alpha_i)}{\partial\alpha_i} = 2\left(D_b^T D_b\alpha_i - D_b^T x_i + \beta L_{ii}\alpha_i + \beta\sum_{k\neq i}^{N}L_{ik}\alpha_k^T\right) + \lambda_1\theta^T. \tag{23}$$

Let the above formula be equal to 0; the optimal solution of (17) can be obtained as

$$\begin{aligned} \alpha_i &= \left(D_b^T D_b + \beta L_{ii}\mathrm{I}\right)^{-1}\left(D_b^T x_i - \beta\sum_{k\neq i}^{N}L_{ik}\alpha_k^T - \frac{\lambda_1\theta^T}{2}\right) \\ &= \Phi^{-1}\left(D_b^T x_i - \beta\Lambda_k L_{ik} - \frac{\lambda_1\theta^T}{2}\right), \end{aligned} \tag{24}$$

where $\Phi = B^T B + \beta L_{ii}\mathrm{I}$ and $\Lambda_k$is the submatrix after the i^{th} column is removed from the matrix α. L_{ik} is the subvector after the i^{th} element is removed from the vector L. I is the unit matrix. For speeding up the convergence of sparse coding, α is initialized by standard sparse coding.

2.4. Foreground Detection Based on Feature Reconstruction. The foreground detection in the video sequence is treated as a reconstruction error estimation process on the background dictionary. The foreground blocks and the background blocks are, respectively, projected on the same background dictionary, and the reconstruction error must be greatly different. Based on the above background dictionary model and the sparse coefficient calculation method, we may distinguish whether the image block belongs to the foreground block or belongs to the background block according to the reconstruction error of each image block. The larger the reconstruction error indicates the larger the difference between the image block and the background, and the higher the probability that the image block is foreground. For characterizing the reconstruction error well, the foreground detection model is constructed as follows:

$$y = \min_{\alpha}\sum_{i=1}^{N}\left(x_i - D_b\alpha_{i2}^2 + \lambda_1\alpha_{i1}\right) + \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i - \alpha_{j2}^2 W_{ij}. \tag{25}$$

The first item in (25) is the feature reconstruction error. If the background block is successfully represented by the background dictionary D_b and the sparse coefficient α, the reconstruction error is small. Since the foreground block has a sparse representation coefficient of almost zero on the dictionary D_b, the reconstruction error is relatively large. The second item in (25) is the regular item. Since some foregrounds are well reconstructed using many background dictionary atoms, the reconstruction error is not sufficient to measure foreground changes. Therefore, we introduce the regularization term to further reflect the change of the foreground. The last item shows that a larger coefficient difference in sparsity coefficients indicates the higher the probability of being a foreground.

Then, y is compared with the threshold T. If y is below the threshold T, y is a background block. Otherwise, y is a foreground block. Through finding a suitable threshold T, we can obtain the foreground of the target.

The proposed framework of the detection algorithm is given in Table 1. Given the current frame X_t (tth frame) and the current dictionary D_{t-1}, the sparse representation α_{t-1} is updated by (24), and the foreground detection result y_t is obtained according to (25). Then, the foreground dictionary is updated according to the foreground detection result y_t, and the background dictionary is updated according to (12). Then, the detection of the next frame X_{t+1} is performed.

For adapting to changes in environmental disturbances and lighting in the background, the background model needs to be dynamically updated over time. However, continuous sparse reconstruction leads to relatively large computational overhead, and if the background update is too frequent, it is easy to introduce error accumulation to affect the accuracy of foreground extraction. Because the background usually does not substantially change over a certain time interval, the background dictionary is usually updated every T frames instead of every frame performing a background update. The specific frame interval depends on the specific situation.

3. Results and Discussion

For verifying the performance of traffic foreground detection at complex intersections, firstly, we compare and evaluate our algorithm with other traffic foreground detection methods based on sparse representations (Section 3.1). Then, we compare and evaluate our algorithm with other typical traffic foreground detection methods in different intersection scenarios (Section 3.2). The test videos use the PV video sequences of i-LIDS [28] and the video sequences captured at actual complex intersections.

For quantitative evaluation, the metrics of precision, recall, and F-measure are used to show the overall accuracy of our method, defined as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F\,\text{measure} &= 2 \cdot \frac{(\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})}. \end{aligned} \tag{26}$$

where TP (true positive) is the number of pixels correctly classified as the foreground. FP (false positive) is the total number of background pixels that have been misclassified as foreground. FN (false negative) is the number of foreground pixels that have been misclassified as background. Precision gives the percentage of correctly detected foreground pixels among all detected foreground pixels. Recall weighs the percentage of foreground pixels that are correctly detected in the total number of foreground pixels. F-measure is the weighted harmonic mean of precision and recall, which measures the overall detection quality of the algorithm. For all three metrics, the higher the value is, the better the performance that it has is.

The algorithms are implemented in MATLAB and run on a desktop with a 2.4 GHz Core2 i5 processor and 4 GB memory.

3.1. Comparative Analysis of Foreground Detection Algorithms Based on Sparse Representation. We test our method on video clips captured at the actual complex intersection. Due to the influence of traffic signals, the traffic flow at the intersection periodically queues and dissipates behind the stop line, resulting in many slow-moving or temporarily stopped vehicles. Moreover, due to the heavy traffic at this intersection, many vehicles are waiting in line for long periods of time. The video resolution is 192×256 pixels.

Our method is compared with two representative foreground detection methods based on sparse representations. In our algorithm, the image block is the basic processing unit. The size of the image block has a certain impact on the processing speed and detection results. The image block size is smaller, the detection accuracy is higher, and the processing speed is slower. The image block size of our algorithm is selected as 8×8, and it takes 1.5 s~2.0 s to process one frame at this size.

Figures 3–5 show the comparison results of background extraction and foreground detection through the standard sparse representation method, the structured sparse representation method, and our proposed method. To save space, we only list three frames with different queuing lengths. In Figures 3–5, the first column is the original video image, the second column is the current background extracted by each method, and the third column is the foreground detection image. The results in Figures 3 and 4 show that the extracted backgrounds are contaminated to varying degrees due to slow-moving or temporarily stopped vehicles at the intersection. The detected foreground is missing because part of the foreground is degraded to the background. Moreover, in Figure 3, due to the independence of the sparse representation, the sparse representations of image blocks belonging to the same object are different, resulting in inconsistencies in background estimation and foreground detection. In Figure 4, the structured sparse representation improves the detection effect. Figure 5 shows that our method provides a significant improvement over other methods. With foreground unrelated limitations in the background updates, we can prevent the slow-moving or temporarily stopped vehicles from blending into backgrounds and recover the accurate backgrounds without smearing and ghosting artifacts. Moreover, due to the manifold regularity term, foreground consistency detection is better. Although the traffic volume at the intersection in the test video clips is heavy and the traffic scenarios are more complicated, our method can better meet the foreground detection requirements of complex intersections.

Table 2 and Figure 6 show the quantitative evaluation results by precision, recall, and F-measure on the test video clips. For the accuracy of the results, we only perform statistics on a road area where there are many slow-moving or temporarily stopped vehicles. Our method achieves the highest F-measure at complex urban intersections. For the

Table 1: Traffic foreground detection algorithm for intersection scenarios.

Input: current frame X_t; background dictionary D_{t-1};
Output: foreground detection result y_t; background update dictionary D_t; sparse representation α_t;
Initialization: initial background dictionaryD_b; initial foreground dictionaryD_f; parameters λ_1, λ_2 and β;
1. Sparse coding: with fixed D_{t-1}, the sparse coefficient α_{t-1} is updated by equation (24);
2. Foreground detection: y_t is updated by equation (25);
3. Foreground dictionary update: the foreground dictionary D_f is updated according to the foreground detection result y_t;
4. Background dictionary update: the dictionary D_b is updated according to formula (12) to get a new background dictionary D_t;
5. Return to the step 1 for the next frame detection.

Figure 3: Detection results of the standard sparse representation method.

standard sparse representation method (Method1) and the structured sparse representation method (Method2), the values of precision and recall show that the selected approaches perform the traffic foreground detection tasks with medium and low performance due to the slow-moving or temporarily stopped vehicles in the complex scenario. Compared with the standard sparse representation method (Method1), the structured sparse representation method (Method2) achieves the higher precision due to foreground consistency detection, but lower recall due to more false positives. The proposed method eliminates most false positives through the limitation of being irrelevant to the foreground in the background update, obtaining the best results.

3.2. Comparative Analysis of Detection Algorithm Performance in Different Intersection Scenarios. We further compare our algorithm with the GMM model and sigma-delta confidence model (proposed in [22]) typically used for traffic foreground detection at intersections. These algorithms are performed in simple intersection scenarios and complex intersection scenarios. A simple intersection scenario refers to a situation where the traffic environment is simple, the traffic volume is small, and the vehicle queuing time is short. A complex intersection scenario refers to a situation where the traffic environment is complex, the traffic volume is heavy, and the vehicle queuing time is relatively long. In the study, the simple intersection scenarios use the PV Hard video sequences of *i*-LIDS [28]. In the PV Hard scenarios, there are multiple slow-moving or temporarily stopped vehicles. The video resolution is 720×576 pixels. The complex intersection scenarios use the captured intersection video sequences. The video resolution is 192×256 pixels.

Figure 7 shows the results extracted by compared methods in simple intersection scenarios on the PV video sequences of *i*-LIDS. Figures 8 and 9 show the results extracted by compared methods in complex intersection scenarios on the captured video clips. Table 3 and Figure 10 show the quantitative evaluation results by precision, recall, and *F*-measure on the test videos.

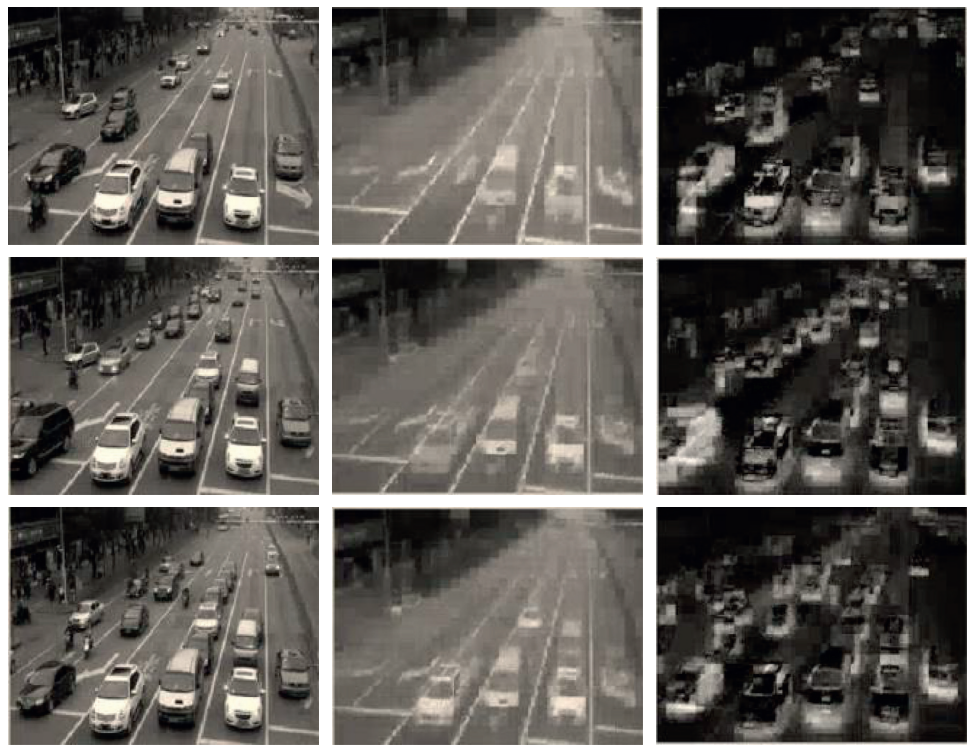

Figure 4: Detection results of the structured sparse representation method.

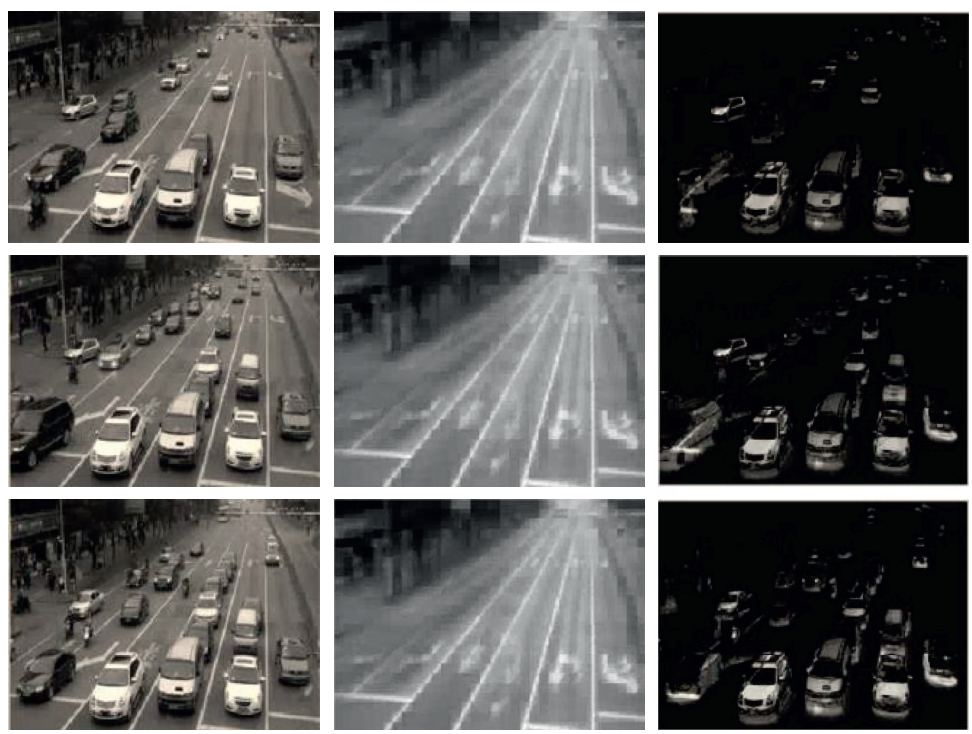

Figure 5: Detection results of our method.

3.2.1. Comparative Analysis of Detection Algorithms in Simple Intersection Scenarios. The intersection scenario in the PV video sequences of *i*-LIDS is relatively simple, and a slight camera shake in the video causes unstable background interference. To save space, Figure 7 shows visual comparison results of foreground detection for only the most challenging one in video clips. The vehicles in the video start to move slowly and queue up temporarily. In Figure 7, the first column is the original video image, the second column is the current background extracted by each method,

Table 2: Comparison of the metrics for test sequences.

Methods	*F*_measure	Precision	Recall
Method1	0.3893	0.3715	0.4089
Method2	0.4186	0.5889	0.3247
Our	0.9171	0.9131	0.9211

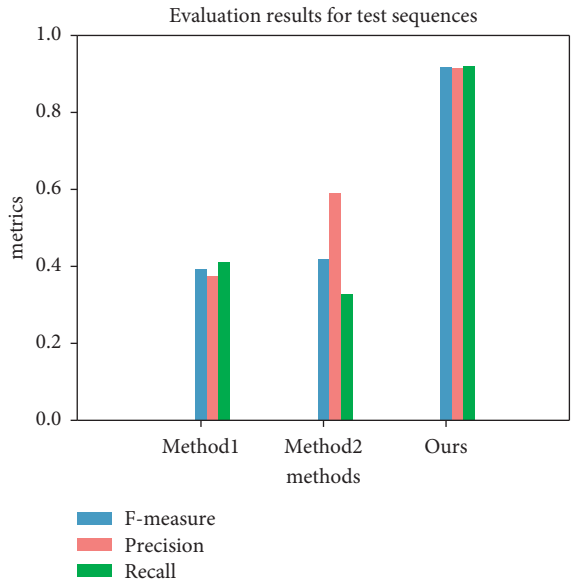

Figure 6: Evaluation results for test sequences.

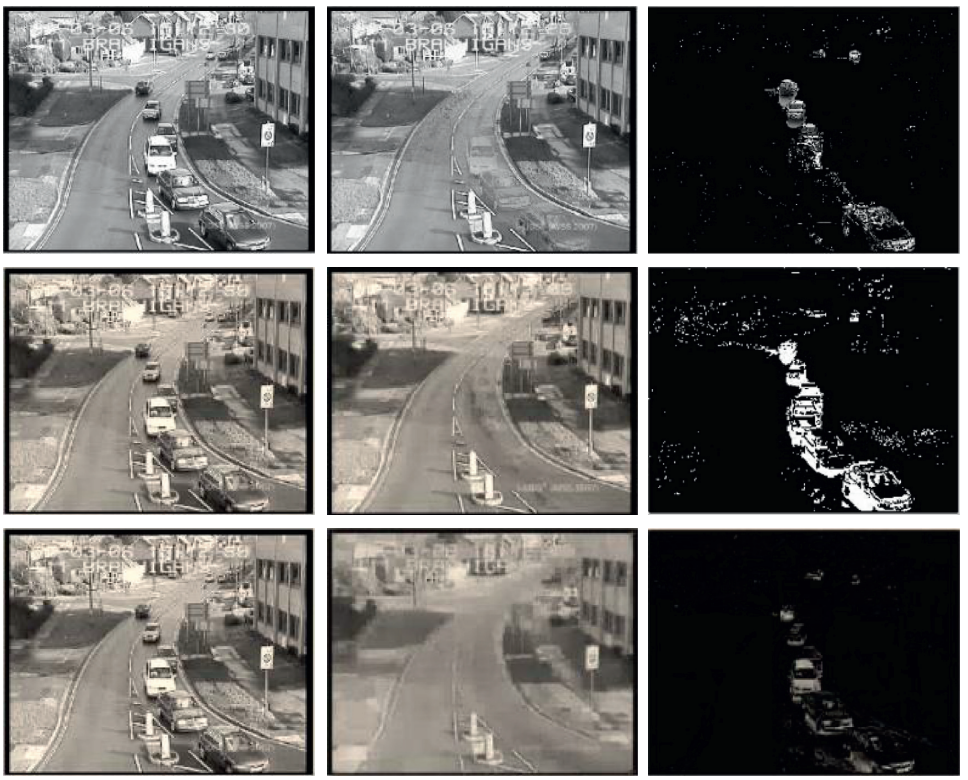

Figure 7: Detection results in a simple intersection scenario.

Figure 8: Detection results in complex intersection scenarios (a small number of queued vehicles).

and the third column is the foreground detection image, and row (a) is the detection results of the GMM model, row (b) is the detection results of the sigma-delta confidence model, and row (c) is the detection results of our method. Using the GMM model, the background is contaminated by slow-moving or temporarily stopped vehicles, and it is failed to detect the intact foreground. To some extent, the sigma-delta confidence model prevents the slow-moving or temporarily stopped vehicles from blending into the background. However, the background recovered using this model is not very accurate, and the extracted background has smear and ghost artifacts. With the foreground unrelated limitations in the background updates, our method achieves promising results. We can prevent slow-moving or temporarily stopped vehicles from blending into backgrounds. The processing speed is significantly improved after converting the image to 192×256 pixels. The sigma-delta confidence model takes about 4.35 frames/s, and the speed advantage is obvious. The GMM model takes about 1.47 frames/s, and our algorithm is 0.65 frames/s (1.54 s/frame).

3.2.2. Comparative Analysis of Different Algorithms in Complex Intersection Scenarios. We compare our method with the GMM model and sigma-delta confidence model (SDCM) in the actual complex intersection scenario. Due to the heavy traffic at this intersection, affected by traffic signals, there are many vehicles queuing in line behind the stop line, and the waiting time is long. Figure 8 shows the comparison results of the three detection algorithms. Figure 8 shows the case in which vehicles have less queue length and shorter queuing time at the intersection. Figure 9 shows the case that vehicles have a long queue length and a long waiting time.

In Figures 8 and 9, the first column is the original video image, at which time vehicles have started to queue; the second column is the background extracted by each algorithm; the third column is the foreground detection. Row (a) is the detection result of the GMM model; row (b) is the detection result of the sigma-delta confidence model; row (c) is the detection result of our algorithm. Figure 8 shows the case where the actual intersection scenario is relatively simple, and the vehicle queue time is not long. Using the GMM model, the background is contaminated by slow-moving and queuing vehicles, resulting in the foreground detection to fail. The background extracted by the sigma-delta confidence model is partially contaminated, and the foreground is not intact. Our method shows great power toward accurate background-foreground separation. Figure 9 shows the difference in detection results of the three

Figure 9: Detection results in complex intersection scenarios (many queued vehicles).

Table 3: Comparison of the metrics for test sequences.

Sequence	Simple intersection scenario (s)			Small queued vehicles (m)			Large queued vehicles (h)		
Methods	F	Pre	Re	F	Pre	Re	F	Pre	Re
GMM	0.4648	0.5149	0.4236	0.3878	0.4023	0.3744	0.2817	0.3011	0.2647
SDCM	0.8054	0.7915	0.8198	0.5850	0.5415	0.6362	0.4468	0.4178	0.4801
Ours	0.9375	0.9426	0.9325	0.9279	0.9305	0.9254	0.9096	0.9117	0.9075

algorithms when the traffic volume becomes heavier and the waiting time becomes longer at the intersection. The background extracted by the GMM model is contaminated seriously, resulting in the foreground being seriously missing. Due to more vehicles and longer waiting times at the intersection, part of the foreground blends into the background with the sigma-delta confidence model, resulting in poor results of the foreground detection. Our method successfully detects the foreground at this complex scenario and recovers accurate background.

Table 3 and Figure 10 show the quantitative evaluation results by precision, recall, and F-measure on the test video clips. For the accuracy of the results, we only perform statistics on a road area where there are many slow-moving or temporarily stopped vehicles. The precision results show that the GMM model performs the foreground detection task with medium and very low accuracy for simple and complex intersection scenarios, respectively. This is due to the slow-moving or temporarily stopped vehicles at the intersection scenario. The sigma-delta confidence model and our method detect foreground regions with high accuracy in the simple sequences, but the sigma-delta confidence model introduces few false positives into the final mask. For the complex sequences, the accuracy of the sigma-delta confidence model is dramatically decreased due to the longer vehicle waiting times; the background extracted by this model presents more severe smearing artifacts. The proposed method achieves promising results for all the video sequences. With the help of foreground unrelated limitations in the background updates, we can prevent the slow-moving or temporarily stopped vehicles from leaking into backgrounds and recover the accurate backgrounds without smearing and ghosting artifacts.

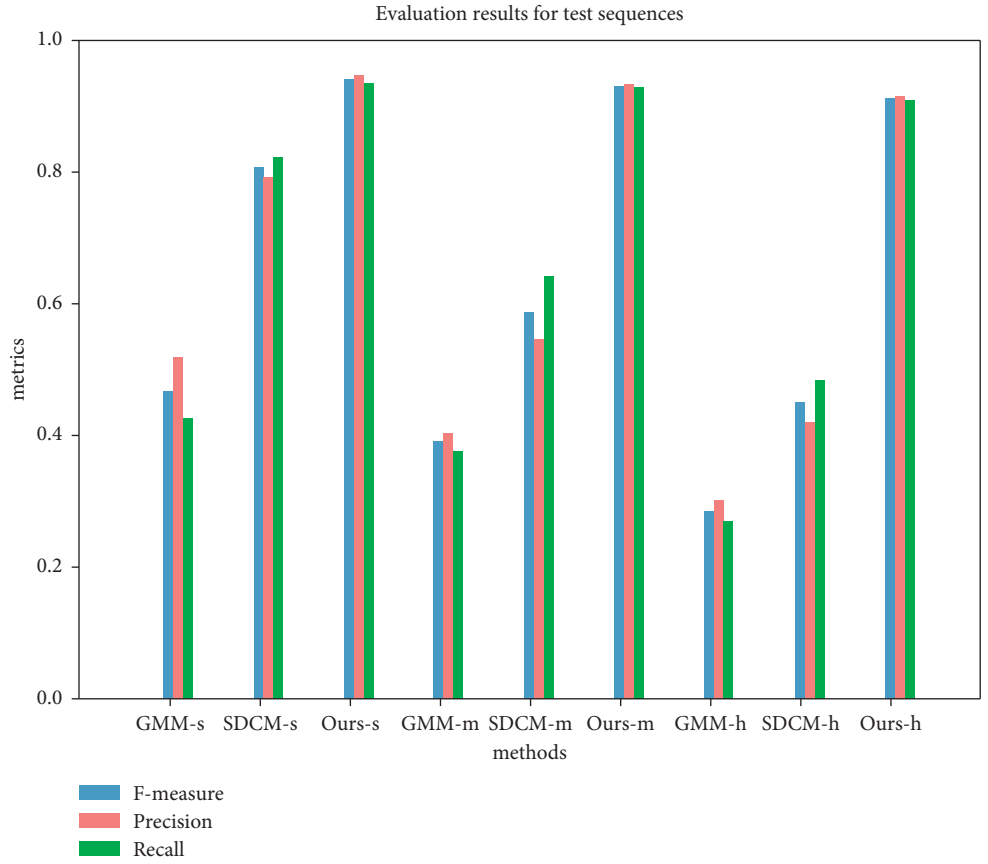

Figure 10: Evaluation results for test sequences.

4. Conclusions

In this study, based on the background subtraction of the sparse representation, a background dictionary update model with unrelated foreground is established. This model can limit the foreground into the background dictionary when the background dictionary is updated, thus avoiding the background pollution and the lack of traffic foregrounds caused by the slow-moving or temporarily stopped vehicles at urban intersections. At the same time, the manifold regular term is introduced to establish a sparse coding model of foreground consistency representation. The model can overcome the problem of foreground discontinuity caused by the large difference in sparse representation coefficients of the same foreground target due to the independence of sparse representation. We conducted experimental verification of the proposed method in real-world urban traffic videos. First, comparing our method with two other background subtraction methods based on sparse representation, the results show that our method not only keeps the background model being unpolluted from slow-moving or temporarily stopped vehicles, but also maintains a continuous and consistent representation of the foreground target. Then, our method is compared with the other two detection methods of traffic foreground at a simple intersection and complex intersection, including the GMM model and sigma-delta confidence model. The results show that our method can maintain good detection results at complex intersections.

With the increasing traffic volume, traffic parameter detection, and traffic state recognition based on vehicle detection and tracking methods at urban intersections are increasingly limited by occlusion, slow-moving, or temporarily stopped vehicles. Our future work is to select appropriate visual features and their combinations to characterize traffic state parameters and identify traffic conditions based on the extracted robust traffic foreground.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 51608054), Hunan Provincial Natural Science Foundation of China (Grant no. 2018JJ3551), and Scientific Research Fund of Hunan Provincial Education Department (Grant no. 18B138).

References

[1] J.-P. Jodoin, G.-A. Bilodeau, and N. Saunier, "Tracking all road users at multimodal urban traffic intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3241–3251, 2016.

[2] C. Li, "Robust vehicle tracking for urban traffic videos at intersections," in *Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 207–213, Colorado Spring, CO, USA, August 2016.

[3] J. Fan, "Improvement of object detection based on faster R-CNN and YOLO," in *Proceedings of the 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 1–4, IEEE, Grand Hyatt Jeju, Republic of Korea, June 2021.

[4] X. Chen, L. Qi, Y. Yang et al., "Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis," *Journal of Advanced Transportation*, vol. 2020, Article ID 7194342, 12 pages, 2020.

[5] A. O. Wahban, "A vehicular queue length measurement system in real-time based on SSD Network," *Transport and Telecommunication*, vol. 22, no. 1, pp. 29–38, 2021.

[6] M. Piccardi, "Background subtraction techniques: a review," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104, 2004.

[7] A. A. Karim, "A proposed background modeling algorithm for moving object detection using statistical measures," *Iraqi Journal of Science*, vol. 58, no. 3A, pp. 1282–1289, 2017.

[8] M. A. Rahman, "An adaptive background modeling based on modified running Gaussian average method," in *Proceedings of the 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 524–527, IEEE, Cox's Bazar, Bangladesh, February 2017.

[9] H. Sajid and S.-C. S. Cheung, "Universal multimode background subtraction," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3249–3260, 2017.

[10] X. Lu, C. Xu, L. Wang, and L. Teng, "Improved background subtraction method for detecting moving objects based on GMM," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 11, pp. 1540–1550, 2018.

[11] Y. Xia, X. Shi, G. Song, Q. Geng, and Y. Liu, "Towards improving quality of video-based vehicle counting method for traffic flow estimation," *Signal Processing*, vol. 120, pp. 672–681, 2016.

[12] V. Kharchenko, "Kernel density estimation for foreground detection in dynamic video processing for unmanned aerial vehicle application," in *Proceedings of the 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, pp. 71–74, IEEE, Kyiv, Ukraine, October 2019.

[13] S. S. Aung and N. War, "Foreground objects segmentation in videos with improved codebook model," in *Proceedings of the 2019 International Conference on Advanced Information Technologies (ICAIT)*, pp. 161–166, IEEE, Jinan, China, October 2019.

[14] M. Van Droogenbroeck and O. Barnich, "ViBe: A disruptive method for background subtraction," *Background Modeling and Foreground Detection for Video Surveillance*, pp. 1–7, CRC Press, Boca Raton, FL, USA, 2014.

[15] A. Manzanera and J. Richefeu, "A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation," in *Proceedings of the Fourth Indian Conference on Computer Vision, Graphics & Image Processing*, Kolkata, India, December 2004.

[16] X. Huang, F. Wu, and P. Huang, "Moving-object detection based on sparse representation and dictionary learning," *Aasri Procedia*, vol. 1, pp. 492–497, 2012.

[17] L. Li, P. Wang, Q. Hu, and S. Cai, "Efficient background modeling based on sparse representation and outlier iterative removal," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 2, pp. 278–289, 2014.

[18] Y. Luo and H. Zhang, "Sparse learning for robust background subtraction of video sequences," in *Proceedings of the International Conference on Intelligent Computing*, pp. 400–411, Fuzhou, China, August 2015.

[19] C. David, V. Gui, and F. Alexa, "Foreground/background segmentation with learned dictionary," in *Proceedings of the International Conference on Circuits, Systems and Signals, CSS 2009*, pp. 197–201, Athens, Greece, December 2009.

[20] H. Xiao, Y. Liu, S. Tan, J. Duan, and M. Zhang, "A noisy videos background subtraction algorithm based on dictionary learning," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 6, 2014.

[21] H. Yang and S. Qu, "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition," *IET Intelligent Transport Systems*, vol. 12, no. 1, pp. 75–85, 2017.

[22] S. Toral, M. Vargas, F. Barrero, and M. G. Ortega, "Improved sigma-delta background estimation for vehicle detection," *Electronics Letters*, vol. 45, no. 1, pp. 32–34, 2008.

[23] A. Manzanera and J. C. Richefeu, "A new motion detection algorithm based on Σ-Δ background estimation," *Pattern Recognition Letters*, vol. 28, no. 3, pp. 320–328, 2007.

[24] Y. Zhang, C. Zhao, and Q. Zhang, "Counting vehicles in urban traffic scenes using foreground time-spatial images," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 61–67, 2016.

[25] Y. Zhang, C. Zhao, J. He, and A. Chen, "Vehicles detection in complex urban traffic scenes using Gaussian mixture model with confidence measurement," *IET Intelligent Transport Systems*, vol. 10, no. 6, pp. 445–452, 2016.

[26] X. Cui, J. Huang, S. Zhang, and D. N. Metaxas, "Background subtraction using low rank and group sparsity constraints," in *Proceedings of the European Conference on Computer Vision*, pp. 612–625, Florence, Italy, October 2012.

[27] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proceedings of the European Conference on Computer Vision*, pp. 751–767, Dublin, Ireland, June 2000.

[28] i-LIDS Dataset for AVSS 2007. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.