

A Multi-View Learning Approach to Foreground Detection for Traffic Surveillance Applications

Kunfeng Wang, Yuqiang Liu, Chao Gou, and Fei-Yue Wang, *Fellow, IEEE*

Abstract—This paper proposes an effective multi-view learning approach to foreground detection for traffic surveillance applications. This approach involves three main steps. First, a reference background image is generated via temporal median filtering, and multiple heterogeneous features (including brightness variation, chromaticity variation, and texture variation, each of which represents a unique view) are extracted from the video sequence. Then, a multi-view learning strategy is devised to online estimate the conditional probability densities for both foreground and background. The probability densities of three features are approximately conditionally independent and are estimated with kernel density estimation. Pixel soft-labeling is conducted by using Bayes rule and the pixel-wise foreground posteriors are computed. Finally, a Markov random field is constructed to incorporate the spatial-temporal context into the foreground/background decision model. The belief propagation algorithm is used to label each pixel of the current frame. Experimental results verify that the proposed approach is effective to detect foreground objects from challenging traffic environments, and outperforms some state-of-the-art methods.

Index Terms—Foreground detection, heterogeneous features, multi-view learning, conditional independence, Markov random field.

I. INTRODUCTION

NOWADAYS, intelligent visual surveillance that extracts various information of urban traffic is attracting more and

more attention in the fields of computer vision and intelligent transportation systems [1], [2]. *Foreground detection* (also referred to as *background subtraction* in some works) is an important early task in these fields. On the basis of foreground detection, many other applications like object tracking, recognition, and anomaly detection, can be implemented [3].

The basic principle of foreground detection is to compare the current frame of a video scene with a background model and detect zones that are significantly different. Although it seems simple, foreground detection in real-world surveillance is often confronted with three challenges [4]–[6]:

Moving cast shadows, caused due to the occlusion of sunlight by foreground objects, often exist in traffic scenes. Shadows can be hard under sunny condition or soft under cloudy condition. Anyway, they can easily be detected as foreground and interfere with the size and shape information of the segmented objects.

Illumination changes are common in traffic scenes. As the sun moves across the sky, the illumination will change slowly. Sometimes it may change rapidly, e.g., when the sun gets into or gets out of a cloud.

Noise is inevitably introduced during the image capture, compression, and transmission process. If the signal-to-noise ratio is too low, it would be difficult to distinguish foreground objects from the background scene.

These challenges are exemplified in Fig. 1. In traffic scenes, numerous foreground objects (including vehicles and pedestrians) appear, move, and finally disappear under certain natural and social rules. Their appearance features (including brightness, chromaticity, and texture) differ significantly from those of the background. The probability distributions of these features have different forms and are time-varying. Besides, foreground objects are usually compact in the image space and move smoothly over time. Hence, spatial-temporal context within the video sequence can be exploited. In light of these, we propose an effective multi-view learning approach to foreground detection for traffic surveillance applications. We extract multiple heterogeneous image features (i.e., brightness variation, chromaticity variation, and texture variation) from the video sequence, and devise a multi-view learning strategy to online estimate the conditional probability densities for both foreground and background. The probability densities of these features are approximately conditionally independent and are estimated through the use of kernel density estimation. Then, spatial-temporal context is incorporated into the decision model under the Markov random field (MRF) framework, and optimal

This work was supported in part by the National Natural Science Foundation of China under Grant 61304200 and the MIIT Project of Internet of Things Development Fund under Grant 1F15E02.

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

K. Wang is with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (phone: 86-10-82544791; fax: 86-10-82544784; e-mail: kunfeng.wang@ia.ac.cn).

Y. Liu was with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and is now with China Academy of Railway Sciences, China (e-mail: yuqiang.liu@ia.ac.cn).

C. Gou is with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and also with Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: gouchao.cas@gmail.com).

F.-Y. Wang is with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and also with the Research Center for Computational Experiments and Parallel Systems, National University of Defense Technology, Changsha 410073, China (e-mail: feiyue@gmail.com).

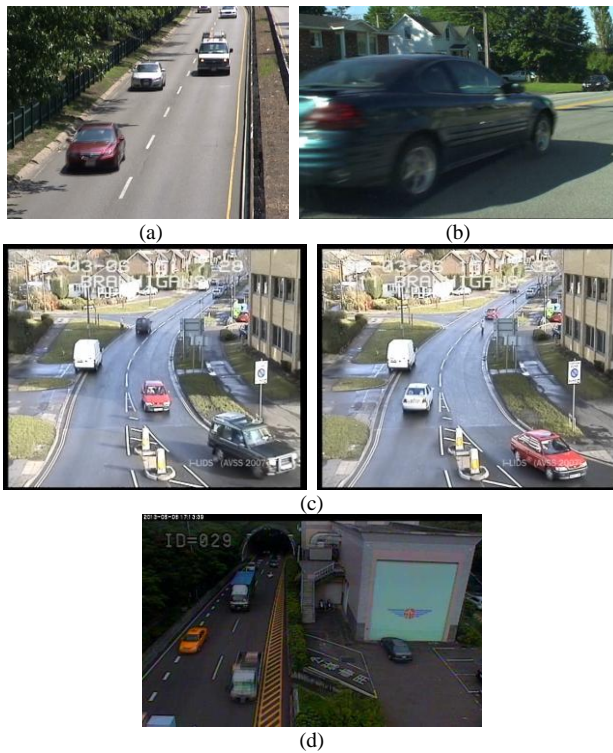


Fig. 1. Examples of foreground detection challenges. (a) Simple challenge with short shadows and slightly swaying trees. (b) Long shadows. (c) Rapid illumination change. (d) Low signal-to-noise ratio due to noise. The video (c) is from AVSS 2007 [47], and other videos are from changedetection.net [5].

foreground segmentation is achieved with belief propagation. With the proposed method, the aforementioned challenges for foreground detection can be alleviated.

The remainder of this paper is arranged as follows. Section II surveys the related works. Section III describes feature extraction, Section IV explains multi-view learning to online estimate the conditional densities for both foreground and background, and Section V introduces the incorporation of spatial-temporal context under the MRF framework. The experimental results are reported in Section VI. Finally, the conclusion is drawn in Section VII.

II. RELATED WORKS

The domain of foreground detection is humongous, and many review papers have been published [5]–[10]. Some researchers [11], [12] classify foreground detection techniques into pixel-level models, region-level models, and frame-level models. But in our opinion, there are hybrid models that do not strictly fit into only one category. In the following subsections, we explore the related works briefly.

A. Sparse Models

Sparse techniques for background subtraction use different variants of principal component analysis (PCA) and matrix decomposition to model the background as a low-rank representation and the foreground as sparse outliers. Oliver *et al.* [13] proposed the eigen-background model, where the PCA was performed on a training sequence. A new frame was projected onto the subspace spanned by the principal components, and the

residues indicated the presence of foreground objects. Tsai *et al.* [14] proposed a similar approach using independent component analysis (ICA). An ICA model was built in the training stage to measure the statistical independency, and the trained de-mixing vector was used to separate the foreground in a new image with respect to the reference background image. Zhao *et al.* [12] proposed a foreground detection approach based on sparse representation and dictionary learning. To build the background model with foreground-present training samples, they designed a robust dictionary learning approach, which simultaneously detected foreground pixels and built a correct background dictionary. Zhou *et al.* [15] addressed the foreground detection challenges with a unified framework of detecting contiguous outliers in the low-rank representation, which integrated foreground detection and background learning into a single optimization process and solved them in a batch manner.

B. Parametric Models

Parametric models are perhaps the most extensively studied models in the foreground detection domain. Gaussian distribution is a common choice. Pfister [16] used a single Gaussian distribution to model the background at each pixel. However, this method cannot handle multimodal background. A substantial improvement was achieved by the Gaussian mixture model (GMM) [17], [18]. First presented in [17], GMM models the observed history of each pixel using a weighted mixture of Gaussians. This model is able to cope with the multimodal nature of many practical situations and lead to good results when repetitive background motions, such as swaying trees and water ripples, are encountered. Since its introduction, GMM has enjoyed tremendous popularity in the surveillance domain [19]–[27]. Lee [19] proposed an effective scheme to improve the convergence rate without compromising model stability in GMM, which was achieved by replacing the global, static retention factor with an adaptive learning rate calculated for every Gaussian at every frame. Martel-Brisson *et al.* [20] built a Gaussian mixture shadow model to learn and remove moving cast shadows. This model was integrated with a GMM for background modeling and foreground detection. Jodoin *et al.* [21] proposed a spatial variation to the traditional temporal modeling framework. This variation allows statistical motion detection with models trained on one background frame. Haque *et al.* [25] proposed perception-inspired background subtraction (PBS), which avoids overreliance on statistical observations by making key modeling decisions based on the characteristics of human visual perception. Haines *et al.* [26] proposed a background subtraction method based on Dirichlet process Gaussian mixture models (DP-GMM), which were used to estimate per-pixel background distributions. This method was said to avoid over-/under-fitting by allowing per-pixel mode counts to be automatically inferred.

Other parametric models have also been used. Cheng *et al.* [28] proposed to use temporal differencing pixels of the Laplacian distribution model, in order to check each block for the presence of either moving object or background. Zhang *et al.* [29] employed the normalized ratio edge difference (NRED)

between the current frame and the background image for moving cast shadow detection. The distribution of NRED in shaded background area was approximated to be a chi-square distribution.

C. Nonparametric and Data-Driven Models

Some researchers employ nonparametric models due to their flexibility in probability density estimation. In [30], the nonparametric kernel density estimation method was proposed to construct a statistical representation of the scene background and detect moving objects in the scene. Sheikh *et al.* [31] proposed a nonparametric density estimation method over a joint domain-range representation of image pixels to model multimodal spatial uncertainties and complex dependencies between the domain (location) and range (color). They further modeled the foreground with the use of temporal persistence and built a MAP-MRF decision framework to augment the detection of objects. This method was susceptible to moving cast shadows. Rivera *et al.* [32] proposed a statistical edge-segment-based method for background modeling in non-ideal circumstances. This method learned the structure of the scene using the edges' behaviors, which were approximated with kernel-density distributions.

As a substitute to probabilistic models, data-driven models that utilize numerical tools like histogram to characterize the samples have been extensively studied for foreground detection. Horprasert *et al.* [33] proposed a computational color model that separates the brightness from the chromaticity component. This model was effective to distinguish shading background from the ordinary background or moving foreground objects. ViBe [34] stores, for each pixel, a set of values taken in the past at the same location or in the neighborhood, and then compares this set to the current pixel value to determine whether that pixel belongs to the background. Heikkilä *et al.* [35] proposed a texture-based method for modeling the background and detecting moving objects. Each pixel was modeled as a group of adaptive local binary pattern histograms that were calculated over a circular region around the pixel. Liao *et al.* [36] extended the work of [35] by proposing a scale invariant local ternary pattern operator and a pattern kernel density estimation technique to effectively model the probability distribution of local patterns in the pixel process. Li *et al.* [37] proposed a Bayesian framework that incorporated spectral, spatial, and temporal features to characterize the background appearance. Under this framework, the background was represented by the most significant and frequent features, i.e., the *principal features*, at each pixel. Lam *et al.* [38] proposed a texture-based method for extracting vehicles from the stationary background that was free from the effect of moving cast shadows. The segmentation method utilized the differences in textural property between the road, vehicle cast shadow, and the vehicle itself. The luminance and chrominance properties were further combined to construct the foreground mask. The selection of thresholds was done with a data-driven iterative algorithm.

D. Machine Learning Models

Some researchers employ machine learning models, such as support vector machine (SVM), neural network, and fuzzy logic, to discriminate between foreground and background. Han *et al.* [39] proposed a multiple feature integration algorithm for background modeling and subtraction, where the background was modeled via kernel density approximation and background and foreground were classified by a supervised SVM. Maddalena *et al.* [40] proposed a self-organizing approach to background subtraction, in which the background model was organized as an artificial neural network of a 2-D flat grid structure, allowing preservation of topological neighborhood relations among the background neurons. Chacon-Murguia *et al.* [41] proposed an adaptive neural-fuzzy method to improve the self-organizing map (SOM) model [40]. Especially, this method included a fuzzy inference module to automatically adjust the threshold parameters involved in the SOM model, making the system independent of the scenario.

Markov random fields (MRF) are widely used to formulate spatial dependencies within each segmentation field and temporal dependencies of consecutive segmentation fields in the video sequence. MRF and other models are in general unified to better discriminate foreground from the background [15], [26], [31], [42]–[44]. The idea of using MRF to impose spatial coherence constraint was included in [15], [26], [31], and [42]. Huang *et al.* [43] proposed a region-level motion-based background subtraction method using MRF. This method consisted of motion-based region segmentation and MRF-based region classification. Spatial and temporal coherence was maintained as prior energy in the MRF model. Wang *et al.* [44] proposed a dynamic conditional random field (DCRF) model for foreground object and moving shadow segmentation in indoor scenes. Both intensity and gradient features were integrated, and models of background and shadow were updated adaptively. Moreover, spatial-temporal context was incorporated into the DCRF model and the segmentation field was estimated by approximate inference.

E. Model Evaluation and Our Contributions

By analyzing the state-of-the-art, we find that a good approach to foreground detection should have three characteristics. First, *it should integrate multiple heterogeneous features*, especially those complementary and uncorrelated ones. Many methods use only pixel intensities (grayscale or color) as features, since they are directly available from images and reasonably discriminative. However, pixel intensities are sensitive to illumination changes and shadows. In fact, some illumination invariant features such as texture can be used to alleviate the disadvantages of pixel intensities. Second, *a good foreground detection method should build, from the observed history, not only the background model, but also the foreground model*. If only the background model was built and foreground pixels were identified purely as outliers, as done in many existing works, then background colored object-parts cannot be identified. Third, *a good foreground detection method should exploit spatial-temporal context within the video sequence*,

TABLE I
INTUITIVE COMPARISON BETWEEN MAJOR REFERENCES AND THE PROPOSED METHOD

Methods	Features	Models	Context
Oliver <i>et al.</i> [13]	Intensity	Background	No
Tsai <i>et al.</i> [14]	Intensity	Background	No
Zhao <i>et al.</i> [12]	Intensity	Background	No
Zhou <i>et al.</i> [15]	Intensity	Background	Spatial
Pfinder [16]	Intensity	Background	No
Stauffer <i>et al.</i> [17], [18]	Intensity	Background	No
Martel-Brisson <i>et al.</i> [20]	Intensity	Background, shadow	No
Jodoin <i>et al.</i> [21]	Intensity	Background	No
Haque <i>et al.</i> [25]	Intensity	Background	No
Haines <i>et al.</i> [26]	Intensity	Background; foreground intensity is assumed to be uniform distribution	Spatial
Zhang <i>et al.</i> [29]	Intensity, ratio edge	Background, shadow	No
Elgammal <i>et al.</i> [30]	Intensity	Background	No
Sheikh <i>et al.</i> [31]	Intensity	Background, foreground	Spatial
Rivera <i>et al.</i> [32]	Edge-segment	Background	No
Horprasert <i>et al.</i> [33]	Brightness, chromaticity	Background	No
ViBe [34]	Intensity	Background	No
Heikkilä <i>et al.</i> [35]	Texture	Background	No
Liao <i>et al.</i> [36]	Texture	Background	No
Li <i>et al.</i> [37]	Color, gradient, color co-occurrence	Background	No
Lam <i>et al.</i> [38]	Luminance, chrominance, texture	Background	No
Han <i>et al.</i> [39]	Color, gradient, Harr-like features	Background	No
Maddalena <i>et al.</i> [40]	Color (HSV)	Background	No
Chacon-Murguía <i>et al.</i> [41]	Color (HSV)	Background	No
Huang <i>et al.</i> [43]	Color, optical flow	Background	Spatial-temporal
Wang <i>et al.</i> [44]	Intensity, gradient	Background, shadow; foreground intensity is assumed to be uniform distribution	Spatial-temporal
Proposed method	Brightness, chromaticity, texture	Background, foreground	Spatial-temporal

which helps to improve the accuracy of foreground/background decision and reduce the reliance on postprocessing techniques. For intuitive comparison, we summarize the major references in Table I, in terms of how the features, models, and context are formulated.

In summary, four contributions are made in this paper:

1) Multiple heterogeneous features regarding brightness, chromaticity, and texture are extracted from the video sequence. These features are robust to shadows and illumination changes, and are approximately conditionally independent given the class label. An iterative search and multiscale fusion strategy is proposed to extract features reliably.

2) A multi-view learning method is devised to online estimate the conditional probability densities for both foreground and background, and pixel soft-labeling is conducted to estimate the pixel-wise foreground posterior.

3) Spatial-temporal contextual constraints are incorporated into the foreground/background decision model under the MRF framework, and optimal foreground segmentation is achieved via belief propagation.

4) A novel, accurate, and robust algorithm is proposed for detecting foreground objects from complex, challenging traffic environments.

III. FEATURE EXTRACTION

In this section, we describe the feature extraction module, which is an important premise for foreground detection. The

features that are insensitive to illumination changes and shadows should be used. Multiple heterogeneous features are unified to better discriminate foreground from the background.

A. Generation of Reference Background Image

First of all, we need to generate and maintain an up-to-date reference background image, in order to represent the inherent structure of the monitored scene. We use temporal median filtering (TMF), which takes the median value at each pixel over a predefined time window as the reference background of that pixel. In our implementation, the reference background image is updated once every 50 frames, by conducting TMF over the recent 500 frames. Note that if traffic volume becomes available from some top-down feedback, the time window can be adjusted accordingly. TMF has two advantages: 1) it can automatically update the reference background and adapt to gradual illumination changes; 2) it can capture the scene's inherent structure even when some background objects, such as trees, are not absolutely static. Fig. 2 shows the generated reference background images for the four scenes in Fig. 1.

B. Extraction of Heterogeneous Features

After generation of the reference background image, we proceed to extract three heterogeneous features from the images, i.e., brightness variation, chromaticity variation, and texture variation. These features denote the brightness, chromaticity, and texture differences between the current image and the reference background image.



Fig. 2. Examples of reference background images generated by temporal median filtering.

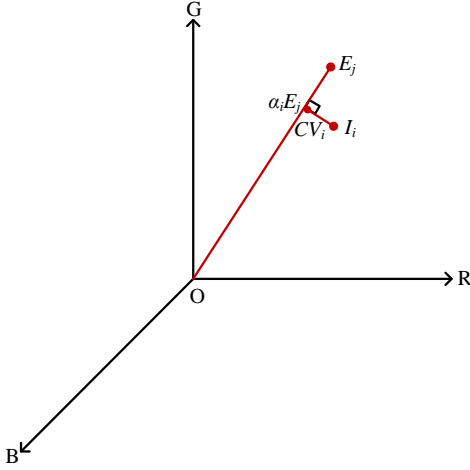


Fig. 3. Extended computational color model in the RGB color space.

Extraction of Brightness Variation and Chromaticity Variation

The extraction of brightness variation and chromaticity variation is inspired by [33], in which static background was assumed. Considering that dynamic background may exist in traffic scenes, we make some extensions to the computational color model proposed in [33]. As shown in Fig. 3, I_i represents the observed color of a given i th pixel in the current image I , and E_j represents the expected color of the j th pixel in the reference background image. We assume that in challenging environments, nonstationary background points (such as the j th point) will move to nearby positions (such as the i th point) in the image space and keep its appearance features. This assumption is reasonable in some degree. The correspondence between the i th pixel in the current image and the j th pixel in the reference background image will be described later in this section.

For a given pixel $i \in I$, we want to compute the brightness and chromaticity variations of I_i from E_j . We first compute α_i , which is equal to the ratio between the pixel's strength of brightness and the expected value. Let $I_i = [I_R(i), I_G(i), I_B(i)]$ and $E_j = [E_R(j), E_G(j), E_B(j)]$ represent the RGB color values. Referring to [33], α_i can be computed as

$$\alpha_i = \frac{I_R(i)E_R(j) + I_G(i)E_G(j) + I_B(i)E_B(j)}{[E_R(j)]^2 + [E_G(j)]^2 + [E_B(j)]^2}. \quad (1)$$

Brightness variation BV_i is defined as the signed distance of $\alpha_i E_j$ from E_j , that is,

$$BV_i = (\alpha_i - 1) \|OE_j\|, \quad (2)$$



Fig. 4. Result of texture variation computation without proper processing. (a) is the current image. The first image in Fig. 2 is the reference background image. The computed texture variation is multiplied by 50 and shown in (b). It is clear that swaying trees cause large texture variations at many background pixels.

where $\|OE_j\|$ denotes the straight-line distance between the origin and the point E_j . According to (2), BV_i is 0 if the brightness of a given pixel in the current image is the same as in the reference background image. BV_i is negative if it is darker and positive if it is brighter than the expected brightness.

As defined by [33], chromaticity variation CV_i is the orthogonal distance between the observed color I_i and the expected chromaticity line OE_j , that is,

$$CV_i = \sqrt{(I_R(i) - \alpha_i E_R(j))^2 + (I_G(i) - \alpha_i E_G(j))^2 + (I_B(i) - \alpha_i E_B(j))^2} \quad (3)$$

It is clear that BV_i and CV_i are both distances in the RGB color space and have the same measure unit. Hence the values of these two features can be quantified directly to integers. This is significant for efficient kernel density estimation. In this work, we choose the computational color model in the RGB space not only because the brightness variation and chromaticity variation have strict geometrical definitions, but also because they are conditional independent (see Section III.C).

Extraction of Texture Variation

In this work, we use ratio edge to characterize the texture variation. For a given i th pixel in the current image, suppose its neighboring region $N(i)$ is an 8-pixel neighborhood (3×3 grid minus the center pixel). We can compute the i th pixel's texture variation TV_i using the current image and the reference background image, that is,

$$TV_i = \sqrt{\sum_{m \in N(i)} \left\{ \left(\frac{I_R(m)}{I_R(i)} - \frac{E_R(n)}{E_R(j)} \right)^2 + \left(\frac{I_G(m)}{I_G(i)} - \frac{E_G(n)}{E_G(j)} \right)^2 + \left(\frac{I_B(m)}{I_B(i)} - \frac{E_B(n)}{E_B(j)} \right)^2 \right\}}$$

where $[I_R(i), I_G(i), I_B(i)]$ and $[E_R(j), E_G(j), E_B(j)]$ have the

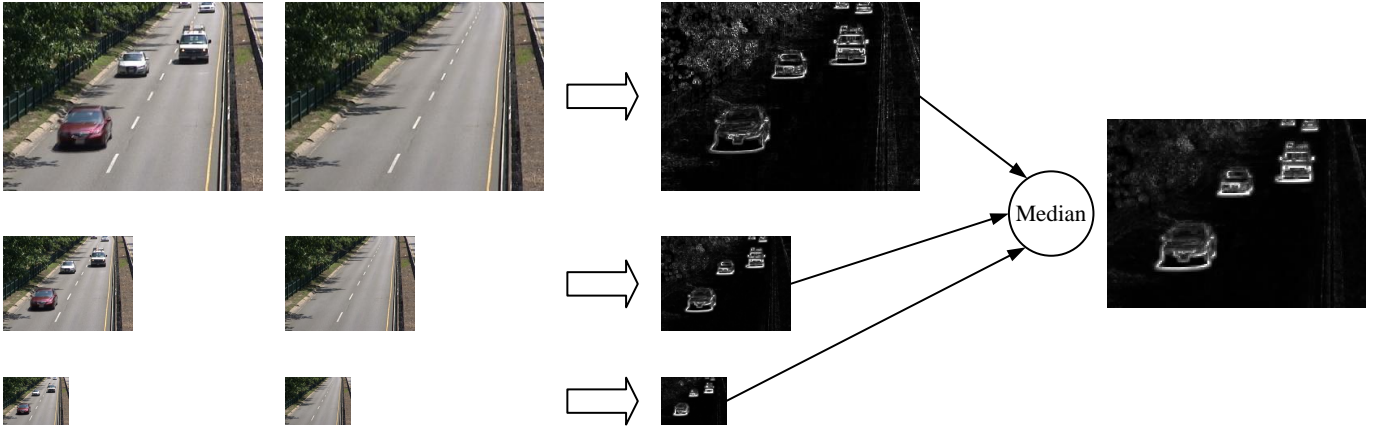


Fig. 6. Example of texture variation computation with the proposed iterative search and multiscale fusion strategy.

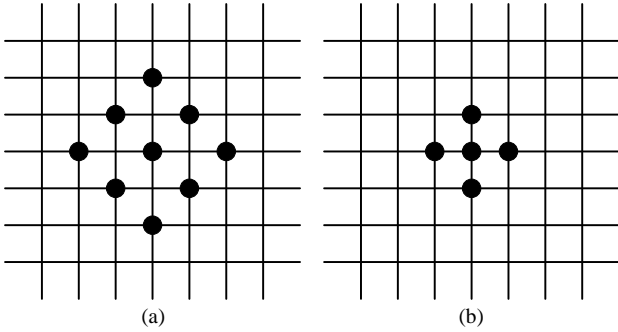


Fig. 5. Pyramid search templates in which black dots denote the search positions in an iteration. (a) Large pyramid template. (b) Small pyramid template.

same meanings as in (1).

The texture variation feature is robust to illumination changes and moving cast shadows. However, it is sensitive to dynamic background. As shown in Fig. 4, without proper processing, swaying trees can cause large texture variation at the background pixels. In our opinion, if the i th pixel in the current image and the j th pixel in the reference background image are precisely matched, this trouble can be mitigated in some degree. Hence, we search for the pixel point j in the reference background image that minimize TV_i around the pixel point i .

Exhaustive search is extremely time consuming and should be avoided for real-time applications. Here we adopt an iterative search strategy. We define two pyramid search templates—a large one and a small one, as shown in Fig. 5. We first conduct coarse search using the large pyramid template. Before the iteration, the pixel point i is initialized as the center point of the search template. At most nine positions need be explored in each iteration. The optimal position (which minimizes TV_i among the nine positions) is set as the center point for the next iteration. This iterative process repeats until the optimal position happens to be the center point of the search template. We then conduct fine search using the small pyramid template. Five positions are explored and the optimal position is finally determined based on the TV_i minimization standard. This final optimal position is considered as the matched pixel j in the reference background image for the pixel i in the current image.



Fig. 7. Brightness variation and chromaticity variation extracted from the same images as in Fig. 6. The brightness variation is added by 128 and shown in (a), and the chromaticity variation is multiplied by 2 and shown in (b).

Based on this association, brightness variation and chromaticity variation are computed with formulas (1)–(3).

However, it is possible in practice that iterative search gets stuck in local minima. When background pixels have large ranges of motion, this becomes more probable. Although increasing the size of search template is able to alleviate this problem, it suffers from a high computational cost that we would like to avoid. Considering that complementary information may exist in multiscale images, we propose a multiscale fusion strategy. The original images (both current image and reference background image) are scaled to 1/2 and 1/4 times. Not only on the original images, we also conduct iterative search on the scaled-down images and extract features therein. Then, the image features of different scales are fused on the original space simply using a median operator. An example of texture variation computation with the proposed iterative search and multiscale fusion strategy is illustrated in Fig. 6. In contrast to Fig. 4, it can be seen that the disturbance of dynamic background has been mitigated. The brightness variation and chromaticity variation extracted from the same images are shown in Fig. 7.

C. Conditional Independence of Features

We now explore the conditional independence of features. According to the definitions, brightness variation and chromaticity variation are orthogonal in the computational color model. Given the class label C that takes on “FG” (foreground) or “BG” (background), the distribution of chromaticity variation is conditionally independent of brightness variation, and vice versa. In addition, texture variation reflects the spatial

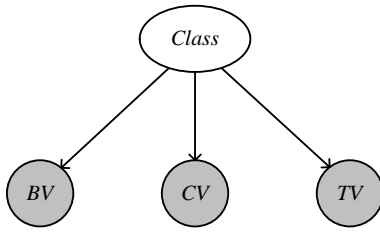


Fig. 8. The naive Bayes model.

layout of neighboring pixels and does not rely on the features of a specific pixel. Hence the distribution of texture variation is conditionally independent of brightness variation and chromaticity variation given the class label C . These independence relations can be represented using a naive Bayes model of Fig. 8. Based on this model, the conditional density can be factorized as

$$p(BV, CV, TV | C) = p(BV | C)p(CV | C)p(TV | C).$$

We record the feature values of the video “highway”, whose ground truth labels are publicly available at changedetection.net. Fig. 9 illustrates the correlation between every pair of features. As can be seen, the correlation coefficients are close to 0. This confirms that given the class label, the three features are approximately conditional independent.

IV. DENSITY ESTIMATION VIA MULTI-VIEW LEARNING

In this section, we describe the conditional density estimation of the aforementioned features. In most existing works, only the background model was built and foreground pixels were identified purely as outliers, or the foreground model was simply assumed to be a uniform distribution. In our opinion, an elaborate foreground model is essential to distinguishing foreground from the background. Motivated by the conditional independence of features, we propose a multi-view learning strategy to learn the foreground model from online data.

A. Conditional Density Estimation

According to the features’ definitions, background pixels (including shadows) should have low brightness variation, low chromaticity variation, and low texture variation, whilst foreground pixels should have widespread brightness variation, high chromaticity variation, and high texture variation. This claim is reasonable and widely recognized in the visual surveillance community [29], [33], [37], [38], and can also be verified by an example of Fig. 10, in which the feature values corresponding to background and foreground are plotted as histogram curves that depict the frequency data.

The conditional independence of features is a key to conditional density estimation. Each feature can be regarded as a unique view, and a multi-view learning strategy is elaborately devised. Due to conditional independence of features, we have

$$p(BV | FG) = p(BV | FG, CV > \tau_{CV} \text{ or } TV > \tau_{TV}), \quad (4)$$

where τ_{CV} and τ_{TV} are thresholds of CV and TV , respectively. In other words, given the class label $C = FG$, the distribution of brightness variation does not depend on the specific values of chromaticity variation and texture variation.

	BV	CV	TV
BV	1.00	0.21	-0.33
CV	0.21	1.00	-0.26
TV	-0.33	-0.26	1.00

(a)

	BV	CV	TV
BV	1.00	0.001	-0.33
CV	0.001	1.00	0.28
TV	-0.33	0.28	1.00

(b)

Fig. 9. Correlations between features. (a) The correlation coefficients between feature values that correspond to the foreground class. (b) The correlation coefficients between feature values that correspond to the background class.

Furthermore, because background pixels (including shadows) have low chromaticity variation and low texture variation, if τ_{CV} and τ_{TV} are large enough, the pixels that satisfy $CV > \tau_{CV}$ or $TV > \tau_{TV}$ can be confidently believed to be foreground pixels. This rule can be written as

$$\text{If } CV > \tau_{CV} \text{ or } TV > \tau_{TV}, \text{ Then } C = FG. \quad (5)$$

Combining (4) and (5), we immediately get

$$p(BV | FG) = p(BV | CV > \tau_{CV} \text{ or } TV > \tau_{TV}). \quad (6)$$

The right hand side of (6) indicates that we can use those pixels that satisfy $CV > \tau_{CV}$ or $TV > \tau_{TV}$ to estimate $p(BV | FG)$.

Similarly, we can estimate the conditional densities of CV and TV given the class label $C = FG$,

$$p(CV | FG) = p(CV | BV > \tau_{BV} \text{ or } TV > \tau_{TV}),$$

$$p(TV | FG) = p(TV | BV > \tau_{BV} \text{ or } CV > \tau_{CV}),$$

where τ_{BV} is a threshold of BV .

Let us underline that in all of our implementations, we fix $\tau_{BV} = \max(40, 40 + \text{median}_{k \in I} BV_k)$, $\tau_{CV} = 20$, and $\tau_{TV} = 3.6$.

Here $\text{median}_{k \in I} BV_k$ denotes the median value of BV in the whole image. It is used to compensate for global brightness variation in all pixels, which may arise from global illumination changes or camera automatic adjustments. The three parameters are critical to our choosing confident foreground pixels and estimating the conditional probability densities for both foreground and background. See the next paragraphs.

From each input frame, we apply the rule (7) to pick out the “confident” foreground pixels, which are then dilated using a square structuring element whose width is 5 pixels to propagate the confidence to spatially neighboring pixels and generate a plausible foreground mask.

$$\text{If } BV > \tau_{BV} \text{ or } CV > \tau_{CV} \text{ or } TV > \tau_{TV}, \text{ Then } C = FG. \quad (7)$$

All the pixels outside the foreground mask constitute a plausible background mask, and are used to estimate the conditional density for background. The computation process is shown in Fig. 11, where it can be seen that most of the non-background pixels are excluded from the plausible background mask.

As the system runs in practice, large amounts of data emerge and are accumulated for density estimation. Fig. 10 illustrates the histograms of the accumulated feature values from the video

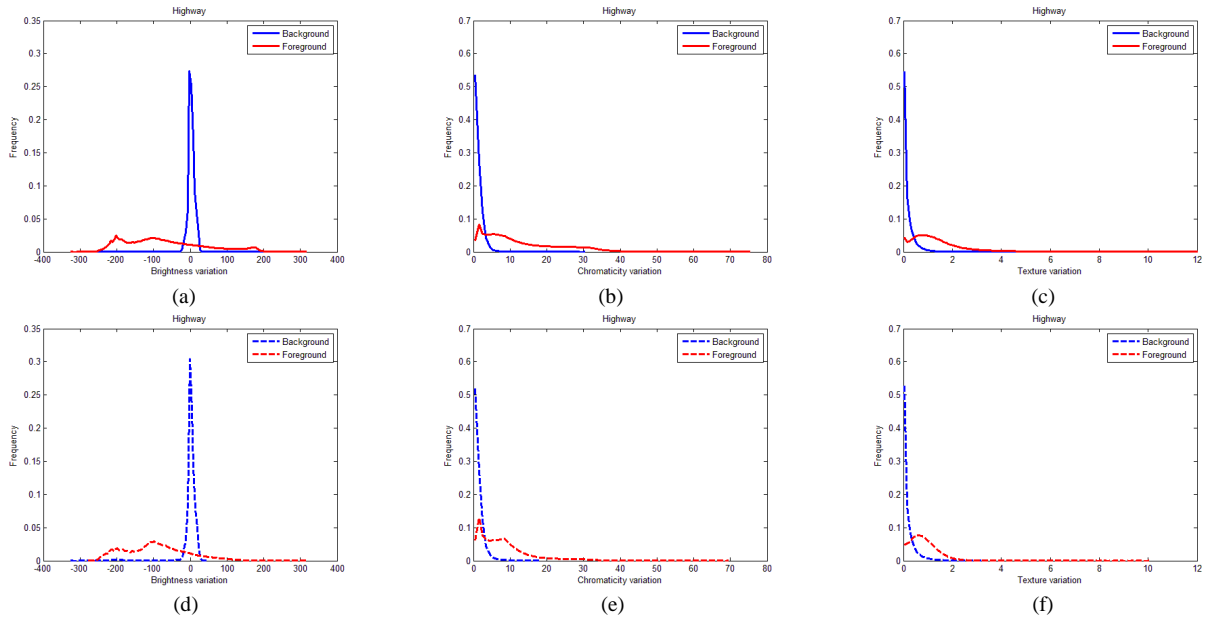


Fig. 10. Frequency histograms of feature values from the video “highway”. (a)(d) Brightness variation. (b)(e) Chromaticity variation. (c)(f) Texture variation. The top row shows feature values that are labeled by ground truth, whilst the bottom row shows feature values that are labeled by multi-view learning. In the pictures, blue curves correspond to the background class, and red curves correspond to the foreground class.

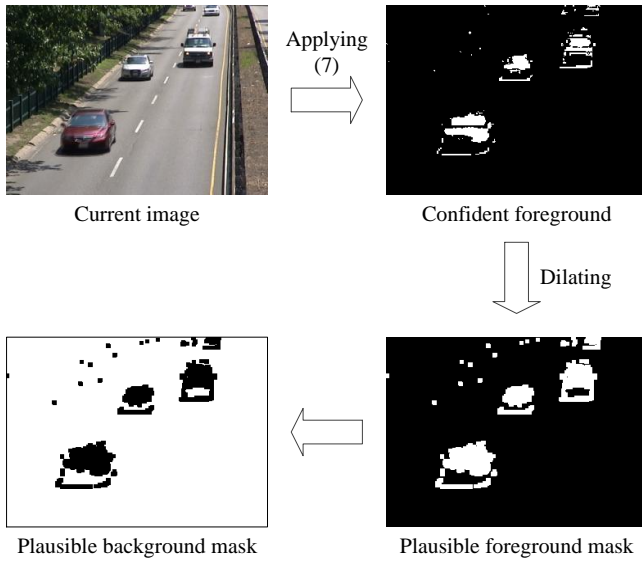


Fig. 11. Computation process of the plausible background mask.

“highway”. The top row of Fig. 10 shows feature values that are labeled by ground truth, whereas the bottom row shows feature values that are labeled by multi-view learning. Comparing the histogram pairs in three columns, it can be seen that the proposed multi-view learning idea is very effective and reliable to generate the frequency data.

From Fig. 10, we also find that the distributions of feature values are complex and cannot be approximated using parametric models. Hence we would like to avoid making assumptions about the specific functional forms of probability distributions. Instead, we use nonparametric kernel density estimation to model their distributions. Brightness variation and chromaticity variation are both distances in RGB color space, so that their values can be quantified directly to integers. Texture variation can also be quantified, using 0.1 as the interval. As a

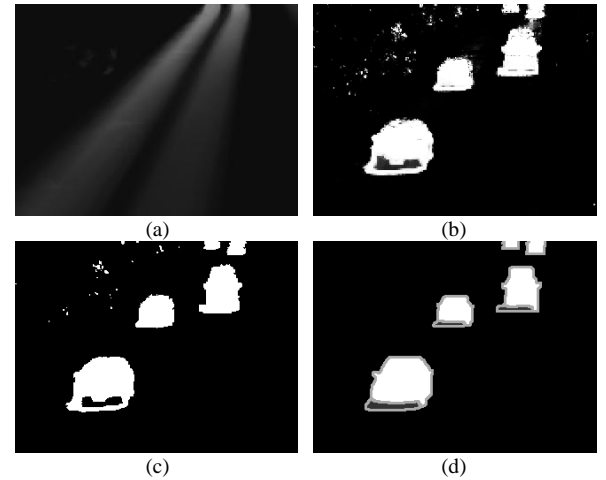


Fig. 12. An example of pixel labeling. (a) Prior probabilities of pixels belonging to foreground. (b) Posterior probabilities of pixels belonging to foreground. The higher the intensity, the more probably the pixel belongs to foreground. (c) Inference result by using belief propagation. (d) Ground truth.

result, the kernel density estimators require little memory, and the computational cost won’t grow with the size of the data set. In order to obtain smooth density models, we use the Gaussian kernel function. The kernel standard deviations σ for three features are fixed to $\sigma_{BV} = 2.0$, $\sigma_{CV} = 2.0$, and $\sigma_{TV} = 0.2$. Note that because the data set is rather large, the standard deviations can be small.

B. Pixel Soft-Labeling with Bayes Rule

After conditional density estimation, we perform pixel soft-labeling with Bayes rule. In other words, we compute the posteriors of background and foreground conditioned on the extracted features. This computation is based on the background likelihood, the foreground likelihood, and the priors at each

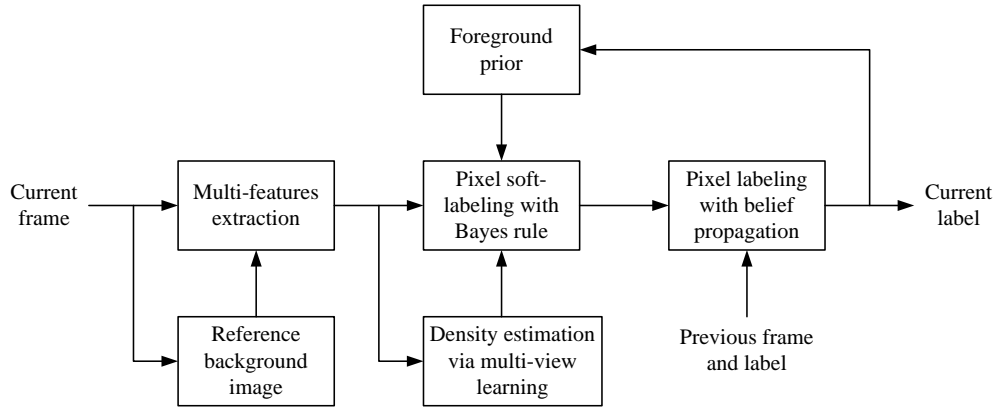


Fig. 13. Block diagram of the proposed method.

pixel. Given an extracted feature $x = [b_v, c_v, t_v]$ at pixel i in the current image, the posteriors (say, soft-labels) of foreground and background are computed via

$$P_i(\text{FG} | x) = \frac{p(x | \text{FG}) \times P_i(\text{FG})}{\sum_{C=\text{FG}, \text{BG}} p(x | C) \times P_i(C)}, \quad (8)$$

$$P_i(\text{BG} | x) = 1 - P_i(\text{FG} | x),$$

where $p(x | C)$ denotes the likelihood and can be computed based on the naive Bayes model in Fig. 8,

$$p(x | C) = p(b_v | C) p(c_v | C) p(t_v | C).$$

$P_i(C)$ denotes the prior probability of foreground and background at pixel i . A prior means the preference over each label. In some existing works, the prior term was ignored and only the likelihoods were used (refer to [30], [31], and [39] for examples). However, we believe that the use of a sophisticated, data-driven prior model is essential to improving the accuracy of foreground detection.

The priors should be spatially distinct. Compared with trees, buildings, and the sky in the scene, the road region should have a higher foreground prior. The priors should also be time-varying. In recent times, if a pixel is labeled as foreground more frequently than before, its foreground prior should increase; if the pixel is labeled as foreground less frequently, its foreground prior should decrease. In light of these, we maintain a dynamic prior model based on the labels of the previous frames,

$$P_{i,t+1}(\text{FG}) = (1 - \rho) P_{i,t}(\text{FG}) + \rho L_{i,t},$$

where $P_{i,t+1}(\text{FG})$ and $P_{i,t}(\text{FG})$ are the i th pixel's foreground priors at time $t+1$ and t , respectively. $L_{i,t}$ denotes the i th pixel's label at time t , which equals 1 if the pixel i is labeled as foreground and equals 0 if labeled as background. ρ denotes the learning rate, fixed to 0.001 empirically.

In the initialization stage, the foreground prior $P_{i,t}(\text{FG})$ is set to a suitable value, such as 0.2. In the updating stage, $P_{i,t}(\text{FG})$ should not be too low, otherwise occasionally emerging objects will be missed. Formally, we demand

$$P_{i,t}(\text{FG}) = \max \{0.01, P_{i,t}(\text{FG})\},$$

which prevents the foreground prior from becoming too low.

An example of pixel soft-labeling is illustrated in Fig. 12(a) and 12(b), including prior and posterior of pixels belonging to foreground. From Fig. 12(a), it can be seen that the road region has much higher foreground priors than the tree region. From Fig. 12(b), it is clear that true foreground objects have high posteriors of belonging to foreground, whereas true background regions have low posteriors of belonging to foreground.

V. PIXEL LABELING WITH BELIEF PROPAGATION

The pixel soft-labeling discussed in Section IV is conducted for each pixel separately, leaving out the contextual constraints from the spatial and temporal neighborhoods of each pixel. This processing is susceptible to local ambiguity and uncertainty. To resolve this issue a grid-structured Markov random field (MRF) [45] is constructed, with a node for each pixel, connected using a four-way spatial neighborhood.

We are facing a binary labeling problem, where each pixel either belongs to the foreground or to the background. Let I be the set of pixels in the current frame and L be the set of labels. The labels correspond to quantities we want to estimate at each pixel: 1 for foreground and 0 for background. A labeling f assigns a label $f_i \in L$ to each pixel $i \in I$. Under the MRF framework, the labels should vary slowly almost everywhere but change dramatically at some places such as pixels along object boundaries. The quality of a labeling is determined by an energy function,

$$E(f) = \sum_{i \in I} D_i(f_i) + \sum_{(i,u) \in N} W(f_i, f_u),$$

where N is the set of undirected edges in the four-connected grid graph. $D_i(f_i)$ is the data term, which measures the cost of assigning label f_i to pixel i . $W(f_i, f_u)$ is the smoothness term, which measures the cost of assigning labels f_i and f_u to two spatially neighboring pixels. A labeling that minimizes this energy corresponds to the maximum a posteriori (MAP) estimation of the MRF.

The data term $D_i(f_i)$ is composed of two parts. The first part is given by the posterior probabilities of a pixel belonging to the foreground and to the background:

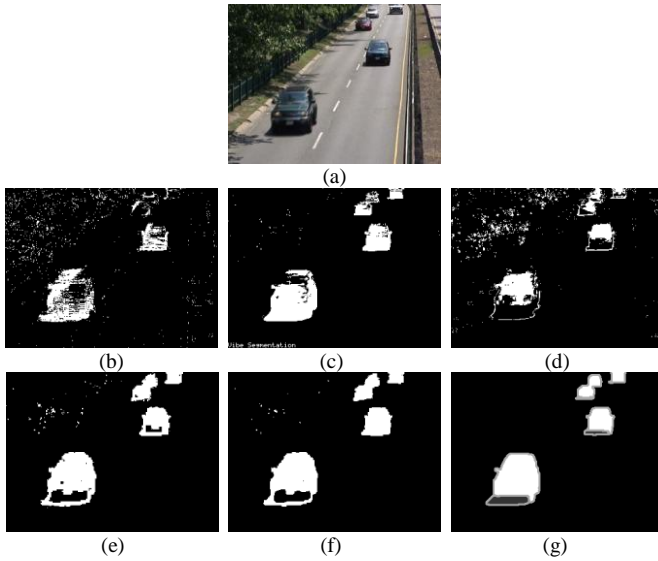


Fig. 14. Comparative foreground/background segmentation results of five methods for one frame taken from the “highway” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

$$D_i^1(f_i) = \begin{cases} -\log P_i(\text{FG} | x), & \text{if } f_i = 1, \\ -\log P_i(\text{BG} | x), & \text{if } f_i = 0, \end{cases}$$

where the posterior probabilities have been computed with (8). This term enforces a per-pixel constraint, and encourages the labeling to coincide with the per-pixel observation.

The second part $D_i^2(f_i)$ enforces the temporal consistency constraint to the labeling. We assume that a pair of associated pixels in consecutive images should have the same label. The current frame (time t) is back-projected to the previous frame (time $t-1$) by estimating the optical flow, so that each pixel $i \in I$ is associated with a pixel v in the previous frame. Since the label f_v has been obtained, $D_i^2(f_i)$ can be defined as

$$D_i^2(f_i) = \begin{cases} 0, & \text{if } f_i = f_v, \\ \gamma, & \text{if } f_i \neq f_v, \end{cases}$$

where $\gamma > 0$ is a weight parameter, and is used to penalize the inconsistent labelings. Considering that optical flow may be erroneous due to noises, large motion, and boundary effects, we choose a small weight: $\gamma = 0.5$.

Taking two parts together, the data term becomes $D_i(f_i) = D_i^1(f_i) + D_i^2(f_i)$. However, it should be noted that if the frame rate of a video is low, the temporal contextual constraint cannot be used, then we have $D_i(f_i) = D_i^1(f_i)$.

The smoothness term $W(f_i, f_u)$ encourages the spatial continuity in the labeling. A cost is paid when two neighboring pixels have different labels. We define the term W as

$$W(f_i, f_u) = \begin{cases} 0, & \text{if } f_i = f_u, \\ \phi * Z(I_i, I_u), & \text{if } f_i \neq f_u, \end{cases}$$

where $\phi = 5.0$ is a weight parameter, and $Z(I_i, I_u)$ is a decreasing function that is controlled by the intensity difference between the pixels i and u . In general, the discontinuity of

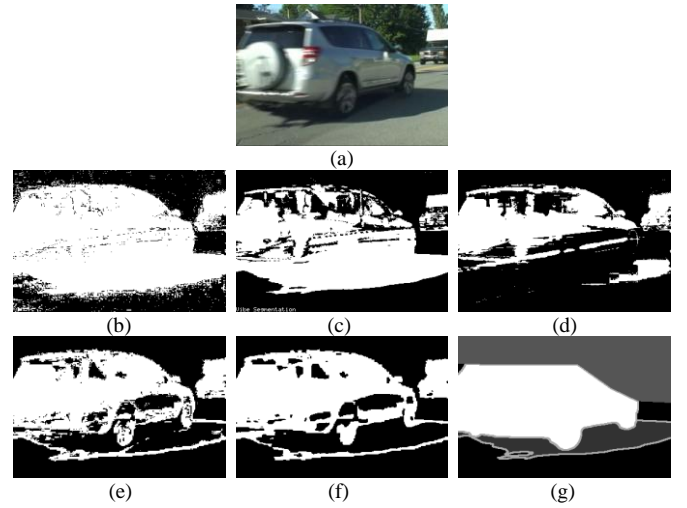


Fig. 15. Comparative foreground/background segmentation results of five methods for one frame taken from the “bungalows” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

segmentation should coincide with the image discontinuity. Hence we choose the function Z as

$$Z(I_i, I_u) = \exp\left(-\frac{\|I_i - I_u\|^2}{\sigma_I}\right),$$

where σ_I is the variance parameter, fixed to 400 in this work.

The optimal label is found by using the loopy belief propagation algorithm. Although belief propagation is exact only when the graph structure has no loop, in practice it has been proved to be an effective approximate inference technique for general graphical models [45]. In the experiments, we declare convergence when the relative change of messages is less than a threshold 10^{-4} . Fig. 12(c) illustrates the inference result of loopy belief propagation. As expected, the segmented foreground is very close to the ground truth shown in Fig. 12(d).

In summary, the block diagram of the proposed method is shown in Fig. 13.

VI. EXPERIMENTAL RESULTS

A. Test Videos and Evaluation Metric

To verify the proposed method we conduct experiments on six benchmark videos. Five videos and their ground truth labels are publicly available at changedetection.net [5], [46]. Another video is available at AVSS 2007 website [47]. However, its ground truth is not provided, so we take some frames uniformly from this video and label them by hand. The challenges contained in these videos are as below:

“Highway” contains shadows and swaying trees.

“Bungalows” contains shadows.

“Backdoor” contains shadows, intermittent shades, rapid illumination changes, and swaying trees.

“AVSS” contains shadows and rapid illumination changes.

“TunnelExit” contains large image noises and swaying trees, and has a low frame rate.

“Turnpike” contains mild image noises and has a low frame

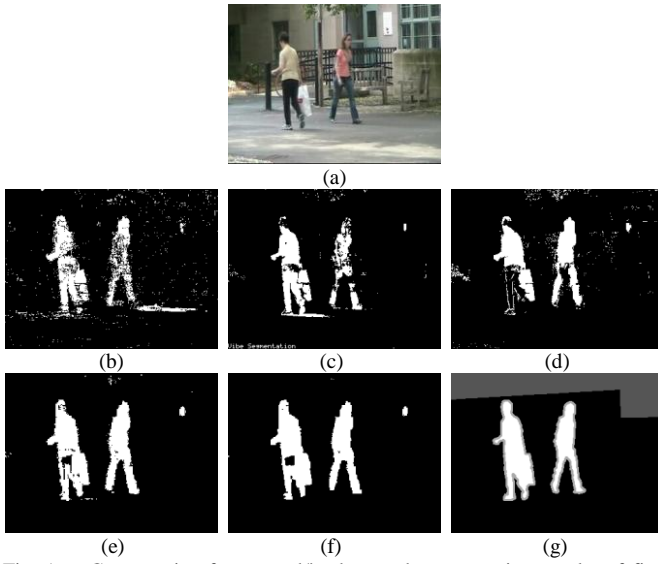


Fig. 16. Comparative foreground/background segmentation results of five methods for one frame taken from the “backdoor” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

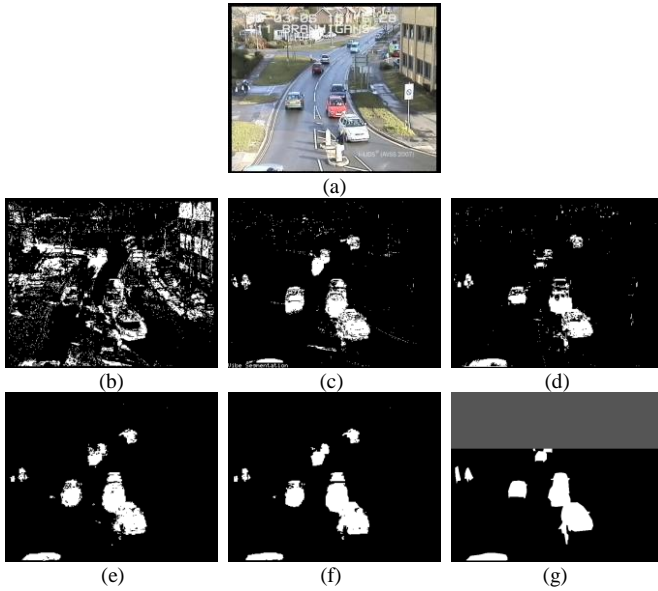


Fig. 17. Comparative foreground/background segmentation results of five methods for one frame taken from the “AVSS” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

rate.

To justify a foreground detection method, an evaluation metric must be selected. Let TP = number of true positives, FP = number of false positives, and FN = number of false negatives. The evaluation metric we use is the *F-measure*, which is the harmonic mean of the *Recall* and *Precision*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{TP + FN}, \\ \text{Precision} &= \frac{TP}{TP + FP}, \\ F\text{-measure} &= 2 \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}. \end{aligned}$$

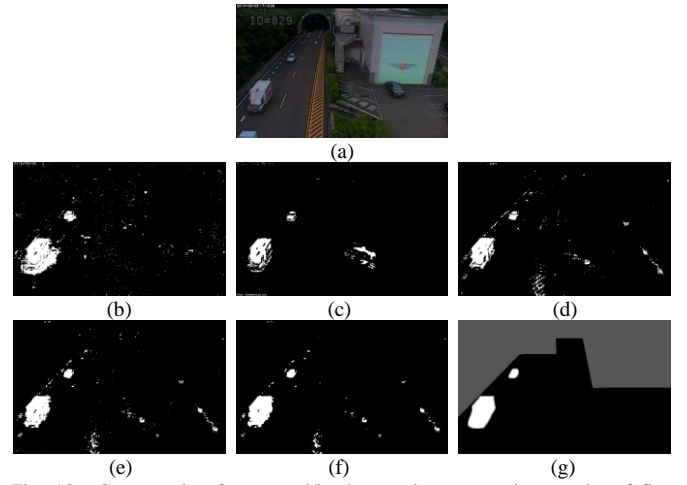


Fig. 18. Comparative foreground/background segmentation results of five methods for one frame taken from the “tunnelExit” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

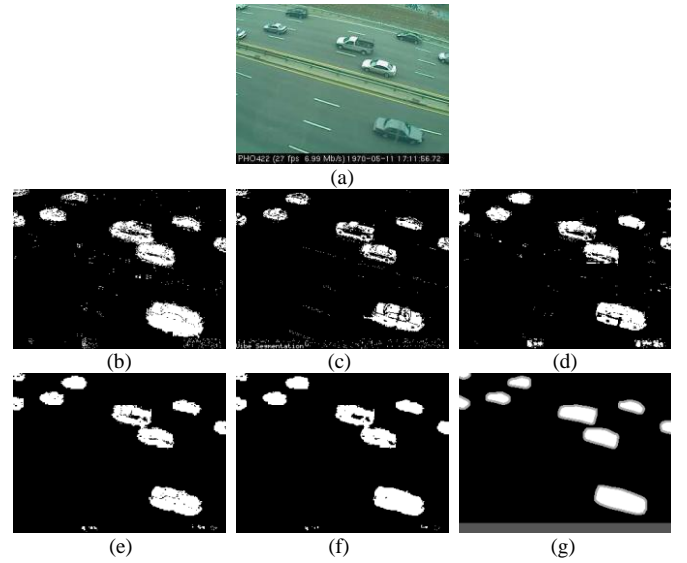


Fig. 19. Comparative foreground/background segmentation results of five methods for one frame taken from the “turnpike” video. (a) Input image. (b) GMM [18]. (c) ViBe [34]. (d) Lam [38]. (e) Rounding the pixel-wise foreground posterior. (f) Proposed method. (g) Ground truth.

Note that the *F-measure* needs to be as high as possible, in order to minimize the segmentation errors.

B. Comparison with Other Methods

The proposed method is compared qualitatively and quantitatively with three existing methods: GMM [18], ViBe [34], and Lam [38]. GMM is a baseline for foreground detection, and has been evaluated by many researchers. In the MATLAB software, a system object called “vision.ForegroundDetector” is offered to detect foreground using Gaussian mixture models. We use it directly. ViBe is a recently proposed method, and its inventors have publicized the program at their project website [48]. Both GMM and ViBe use only pixel intensities as features. By contrast, Lam *et al.* [38] combined luminance, chrominance, and texture features in their model, but they detected foreground based on hard-thresholding. We implement this method in

TABLE II
QUANTITATIVE COMPARISON OF FIVE METHODS ON THE TEST VIDEOS

	Method	Highway	Bungalows	Backdoor	AVSS	TunnelExit	Turnpike	Mean
Recall	GMM [18]	0.5182	0.9653	0.9106	0.5298	0.8158	0.8735	0.7689
	ViBe [34]	0.7962	0.8012	0.7847	0.7627	0.5039	0.6825	0.7219
	Lam [38]	0.6577	0.5073	0.6842	0.7041	0.5489	0.8542	0.6594
	Rounding foreground posterior	0.9407	0.7017	0.8463	0.8641	0.6646	0.9130	0.8217
	Proposed method	0.9567	0.7030	0.8429	0.8815	0.6531	0.9252	0.8271
Precision	GMM [18]	0.3636	0.6406	0.4615	0.3499	0.3650	0.7048	0.4809
	ViBe [34]	0.8590	0.6756	0.6894	0.5591	0.7787	0.8500	0.7353
	Lam [38]	0.6155	0.8410	0.6985	0.7695	0.4662	0.8474	0.7064
	Rounding foreground posterior	0.8736	0.8667	0.9381	0.7515	0.4349	0.9022	0.7945
	Proposed method	0.9160	0.8912	0.9528	0.7849	0.5231	0.9213	0.8316
F-measure	GMM [18]	0.4273 (5)	0.7701 (3)	0.6126 (5)	0.4214 (5)	0.5043 (4)	0.7801 (4)	0.5860 (5)
	ViBe [34]	0.8264 (3)	0.7330 (4)	0.7339 (3)	0.6452 (4)	0.6119 (1)	0.7571 (5)	0.7179 (3)
	Lam [38]	0.6359 (4)	0.6328 (5)	0.6913 (4)	0.7353 (3)	0.5042 (5)	0.8508 (3)	0.6751 (4)
	Rounding foreground posterior	0.9059 (2)	0.7755 (2)	0.8898 (2)	0.8039 (2)	0.5258 (3)	0.9075 (2)	0.8014 (2)
	Proposed method	0.9359 (1)	0.7860 (1)	0.8945 (1)	0.8304 (1)	0.5809 (2)	0.9233 (1)	0.8252 (1)

MATLAB. It should be noted that for a fair comparison, all the methods are stripped of any postprocessing operation.

In this work, it is critical to incorporate the spatial-temporal contextual constraints into the foreground/background decision model. However, during the computation we also record the binary classification result by rounding the pixel-wise foreground posterior of (8), and this result is analyzed to verify the necessity of using spatial-temporal context.

Fig. 14–19 show the comparative foreground/background segmentation results of five methods for six typical frames of the test videos. Foreground and background pixels are shown in white and black respectively.

1) *Highway*: an example from this video is shown in Fig. 14. GMM and ViBe fail to remove shadows, and classify many foreground pixels as background. GMM also classifies many background pixels of the swaying trees as foreground. Lam [38] succeeds in removing shadows, but causes many false negatives and false positives. Many foreground pixels are classified as background, and many background pixels of the swaying trees are classified as foreground. By contrast, the proposed method can remove shadows when the vehicles move to the bottom of the image and the shadow areas are big enough. Meanwhile, very few segmentation errors are caused.

2) *Bungalows*: an example from this video is shown in Fig. 15. Once again for this video, GMM and ViBe fail to remove shadows. Lam [38] is able to remove most of the shadows, but classifies many foreground pixels as background. The proposed method can remove shadows effectively. However, because the vehicle body has little texture and similar color with the background, it causes some false negatives, but fewer than those caused by ViBe and Lam [38].

3) *Backdoor*: an example from this video is shown in Fig. 16. Shadows in this video are complex, and may be soft, hard, or

intermittent. Besides, rapid illumination changes occur irregularly. GMM and ViBe are still affected by the shadows, and classify many pixels on the human bodies as background. Lam [38] is insensitive to shadows, but negates many true foreground pixels, such as pixels on the legs of the left person in Fig. 16(d). By contrast, the proposed method is hardly affected by shadows and segments out the persons accurately.

4) *AVSS*: an example from this video is shown in Fig. 17. Rapid illumination changes occur frequently in this video. GMM is affected the most, classifying many background pixels as foreground. The other methods are insensitive to illumination changes. ViBe and Lam [38] classify many foreground pixels on the vehicle bodies as background. By contrast, the proposed method causes very few segmentation errors.

5) *TunnelExit*: an example from this video is shown in Fig. 18. There are large image noises in this video, and many objects have little texture and similar color with the background. GMM, Lam [38], and the proposed method cause many false positives, classifying the noisy background pixels as foreground. ViBe and Lam [38] cause many false negatives, classifying many foreground pixels on the vehicle bodies as background.

6) *Turnpike*: an example from this video is shown in Fig. 19. There are mild image noises in this video. GMM is affected the most, causing many false positives. The other methods are less sensitive to the mild noises. However, ViBe and Lam [38] classify many foreground pixels on the vehicle bodies as background. By contrast, the proposed method causes very few segmentation errors.

The recalls, precisions, and F-measures of five methods on the test videos are reported in Table II. Note that the numbers in bold indicate the best performance for the metrics, and the F-measures are ranked. As can be seen, the proposed method outperforms other competing methods on five test videos. The

TABLE III
COMPUTATIONAL TIME FOR PROCESSING ONE FRAME

Module	Programming language	Time (second)
Multi-features extraction	C MEX	1.4
Density estimation via multi-view learning	MATLAB	0.1
Pixel soft-labeling with Bayes rule	MATLAB	1.2
Pixel labeling with belief propagation	C MEX	1.6

only exception is the highly noisy video “*tunnelExit*”, where ViBe performs the best. This is perhaps because ViBe has the ability of representing multimodal background. By contrast, the proposed method generates reference background images by using temporal median filtering, making it essentially a single-modal model. Despite this, our method outperforms GMM and Lam [38] under this adverse environment. According to the mean F-measure metric, our method is the best one.

On the other hand, if we analyze the experimental results (see Fig. 14–19 and Table II) of rounding the pixel-wise foreground posterior and other methods, we can acquire some new findings. Due to the use of multiple heterogeneous features and multi-view learning, the pixel-wise foreground posterior computed with equation (8) is rather informative. Leaving out the spatial-temporal contextual constraints in the video sequence, rounding the pixel-wise foreground posterior has outperformed GMM [18], ViBe [34], and Lam [38], according to the mean metrics. Introducing the MRF framework helps to further improve the foreground segmentation accuracy, with the mean F-measure increased by 2.38%.

C. Computational Cost

The proposed method is implemented on a PC with 2.50GHz Intel Core i5-3210M CPU and 4G memory. The multi-features extraction module and the pixel labeling with belief propagation module are implemented using C MEX, and the rest modules are implemented using MATLAB. The computational time is monitored by the “tic” and “toc” functions. Taking the video “highway” (with 320×240 pixel resolution) for example, the average computational time for processing one frame is about 4.3 seconds. Cost details of the main modules are shown in Table III. The implementation at its present stage cannot achieve real-time processing. In the future, we will study efficient MRF inference algorithms and use parallel computing platforms such as GPU to speed up the computation.

D. Limitations of the Proposed Method

In this work, we update the reference background image once every 50 frames, by conducting TMF over the recent 500 frames. That way, gradual scene changes can be handled, but in practice it is sometimes expected that the TMF learning period (time window) varies according to the surroundings. For instance, when the traffic volume is low, the learning period is expected to be properly short, in order to capture the recent scene change and to reduce the computational cost; when the traffic volume is high, the learning period is expected to be long enough, in order

to prevent foreground objects from being absorbed into the background image. However, traffic volume is a kind of semantic information. According to Toyama *et al.* [11], background subtraction as a low-level module should not attempt to extract the semantics of foreground objects on its own. Of course, background subtraction may work as a component of larger systems that seek high-level understanding of image sequences. In that case, it would be feasible to feed traffic volume back to the background subtraction module and adjust the TMF learning period accordingly.

Since the conditional probability density of the foreground is estimated via multi-view learning, it is expected that there are enough foreground objects in the scene. For a video sequence with only one or two objects, the estimated foreground model may be unreliable. However, in practice several objects would be enough. For example, although there are only 7 foreground objects in the entire video “backdoor”, our approach is able to detect foreground pixels accurately, as shown in Fig. 16. In long-term traffic surveillance, because many objects (including vehicles and pedestrians) usually appear at irregular intervals, this problem can be avoided.

Another potential problem is that the proposed method cannot represent the multimodal background. Due to our proposed iterative search and multiscale fusion strategy in feature extraction, slight background movement is tolerable. However, dramatic camera shakes, highly dynamic background, and large image noises may deteriorate the performance of our method. Solving this problem will be our future work.

VII. CONCLUSION

For detecting foreground objects from traffic environments, this paper proposed an effective multi-view learning method to model both the foreground and the background. First of all, multiple heterogeneous features including brightness variation, chromaticity variation, and texture variation are extracted from the video sequence. These features are robust to shadows and illumination changes, and are approximately conditionally independent given the class label. A multi-view learning method is devised to online estimate the conditional densities for both foreground and background. This makes our method different from many existing ones, which build only the background model and recognize foreground pixels purely as outliers. Pixel soft-labeling is conducted using Bayes rule and pixel-wise foreground posteriors are estimated. Finally, under the MRF framework, spatial-temporal contextual constraints are incorporated into the foreground/background decision model, and optimal foreground segmentation is achieved by belief propagation.

The experimental results have verified the usefulness and effectiveness of the proposed method. Quantitative and qualitative comparisons with the existing methods have shown that when confronted with challenges like shadows, illumination changes, and mild image noises, our method can improve the accuracy of foreground detection. Some limitations of our method have been discussed. As improvements to our method, multimodal background modeling and computation speedup

will be our future work.

REFERENCES

- [1] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.
- [2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
- [3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.
- [4] F. Porikli, F. Bremond, S. L. Docket, J. Ferryman, A. Hoogs, B. C. Lovell, S. Pankanti, B. Rinner, P. Tu, and P. L. Venetianer, "Video surveillance: Past, present, and now the future [DSP Forum]," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 190–198, May 2013.
- [5] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "A novel video dataset for change detection benchmarking," *IEEE Trans. Image Processing*, vol. 23, no. 11, pp. 4663–4679, Nov. 2014.
- [6] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11–12, pp. 31–66, May 2014.
- [7] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, 2011.
- [8] S. Brutzer and B. Hferlin, G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [9] D. H. Parks and S. S. Fels, "Evaluation of background subtraction algorithms with post-processing," *Proc. IEEE Fifth Int'l Conf. Advanced Video and Signal Based Surveillance*, pp. 192–199, 2008.
- [10] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Computer Vision and Image Understanding*, vol. 122, pp. 4–21, May 2014.
- [11] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," *Proc. IEEE Int. Conf. Computer Vision*, Sept. 1999, pp. 255–261.
- [12] G. Zhao, X. Wang, and W.-K. Cham, "Background subtraction via robust dictionary learning," *EURASIP Journal on Image and Video Processing*, Feb. 2011.
- [13] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.
- [14] D.-M. Tsai and S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Processing*, vol. 18, no. 1, pp. 158–167, Jan. 2009.
- [15] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.
- [16] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pffinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jun. 1997.
- [17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Ft. Collins, CO, Jun. 1999, vol. 2, pp. 246–252.
- [18] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.
- [19] D. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.
- [20] N. Martel-Brissou and A. Zaccarin, "Learning and removing cast shadows through a multidistribution approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1133–1146, July 2007.
- [21] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Statistical background subtraction using spatial cues," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 12, pp. 1758–1763, Dec. 2007.
- [22] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of Gaussians-based background subtraction for bayer-pattern image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 365–370, Mar. 2011.
- [23] D. Mukherjee, Q. M. J. Wu, and T. M. Nguyen, "Multiresolution based Gaussian mixture model for background suppression," *IEEE Trans. Image Processing*, vol. 22, no. 12, pp. 5022–5035, Dec. 2013.
- [24] D. Mukherjee, Q. M. J. Wu, and T. M. Nguyen, "Gaussian mixture model with advanced distance measure based on support weights and histogram of gradients for background suppression," *IEEE Trans. Industrial Informatics*, vol. 10, no. 2, pp. 1086–1096, May 2014.
- [25] M. Haque and M. Murshed, "Perception-inspired background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 12, pp. 2127–2140, Dec. 2013.
- [26] T. S. F. Haines and T. Xiang, "Background subtraction with Dirichlet process mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, Apr. 2014.
- [27] R. H. Evangelio, M. Patzold, I. Keller, and T. Sikora, "Adaptively splitted GMM with feedback improvement for the task of background subtraction," *IEEE Trans. Information Forensics and Security*, vol. 9, no. 5, pp. 863–874, May 2014.
- [28] F.-C. Cheng, S.-C. Huang, and S.-J. Ruan, "Scene analysis for object detection in advanced surveillance systems using Laplacian distribution model," *IEEE Trans. Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 41, no. 5, pp. 589–598, Sept. 2011.
- [29] W. Zhang, X. Z. Fang, X. K. Yang, and Q. M. J. Wu, "Moving cast shadows detection using ratio edge," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1202–1214, Oct. 2007.
- [30] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, July 2002.
- [31] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
- [32] A. R. Rivera, M. Murshed, J. Kim, and O. Chae, "Background modeling through statistical edge-segment distributions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1375–1387, Aug. 2013.
- [33] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proc. IEEE Int. Conf. Computer Vision FRAME-RATE Workshop*, 1999.
- [34] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Processing*, vol. 20, no. 6, pp. 1709–1724, June 2011.
- [35] M. Heikkilä and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.
- [36] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1301–1306.
- [37] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.
- [38] W. W. L. Lam, C. C. C. Pang, and N. H. C. Yung, "Highly accurate texture-based vehicle segmentation method," *Opt. Eng.*, vol. 43, no. 3, pp. 591–603, Mar. 2004.
- [39] B. Han and L. S. Davis, "Density-based multifeature background subtraction with support vector machine," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 1017–1023, May 2012.
- [40] L. Maddalena and A. Petrosino, "ViBe: A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1168–1177, July 2008.
- [41] M. I. Chacon-Murguia and S. Gonzalez-Duarte, "An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems," *IEEE Trans. Industrial Electronics*, vol. 59, no. 8, pp. 3286–3298, Aug. 2012.
- [42] Z. Liu, K. Huang, and T. Tan, "Cast shadow removal in a hierarchical manner using MRF," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 56–66, Jan. 2012.
- [43] S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, "Region-level motion-based background modeling and subtraction using MRFs," *IEEE Trans. Image Processing*, vol. 16, no. 5, pp. 1446–1456, Mar. 2007.

- [44] Y. Wang, K.-F. Loe, and J.-K. Wu, "A dynamic conditional random field model for foreground and shadow segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 279–289, Feb. 2006.
- [45] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
- [46] Changedetection.net, <http://www.changedetection.net/>.
- [47] 2007 IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS 2007), http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html.
- [48] ViBe - a powerful technique for background detection and subtraction in video sequences, <http://www.motiondetection.org/>.



Kunfeng Wang received the Bachelor degree in Materials Science and Engineering from Beihang University, Beijing, in 2003 and the Ph.D. degree in Control Theory and Control Engineering from the Graduate University of Chinese Academy of Sciences, Beijing, in 2008.

Since 2008, he has been an Assistant Professor in The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include

intelligent video analytics, intelligent transportation systems, and machine learning.



Yuqiang Liu received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2010 and the Ph.D. degree in Control Theory and Control Engineering from the University of Chinese Academy of Sciences, Beijing, in 2015. He is now a postdoctor at China Academy of Railway Sciences.

His research interests include computer vision, intelligent transportation systems, and probabilistic graphical models.



Chao Gou received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. He is currently working toward the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is an intern at Qingdao Academy of Intelligent Industries, Qingdao, China.

His research interests include intelligent transportation systems, image processing, and pattern recognition.



Fei-Yue Wang (S'87–M'89–SM'94–F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the University of Arizona, Tucson, AZ, USA, in 1990, and became a Professor and Director of the Robotics and Automation Laboratory and the Program for Advanced Research in Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center with the Chinese

Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Overseas Chinese Talents Program. Since 2002, he has been the Director of the Key Laboratory of Complex Systems and Intelligence Science with CAS, and is currently the Director of the State Key Laboratory of Management and Control for Complex Systems, as well as the Director of the Research Center for Computational Experiments and Parallel Systems Technology, National University of Defense Technology. His current research interests include social computing, web science, complex systems, and intelligent control.

Prof. Wang was the recipient of the National Prize in Natural Sciences of China and the Outstanding Scientist Award by ACM for his work in intelligent

control and social computing, in 2007. He was an Editor-in-Chief of the International Journal of Intelligent Control and Systems and the World Scientific Series in Intelligent Control and Intelligent Automation, from 1995 to 2000. He is currently an Editor-in-Chief of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He has served as chairs for over 20 IEEE, ACM, INFORMS, and ASME conferences. He was the President of IEEE Intelligent Transportation Systems Society, from 2005 to 2007, the Chinese Association for Science and Technology, New York, NY, USA, in 2005, and the U.S. Zhu Kezhen Education Foundation, New York, from 2007 to 2008. He is currently the Vice President of the ACM China Council and the Vice President/Secretary-General of Chinese Association of Automation. He is a member of Sigma Xi and an Elected Fellow of INCOSE, IFAC, ASME, and AAAS.