Spatio-Temporal Traffic Scene Modeling for Object Motion Detection

JiuYue Hao, Chao Li, Zuwhan Kim, and Zhang Xiong

***Abstract*—Moving object detection is an important component of a traffic surveillance system. Usual background subtraction approaches often poorly perform on a long outdoor traffic video due to vehicles waiting at an intersection and gradual changes of illumination and background shadow position. We present a fast and robust background subtraction algorithm based on unified spatio-temporal background and foreground modeling. The correlation between neighboring pixels provides high levels of detection accuracy in the dynamic background scene. Our Bayesian fusion method, which establishes the traffic scene model, combines both background and foreground models and considers prior probabilities to adapt changes of background in each frame. We explicitly model both temporal and spatial information based on the kernel density estimation (KDE) formulation for background modeling. Then, we use a Gaussian formulation to describe the spatial correlation of moving objects for foreground modeling. In the updating step, a fusion background frame is generated, and reasonable updating rates are also proposed for the traffic scene. The experimental results show that the proposed method outperforms the previous work with less computation and is better suited for the traffic scenes.**

***Index Terms*—Bayesian method, real-time traffic surveillance system, scene modeling, spatio-temporal modeling.**

I. Introduction

A COUNTLESS number of traffic surveillance cameras and intelligent devices have become a part of our daily lives. The information they provide affects processes such as personal path planning, safety enhancement, and policy making [1]. In most of these processes and applications, traffic object detection is a fundamental and critical task. A common approach for moving object detection is the background modeling method.

Manuscript received May 12, 2011; revised November 1, 2011, March 3, 2012, and June 18, 2012; accepted July 25, 2012. Date of publication October 18, 2012; date of current version February 25, 2013. This work was supported by the National High Technology Research and Development Program of China (2011AA010502), the National Science and Technology Pillar Program (2012BAH07B01), and the State Key Laboratory of Software Development Environment (SKLSDE-2012ZX-04). The Associate Editor for this paper was N. Papanikolopoulos.

J. Y. Hao is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: haojiuyue@gmail.com).

C. Li is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Shenzhen Key Laboratory of Data Vitalization, Research Institute in Shenzhen, Beihang University, Beijing 100191, China (e-mail: licc@buaa.edu.cn).

Z. Kim is with the California Partners for Advanced Transportation Technology, University of California, Berkeley, CA 94804 USA (e-mail: zuwhan@gmail.com).

Z. Xiong is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Shenzhen Key Laboratory of Data Vitalization, Research Institute in Shenzhen, Beihang University, Beijing 100191, China (e-mail: xiongz@buaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2012.2212432

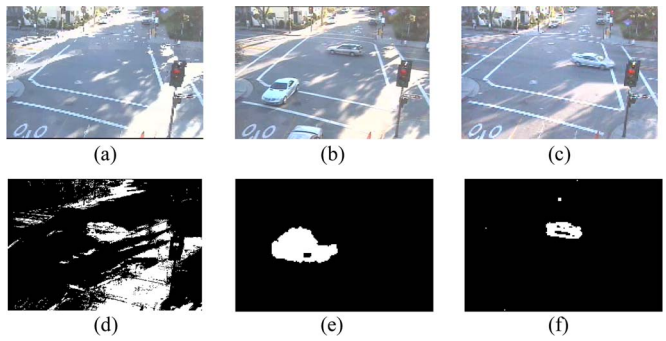


Fig. 1. Background subtraction is still a challenging problem. The original frames are from an hour-long traffic surveillance video at a real intersection. Note the tree shadow direction changes in the background. The second row is the detection results of (c). Two typical background subtraction algorithms [[5] and [19] for (d) and (e)] fail. Our algorithm (f) shows a more robust result.

Even though many background modeling algorithms have been proposed in the available research literature, the problems of detecting or extracting moving objects in a complex outdoor environment on a real-time system are still far from being fully resolved.

As shown in Fig. 1, in the traffic surveillance system, there are many challenges.

1) A background scene dramatically changes over time by the shadow of background objects (e.g., trees) and varying illumination, as shown in Fig. 1(a)–(c).
2) The shadow moves with the wind in the trees, which makes the detection result too noisy, as shown in Fig. 1(d).
3) The moving objects have similar colors to those of the road and the shadow; thus, the background may be falsely detected as an object [Fig. 1(d)].
4) The vehicles waiting for a signal light corrupt the background model.
5) The computation time needs to be low because most applications require real-time detection.

To address these challenges, we propose a spatio-temporal scene model for moving object detection in a dynamic scene. Some of the distinctive features included in this work are here.

1) A Bayesian fusion method is introduced to combine both background and foreground models.
2) A spatio-temporal model based on kernel density estimation (KDE) is introduced to handle dynamic background; the Gaussian foreground model is used to describe the spatial correlation of foreground pixel and reduce computation.
3) The frame fusion algorithm is proposed for a robust updating stage.

II. Related Work

Various background modeling algorithms for complex outdoor scenes have been proposed [1]. The most popular approaches for background modeling are based on probabilistic models. In these methods, the probability distribution of the pixel values is estimated by a number of different techniques. The first category of research establishes the temporal information of the background model.

Earlier works based on the background pixel values, such as the W4 system [3] and the LBP sequence [4], have low time cost. In 1999, Stauttfer and Grimson [5] proposed the mixture of Gaussians (MoG), which is based on the temporal probability density modeling of individual pixels. It is the most popular temporal modeling approach. In [6], Cheung compared various background subtraction algorithms, including the median filter, approximated median filter, Kalman filter, and MoG in urban traffic video sequences. Cheung's experimental results demonstrated that the MoG achieves the best precision and recall. Many algorithms have been proposed to improve the MoG and its updating process, particularly the autodecision strategy of parameters [7], [8]. Tuzel *et al.* [9] proposed estimating the probability distribution of the mean and covariance of each Gaussian distribution using recursive Bayesian learning. Han *et al.* [10] introduced a density model that dynamically detected Gaussian components by the variable-bandwidth mean shift and allowed the number of modes to adapt in time. Klare and Sarkar [11] improved the MoG background model classifiers based on the original color and Haar features to adapt to significant changes in illumination.

However, when the actual density function has a large number of peaks or if a certain peak constantly changes, it is hard to describe the density model of each pixel by parametric functions. To address some of these issues, Elgammal *et al.* [12] used a KDE approach, where they represented a background model by individual pixels of the last N frames. This method did not require any assumptions on a density model and worked well for complex dynamic scenes. Based on a nonparametric density estimation method, Mittal and Paragios [13] proposed a motion-based background model, which utilized two components for the optical flow and three components for the intensity in the normalized color space. Caseiro *et al.* [14] defined a nonparametric Riemannian framework on a tensor field with an application to foreground segmentation. However, nonparametric density estimation needs a large storage memory, has high time cost, and is hard to apply to real-time systems.

The second category includes both spatial and temporal aspects into the background modeling. Cristani and Murino [15] proposed a spatial-time adaptive per pixel MoG (S-TAPPMOG) by considering a neighborhood zone throughout a sampling process. Liu *et al.* [16] represented an information saliency map, which was calculated from spatio-temporal volumes with both spatial and temporal saliencies' models by KDE. Babacan and Pappas [17] presented a novel MoG background subtraction based on a Bayesian formulation. A Gibbs–Morkov random field helped to exploit spatiotemporal dependencies between pixels in [17].

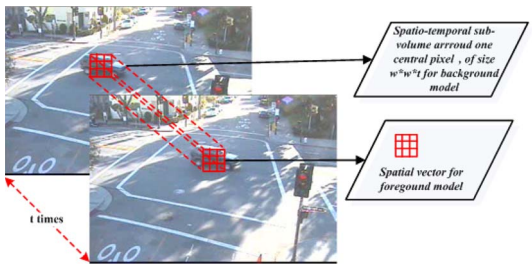


Fig. 2. Spatio-temporal model of one pixel, with w as the width of window size and t as the symbol of times.

Recently, both background and foreground models have been described with spatial and temporal information. Mahadevan and Vasconcelos [18] equated background subtraction to the problem of saliency detection using spatio-temporal patches, which establish both the center and surrounding models. The most notable work is by Sheikh and Shah [19], which employed RGB features and location features into models, and both the background and foreground were by KDE to augment the detection of objects. Here, spatial information was implicitly modeled by adding location features.

In this paper, we present a Bayesian formulation using temporal coherence for combining both background and foreground models. We calculate both foreground and background probabilities using the Bayesian rule. The Bayesian fusion method is part of our modeling method. It ensures that the foundation of modeling is better than many other literatures. For example, Sheikh and Shah [19] separated the modeling part and the MAP-MRF part. MAP-MRF does not have any contribution for calculating background and foreground modeling. It just used the result of the probability to classify the background and foreground pixels. In addition, the MRF method is a high time cost and cannot be used in a long-term surveillance system.

Second, our simple scene modeling method ensures high efficiency and accuracy. We use a KDE method to represent the spatial-temporal background and a single Gaussian function to represent *only* the spatial foreground model. Moving objects do not have uniform features; therefore, it is difficult to describe them during different periods of time with a statistic model. Thus, only neighboring spatial information is introduced for modeling the foreground model.

III. Spatio-Temporal Scene Model

In this section, we give the details of our Bayesian fusion method, which is part of our modeling method. It has contributions to model both background and foreground and to give different weights to fusing them as a uniform scene modeling. As shown in Fig. 2, the model includes both temporal relations and spatial context.

A. Background and Foreground Model Fusion

For detecting a moving object, each pixel can be classified into a background or a foreground pixel. Let x_t be the feature vector (color and position) of a pixel A in a frame at time t. In this paper, the feature vector x_t is a joint domain-range representation [19], where the space of the image lattice is the

domain (x, y) and some color space, e.g., (r, g, b). We define a random variable B_t given by

$$B_t = \begin{cases} \text{bg}_t, & \text{if } x_t \text{ belongs to the background at time } t \\ \text{fg}_t, & \text{if } x_t \text{ belongs to the foreground at time } t \end{cases}.$$

By applying the Bayesian rule, we consider those prior probabilities at $t-1$ to establish the model at time t. The probability $P(\text{bg}_t|X_t, X_t^s)$ that observing x_t belongs to a background, with $X_{t-1} = \{x_1, x_2, \ldots, x_{t-1}\}$ and $X_t^s = \{x_1', x_2', \ldots, x_s'\}$, is given as follows:

$$\begin{aligned} P\left(\text{bg}_t|X_t, X_t^s\right) &= P\left(\text{bg}_t|x_t, X_{t-1}, X_t^s\right) \\ &= \frac{P\left(x_t|\text{bg}_t, X_{t-1}, X_t^s\right) P\left(\text{bg}_t|X_{t-1}, X_t^s\right)}{P\left(x_t|X_{t-1}, X_t^s\right)} \end{aligned} \tag{1}$$

where X_{t-1} is a temporal vector set with a $t-1$ element before time t.

We define a window centered at pixel A, and w is the width of this window. x_s' is one neighboring pixel vector in the window, and X_t^s is the neighborhood vector set with all neighbors.

The normalization factor is given by

$$\begin{aligned} &P\left(x_t|X_{t-1}, X_t^s\right) \\ &= \sum_{B_t \in \{\text{bg}_t, \text{fg}_t\}} P\left(x_t|B_t, X_{t-1}, X_t^s\right) P\left(B_t|X_{t-1}, X_t^s\right). \end{aligned} \tag{2}$$

We can rewrite the equation by using probability density functions

$$\begin{aligned} &P\left(\text{bg}_t|x_t, X_{t-1}, X_t^s\right) \\ &= \frac{f\left(x_t|\text{bg}_t, X_{t-1}, X_t^s\right) P\left(\text{bg}_t|X_{t-1}, X_t^s\right)}{\sum_{B_t \in \{\text{bg}_t, \text{fg}_t\}} f\left(x_t|B_t, X_{t-1}, X_t^s\right) P\left(B_t|X_{t-1}, X_t^s\right)}. \end{aligned} \tag{3}$$

The prior probability can be obtained by applying marginalization on B_{t-1}, i.e.,

$$\begin{aligned} P(\text{bg}_t|X_{t-1}, X_t^s) = &\sum_{B_{t-1} \in \{\text{bg}_{t-1}, \text{fg}_{t-1}\}} P(\text{bg}_t|B_{t-1}, X_{t-1}, X_t^s) \\ &\times P\left(B_{t-1}|X_{t-1}, X_t^s\right) \end{aligned} \tag{4}$$

where $P(\text{bg}_t|B_{t-1}, X_{t-1}, X_t^s) = P(\text{bg}_t|B_{t-1})$ is the probability of the status changes. For example, $P(\text{bg}_t|\text{bg}_{t-1})$ is the probability that a background pixel at $t-1$ remains in the background at time t. $P(\text{bg}_t|\text{fg}_{t-1})$ is a probability that a foreground pixel at $t-1$ becomes part of the background at time t. We assume that the neighboring pixels' temporal changes from time $t-1$ to t have little effects; thus, we assume that

$$P\left(B_{t-1}|X_{t-1}, X_t^s\right) \approx P\left(B_{t-1}|X_{t-1}, X_{t-1}^s\right). \tag{5}$$

Finally, function (4) can be computed as

$$\begin{aligned} P\left(\text{bg}_t|X_{t-1}, X_t^s\right) = &\sum_{B_{t-1} \in \{\text{bg}_{t-1}, \text{fg}_{t-1}\}} P(\text{bg}_t|B_{t-1}) \\ &\times P\left(B_{t-1}|X_{t-1}, X_{t-1}^s\right). \end{aligned} \tag{6}$$

B. Background Modeling

Based on the definition of x_t, X_t, and X_t^s in Section III-A, we now introduce the method of background modeling.

We establish the background model combining both the temporal set and the spatial set. $Y = \{X_t, X_t^s\} = \{y_1, \ldots, y_i, \ldots, y_N\}$, and N is the number of samples. The probability background density function in (3) is expressed as follows:

$$\begin{aligned} f_{\text{bg}}(x_t) &= f\left(x_t|\text{bg}_t, X_{t-1}, X_t^s\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} K_H(x_t - y_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \|H\| K\left(H^{-1/2}(x_t - y_i)\right) \end{aligned} \tag{7}$$

where K is the kernel estimator function and is a Gaussian kernel in this paper. H is the bandwidth matrix and calculated as $H(y_i) = h(y_i)I$ [19].

Finally, $f_{\text{bg}}(x_t)$ can be solved as follows:

$$\begin{aligned} f_{\text{bg}}(x_t) = (2\pi)^{-\frac{d}{2}} N^{-1} \sum_{i=1}^{N} H^{\frac{-1}{2}} \\ \times \exp\left(-\frac{1}{2}(x_t - y_i)^T H^{-1}(x_t - y_i)\right) \end{aligned} \tag{8}$$

where d is the dimension of x_t.

C. Foreground Modeling

Based on background modeling, we get a temporary judgment on the class of each pixel. The foreground model is built using these foreground pixel candidates. We assume that pixels belonging to a foreground object have uniform features. In other words, if a pixel is detected as a foreground, its nearby pixels will have high probability of being foreground pixels. Thus, we just utilize the spatial coherence and define $f_{\text{fg}}(x_t) = f(x_t|\text{fg}_t, X_{t-1}, X_t^s) \approx f(x_t|\text{fg}_t, X_t^s)$.

Note that most of the previous work [16]–[19] calculated the foreground model by KDE using temporal vector X_{t-1}. However, it is difficult to describe the quickly changed foreground, which includes too many different features. Thus, we just use neighbor spatial to establish an accurate spatial foreground model. It is important to note that the single Gaussian model is accurate in a small neighborhood.

For the foreground candidate pixel, let x_t be the RGB color feature vector and $X_t^s = \{x_1', x_2' \ldots x_s'\}$ be its neighbor vector set in a $w \times w$ region, where w is the width of the window. To calculate the foreground probability of a pixel in the current frame, a Gaussian distribution is used to model the spatial color distribution as follows:

$$f_{\text{fg}}(x_t) = f\left(x_t|\text{fg}_t, X_{t-1}, X_t^s\right) \approx f\left(x_t|\text{fg}_t, X_t^s\right) \tag{9}$$

where μ_t^s is the mean, and $\sum_t^s$ is the covariance of X_t^s, i.e.,

$$\mu_t = \frac{1}{s}\sum_{i=1}^{s} x_i' \tag{10}$$

$$\sum_t = \frac{1}{S-1}\sum_{i=1}^{s} \left(x_i' - \mu_t\right)^2. \tag{11}$$

Fig. 3. Background fusion result. (a) Original frame. (b) Original frame. (c) Fusion frame for updating.

D. Updating and Detection Stage

We adapt a blind-update mechanism for the background modeling, where all pixels are used to build the background model without classification. This also helps remove ghosts, where the stain of a false positive remains in the background model. In contrast, the selective-update mechanism is introduced for the foreground, which updates the model by a pixel classified as foreground.

In our paper, we update the background and foreground models after ρ_b and ρ_f frames, respectively. Note that the background model does not have to be frequently updated. In fact, it is better *not* to update the background model in every frame to minimize the corruption by slow-moving or stopping foreground objects. On the other hand, the foreground model needs short-term updating time, which ensures a correct real-time object motion model.

In the surveillance video, the frames that include a pure background scene are very few. For sample selection in the background-updating stage, we propose a new strategy. This strategy fuses the video frames into one pure background scene to avoid any motion object being added into the background model. The method selects pixels that have the largest background probability at each location to construct a new frame described as follows:

```
If updateNum <= ρ_b
  updateim = (prob_t <= prob_t − 1) * updateim +
(prob_t > prob_t − 1) * currentim;
else
    // reinitialize
    updateim = currentim;
    updateNum = 1.
```

As shown in Fig. 3, during one updating period, although the traffic scene includes many moving objects, our fusion method can successfully extract a pure background scene in Fig. 3(c).

For traffic scene analysis, we need to consider vehicles waiting for the traffic signals, where the wait time can be as long as 90 s. We consider that the frame rate of the surveillance system is 25 frame/s, and an object can stay motionless for up to 2250 frames. However, our fusion background image will slow down the process of object merger into the background scene. Thus, we set the background update frames ρ_b to be 100–500 frames (4–20 s), which is also reasonable for adapting changes in illumination. To keep the real-time change of a moving object, we set the foreground update frame ρ_f at one frame (0.04 s).

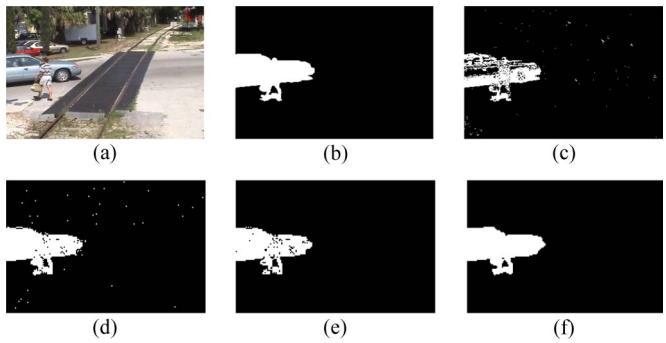


Fig. 4. Background subtraction in a nominally moving camera. (a) Original frame. (b) Ground truth. (c) MoG result. (d) CMU KDE. (e) Spatial-KDE. (f) Our ST method.

Although we use a more complex background model (spatio-temporal), but because the background update rate is extremely low, the extra computation is manageable. For the foreground model, which requires updating every frame, we use a fast Guassian foreground model. Therefore, the overall computation is significantly lessened.

Finally, foreground detection is done by a simple likelihood ratio classifier, where κ is a threshold value, i.e.,

$$\delta(x_t)\begin{cases}0, & \text{if } -\ln\frac{P(\text{bg}_t|X_t,X_{t-1}^s)}{P(\text{fg}_t|X_t,X_{t-1}^s)} > \kappa \\ 1, & \text{otherwise}\end{cases}. \tag{12}$$

IV. Experimental Results

In this section, we present a set of experimental results. All the algorithms were implemented with Matlab on a Windows XP platform. We applied our model (spatio-temporal model) to detect moving foreground objects in various real video sequences. The experiments are composed of two parts. First, the proposed method is compared with existing methods, i.e., MoG [5], KDE [19], and Spatial-MoG [17], using video sequences in [19]. Second, we evaluate the performance of our proposed method using a 1-h traffic video sequence.

Our threshold for the foreground detection κ was chosen as -1. The number of neighboring pixels was four. For the MoG, a three-component mixture was modeled with a learning rate of 0.01, and the threshold probability $p(x_t)$ was chosen as 10^{-7}. All three methods used 200 frames at initial stages, and shadow removal and postprocesses were not used in the presentation of these results.

A. Evaluation of the Proposed Method

The data set of [19] was taken from a camera mounted on a tall tripod and included 500 frames. Due to wind, the tripod sways back and forth, causing nominal motion of the camera. The results are shown in Fig. 4. Fig. 4(a) and (b) shows the original frame and ground truth, respectively. From the MoG result in Fig. 4(c), it is evident that the nominal motion of the camera causes degradation in the performance with high false negatives. Fig. 4(d) shows the result of KDE in CMU, which has better recall rate than MoG. To provide evidence for the benefit of spatial information in the background model, we use the spatial information algorithm (S-MoG) in [17] for comparison.

TABLE I
AVERAGE PRECISION AND RECALL

Method	Precision	Recall
MoG	89.02%	70.25%
KDE[19]	90.90%	73.98%
S-MoG[17]	95.87%	79.96%
ST-KDE	95.92%	84.22%

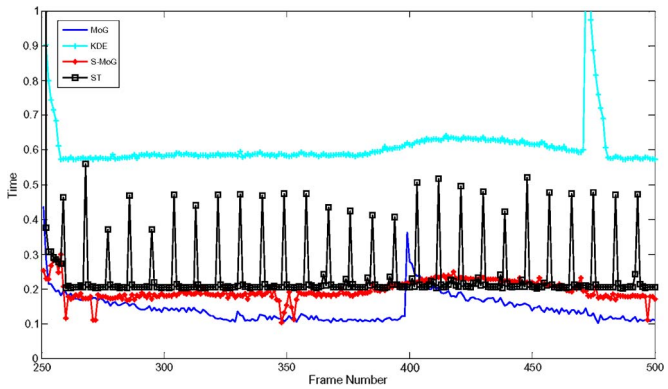


Fig. 5. Comparison of execution time.

The result in Fig. 4(e) shows that S-MoG can effectively eliminate noise in the background. Our spatial-temporal model (ST) with Bayesian fusion information, Gaussian foreground model, and fusion background is shown in Fig. 4(f), which successfully deals with background noise. Our ST method also removes holes in the foreground found in the S-MoG method. It performs best in detecting the integrated moving object.

Then, quantitative experimentation was performed on these methods. The precision and recall rates for all the methods were compared with these four methods. Overall results are shown in Table I, which shows that our proposed ST-KDE model has an average precision of 95.92% and an average recall of 84.22%, which are better than those of S-MoG, KDE, and MoG.

We compared the execution time of all four methods (excluding the initialization background step). As shown in Fig. 5, our method is at the second position because of the Gaussian function used in the foreground. This video is short; therefore, for our ST-model method, we updated the background every ten frames, and the original KDE is updated for every frame. To sum up, the KDE method took an average of 0.619 s to process one frame. The S-MoG and MoG took 0.144 and 0.195 s respectively. Our method (ST-KDE) took 0.26 s. To summarize, our approach only uses 40% of the computation, compared to KDE. Note that the experiments were done on Matlab, and with code optimization, it can be used for real-time applications. In a long video, we do not need to quickly update the background (as discussed in Section III-D). The background model will be updated for every 400 frames. The execution time will be lower than in a short video.

B. Comparison on Traffic Video

Instead of using short video clips, as presented in the previous section, we present the results of an hour-long traffic video gathered at a real intersection. The whole video includes

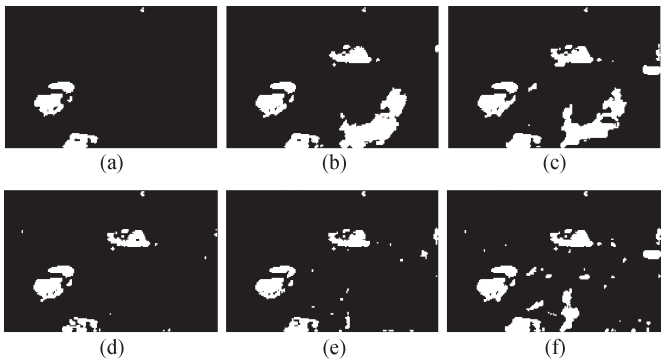


Fig. 6. Parameter comparison. Note that the number is the value of ρ_b. (a) KDE-10. (b) KDE-50. (c) KDE-100. (d) ST-100. (e) ST-400. (f) ST-500.

moving shadows over the time of day and wind effects. The traffic scene includes various types of objects, such as pedestrians, bicycles, motorcycles, and vehicles, and various events, such as turning and waiting for signal lights. During the short experiment, we prove that the scene model, including both foreground and background models (ST), is better than the background model, such as S-MoG. Note that most of the recent algorithms just use short video clips. In our paper, we use a long time video to illustrate our algorithm and prove that our algorithm can be used in a real surveillance system. An example of comparative results at different times is shown in Fig. 1 first. Both MoG and KDE fail with the moving shadow.

First, the updating rate is a significant parameter for background modeling. For fair comparison, we tried various background update rates, which lead to the highest recall and precision rate for both KDE and ST-KDE in a traffic surveillance system. The original frame, which is at 5 min, is shown in Fig. 1(b).

As shown in Fig. 6, the first row includes results from KDE modeling, with ρ_b every 10, 50, and 100 frames. The second row includes the results from ST modeling, with ρ_b every 100, 400, and 500 frames. A small update frame fails to distinguish the moving object from the background. It eventually fails detection on most vehicles, as shown in Fig. 6(a), which is based on the KDE model with a ten-frame update rate for background modeling. As shown in Fig. 6(b) and (c), the experimental results become better, with the KDE update frame increasing to 50 and 100 frames. However, because the update frame is too high for the KDE model, it cannot adapt to the background changes and creates too much false positives.

Our ST model can successfully detect entire objects with different update frames and has better results than KDE. However, the false positive becomes high when ρ_b is overly increased in Fig. 6(f).

According to our analysis in Section III-D, to deal with vehicles waiting for the signal or temporarily stopping, if the update time is too low, the vehicle will quickly merge into the background. The reason that our model does not need to frequently update the background model is that our Bayesian fusion method adds prior probability $(P(B_{t-1}|X_{t-1}, X^s_{t-1}))$ at $t-1$ into our ST model. We recalculate our ST model with temporal information at $t-1$ time in every frame. Finally, we chose the KDE ρ_b at 50 frames and the ST update rate at 400 frames for further comparison. Our ST update frames

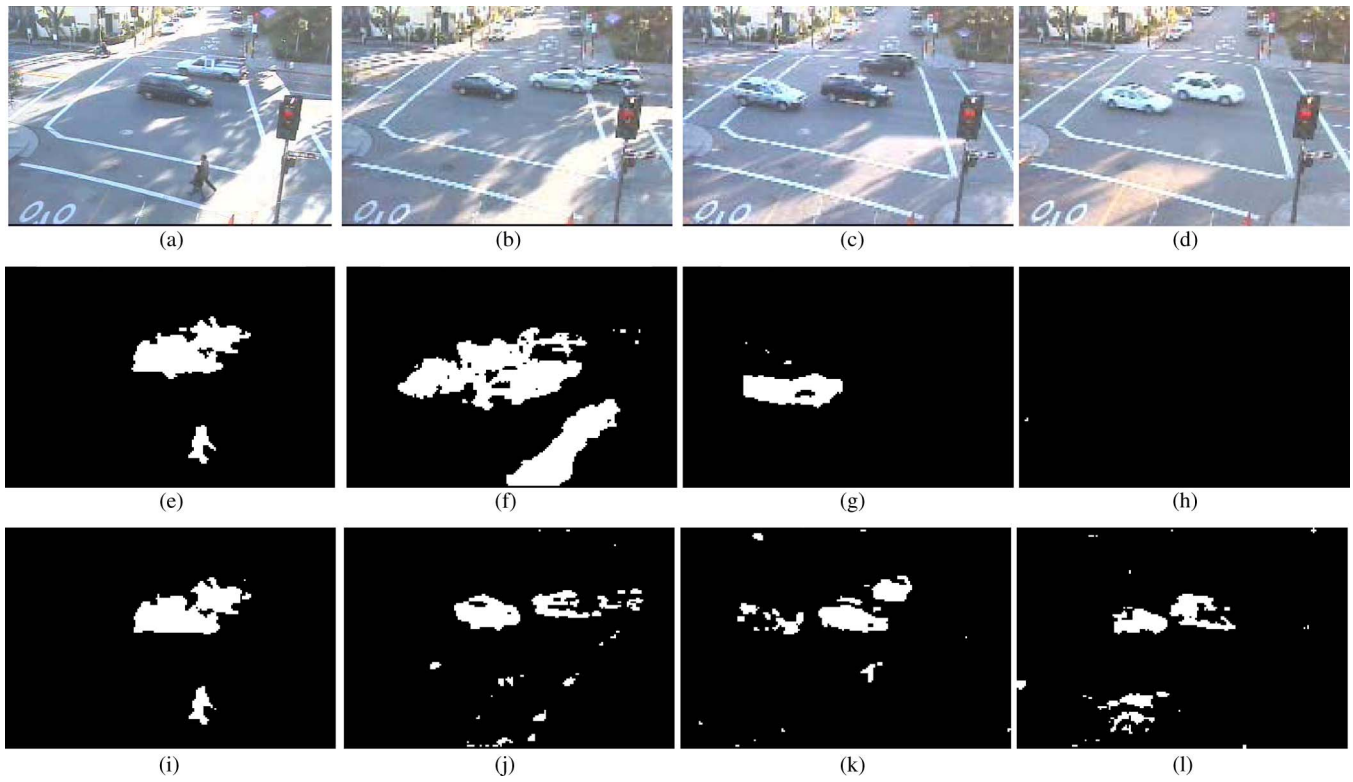


Fig. 7. The set of results is shown on KDE, with our proposed method (ST) during 1-hr traffic surveillance. (a) Original frame at 00:00:53. (b) Original frame at 00:12:45. (c) Original frame at 00:37:20. (d) Original frame at 00:47:04. (e)–(h) KDE. (i)–(l) ST.

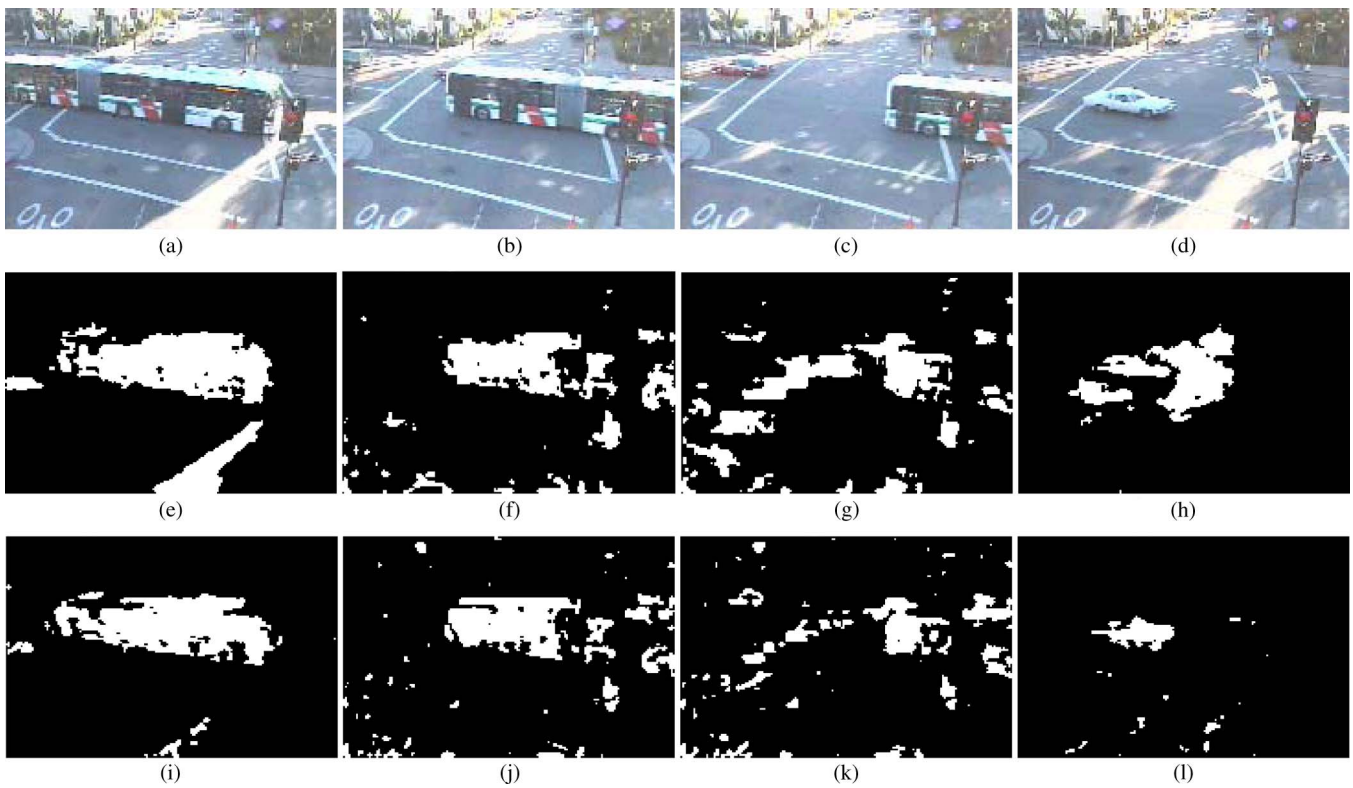


Fig. 8. Recovery of the background model after quick camera shaking. In the time of (d) and (g), the camera quickly shakes. (a) Original frame at 00:20:14. (b) Original frame at 00:20:15. (c) Original frame at 00:20:16. (d) Original frame at 00:20:17. (e)–(h) KDE. (i)–(l) ST.

provide an additional advantage by further reducing the computation time and better detection of the vehicles waiting for the signals.

Example results are shown in Fig. 7. Our proposed method obtains the best results in different periods of time and successfully eliminates both shadow shake and illumination change.

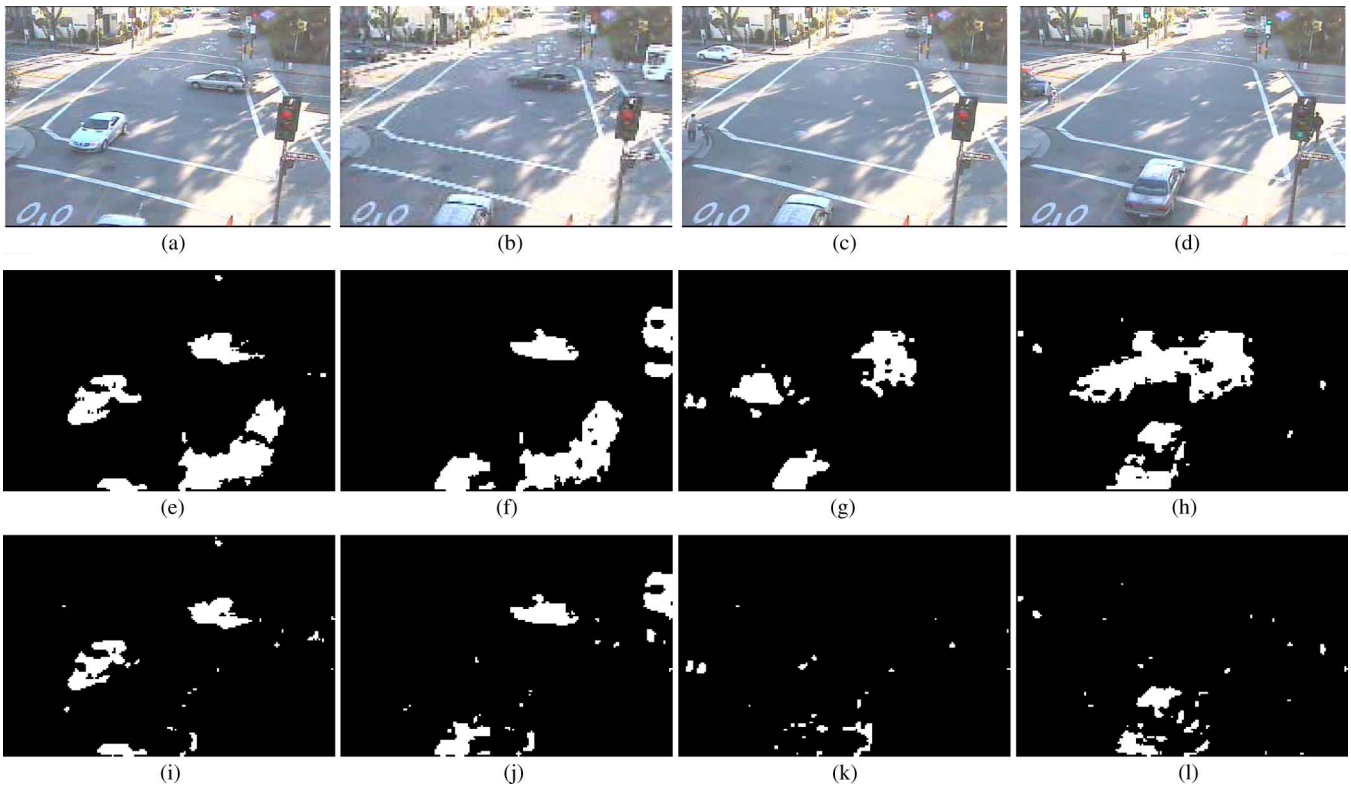


Fig. 9. Detection of the vehicle waiting for the signal light (for 84 s). The status is changing from coming to waiting to leaving. (a) Coming at 00:14:14. (b) Waiting at 00:14:16. (c) Waiting at 00:15:40. (d) Leaving at 00:15:43. (e)–(h) KDE. (i)–(l) ST.

The results with the KDE model illustrate two main defects, i.e., overfitting [Fig. 7(g) and (h)] and high false positive rate using temporal information in the foreground [Fig. 7(f) and (g)].

The results in Fig. 8 are shown in a difficult scenario where the background rapidly changes due to camera jitter or quick illumination change. As shown in Fig. 8(g) and (k), the detection results become terribly poor. After camera shaking, our ST model obtains a great result after sudden background changes happened in 1 s, as shown in Fig. 8(l).

Finally, we show our approach in detecting vehicles waiting for the red signal light or stopping for a short time period, as shown in Fig. 9. The waiting time for the vehicle was 84 s. In our ST model, our update rate can successfully deal with different waiting times. As shown in Fig. 9(k), the vehicle waiting for the traffic signal was going to merge into the background after 86 s and was redetected after restarting in Fig. 9(l). However, the KDE model failed to deal with different objects waiting in the background. The object will never merge into the background, as shown in Fig. 9(g), which is not reasonable in a traffic scene.

V. Conclusion

A robust spatio-temporal scene model (ST-KDE) has been introduced in this paper. The ST-KDE model handles both rapidly (such as shaking trees or camera jitter) and slowly (moving shadows over the time of day) changing background and still detects temporarily stopped objects (vehicles standing for traffic lights). ST-KDE has shown better performance, compared to previous approaches, and has been computationally efficient enough for real-time applications.

Our future work will concentrate on exploring a shadow elimination algorithm and a solution for the disappearance of small objects that have features similar to the road.

References

[1] J. Y. Hao, H. Shen, C. Li, Z. Xiong, and E. Hussain, "Vehicle behavior understanding based on movement string," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, 2009, pp. 243–248.

[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understanding*, vol. 104, no. 2/3, pp. 90–126, Nov./Dec. 2006.

[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[4] J. Yao and J. M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE CVPR*, 2007, pp. 1–8.

[5] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE CVPR*, 1999, pp. 246–252.

[6] S. C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Proc. SPIE-Int. Soc. Opt. Eng.*, vol. 5308, pp. 881–892, 2004.

[7] S. Atev, O. Masoud, and N. Papanikolopoulos, "Practical mixtures of Gaussians with brightness monitoring," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, 2004, pp. 423–428.

[8] Y. Sun and B. Yuan, "Hierarchical GMM to handle sharp changes in moving object detection," *Electron. Lett.*, vol. 40, no. 13, pp. 801–802, Jun. 2004.

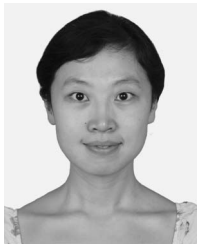
[9] O. Tuzel, F. Porikli, and P. Meer, "A Bayesian approach to background modeling," in *Proc. IEEE CVPR*, 2005, p. 58.

[10] B. Han, D. Comaniciu, and L. Davis, "Sequential kernel density approximation trough mode propagation: Application to background modeling," in *Proc. ACCV*, 2004, pp. 1–6.

[11] B. Klare and S. Sarkar, "Background subtraction in varying illuminations using an ensemble based on an enlarged feature set," in *Proc. IEEE CVPR*, 2009, pp. 66–73.

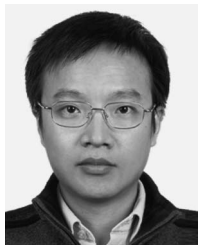
[12] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. ECCV*, 2000, pp. 751–767.

[13] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE CVPR*, 2004, pp. 302–309.
[14] R. Caseiro, J. Henriques, P. Martins, and J. Batista, "A nonparametric Riemannian framework on tensor field with application to foreground segmentation," in *Proc. IEEE ICCV*, 2011, pp. 1–8.
[15] M. Cristani and V. Murino, "A spatial sampling mechanism for effective background subtraction," in *Proc. VISAPP*, 2007, pp. 403–412.
[16] C. Liu, P. C. Yuen, and G. Qiu, "Object motion detection using information theoretic spatio-temporal saliency," *Pattern Recognit.*, vol. 42, no. 11, pp. 2897–2096, Nov. 2009.
[17] S. D. Babacan and T. N. Pappas, "Spatiotemporal Algorithm for Background Subtraction," in *Proc. IEEE ICASSP*, 2007, pp. I-1065–I-1068.
[18] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scene," in *Proc. IEEE CVPR*, 2008, pp. 1–6.
[19] 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.



JiuYue Hao was born in 1984. She received the Bachelor's degree from Communication University of China, Beijing, China, in 2006 and the Ph.D. degree from Beihang University, Beijing, in 2012.

She was a Research Engineer with The First Research Institute of Ministry of Public Security, Beijing. She is currently with the School of Computer Science and Engineering, Beihang University. She was a Visiting Student with California Partners for Advanced Transportation Technology, University of California, Berkeley. Her research interests include intelligent transportation systems, pattern recognition, and computer vision.



Chao Li received the Bachelors and Ph.D. degrees in computer science and technology from Beihang University, Beijing, China, in 1996 and 2005, respectively.

He is currently an Associate Professor and Master's Supervisor with the School of Computer Science and Engineering, Beihang University. His current research interests include data vitalization and computer vision.

Zuwhan Kim received the Bachelors and Master's degrees from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1993 and 1996, respectively, and the Ph.D. degree from the University of Southern California, Los Angeles, in 2001, all in computer science.

He is currently a Research Engineer and Development Engineer with the California Partners for Advanced Transportation Technology, University of California, Berkeley. He has extensive experience in both basic and applied research and multidisciplinary experience with a wide variety of academic fields, including intelligent transportation systems, robotics, geographic information systems, statistics, and cognitive science. He has been a Reviewer for renowned journals including *Machine Vision and Application* (a regular reviewer), *Transportation Research Part-C*, and *Photogrammetric Engineering and Remote Sensing*. His research interests are computer vision and pattern recognition.

Dr. Kim has served as a Reviewer for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the IEEE TRANSACTIONS ON IMAGE PROCESSING.

Zhang Xiong received the Bachelors degree from Harbin Engineering University, Harbin, China, in 1982 and the Masters degree from Beihang University, Beijing, China, in 1985.

He is currently a Professor and Ph.D. Supervisor with the School of Computer Science and Engineering, Beihang University. His research interests include computer vision, wireless sensor networks, and information security.