

# A Probabilistic Framework Based on KDE-GMM Hybrid Model for Moving Object Segmentation in Dynamic Scenes

Zhou Liu, Wei Chen, Kaiqi Huang, Tieniu Tan

► **To cite this version:**

Zhou Liu, Wei Chen, Kaiqi Huang, Tieniu Tan. A Probabilistic Framework Based on KDE-GMM Hybrid Model for Moving Object Segmentation in Dynamic Scenes. The Eighth International Workshop on Visual Surveillance - VS2008, Oct 2008, Marseille, France. 2008. <inria-00325761>

**HAL Id: inria-00325761**

**<https://hal.inria.fr/inria-00325761>**

Submitted on 30 Sep 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Probabilistic Framework Based on KDE-GMM Hybrid Model for Moving Object Segmentation in Dynamic Scenes

Zhou Liu, Wei Chen, Kaiqi Huang and Tieniu Tan  
National Laboratory of Pattern Recognition,  
Institute of Automation, Chinese Academy of Sciences, Beijing, P.R.China  
{zliu, wchen, kqhuang, tnt}@nlpr.ia.ac.cn

## Abstract

*In real scenes, dynamic background and moving cast shadow always make accurate moving object detection difficult. In this paper, a probabilistic framework for moving object segmentation in dynamic scenes is proposed. Under this framework, we deal with foreground detection and shadow removal simultaneously by constructing probability density functions (PDFs) of moving objects and non-moving objects. Here, these PDFs are constructed based on KDE-GMM hybrid model (KGHM) which has advantages of KDE and GMM. This KGHM models the spatial dependencies of neighboring pixel colors to deal with highly dynamic scenes. Moreover, in this framework, tracking information is used to refine the PDF of moving objects. Experimental results demonstrate the effectiveness of our method.*

## 1 Introduction

Real-time segmentation of moving objects is a crucial step in visual surveillance systems. This is because subsequent processes, such as tracking and behavior recognition, are heavily dependent on its output. Background subtraction is a commonly used technique to detect moving objects in videos, as it can provide a more complete set of feature data describing the moving targets compared with other approaches [5]. Accurate moving object detection could be difficult due to potential variability, such as moving cast shadow and dynamic background. Therefore, an accurate and robust algorithm for real scenes is needed.

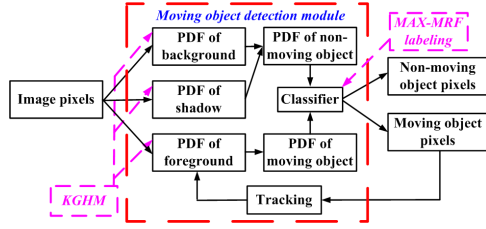
Background modeling can be regarded as a classification problem (foreground or background), and some probability density estimation methods have been used, such as Gaussian Mixture Model(GMM) [14] and Kernel Density Estimation (KDE) [3]. GMM is a widely used approach due to its self learning capacity and its robustness to variations in lighting. However, it still has some shortcomings.

The number of Gaussians should be decided beforehand. Another limitation is that it does not explicitly model the spatial dependencies of neighboring background pixel colors. Therefore, some false positive pixels will be produced in highly dynamic scenes where dynamic texture does not repeat exactly [13](as shown in Section 2 and 3). Another density estimation method used for background modeling is KDE. In [3], the authors use the KDE method to represent the color distribution for each pixel. Then, Sheikh and Shah [13] directly model the dependencies between the domain (location) and the range (color) by using a non-parametric density estimation method over a joint domain-range representation of image pixels. They also introduce a foreground model to improve detection results. However, this method does not consider moving cast shadow, which will cause problems, such as object merging and shape distortion. Besides these, the authors in [7] propose a histogram based method to estimate the color distribution.

Some other methods which are different from density estimation methods have also been used, such as Kalman filter [6] and auto-regressive model method [10]. The Kalman filter is employed to update slow and gradual changes in the background. However, it will fail when dealing with dynamic background. In [10], the authors propose an on-line auto-regressive model to capture and predict the behavior of the dynamic scenes.

The proposed method has two novel contributions:

- 1) A probabilistic framework for moving object segmentation in dynamic scenes is proposed. Under this framework as in Figure 1, we deal with foreground detection and shadow removal simultaneously by constructing PDFs of moving objects and non-moving objects. These PDFs can be constructed based on GMM[14], KDE[13, 3] and histogram[7]. Therefore, the method in [13] can be considered as a special case of this framework, if we neglect the PDF of shadow and the tracking information. The method in [12] also models background, foreground and shadow, simultaneously, based on HMM. However, the model parameters are learned off-line, and without additional constraints,



**Figure 1. Block diagram of the proposed method.**

this method fails to distinguish shadow from dark vehicles. Moreover, in our framework, tracking information is used to refine the PDF of foreground. The advantages of unifying background modeling, moving object detection and shadow removal will be discussed in Section 2.4.

2) An KDE-GMM hybrid model(KGHM) is proposed to construct the PDFs of background, foreground and shadow. It overcomes some shortcomings of both KDE and GMM for estimating the PDFs. Compared with GMM, it models spatial dependencies of neighboring pixels color explicitly. Moreover, the KGHM can determine the number of Gaussian components in the corresponding GMM part, dynamically, by Gaussian merging and deleting rules. The comparison details with KDE in [13] are given Section 2.1.

## 2 Framework for Moving Object Segmentation Based On KGHM

In this section, to segment moving objects, we first deduce the KGHM for PDFs of background, foreground and shadow. It is inspired by [13], [4]. Then, a three-category classification problem(background, foreground and shadow) is converted to a two-category problem by constructing PDFs of moving objects and non-moving objects. At last, a classifier, such as likelihood ratio classifier, can be used for segmentation. Here, to enforce spatial context, the MAP-MRF labeling method is used. Furthermore, the feedback of tracking is used to refine the PDF of foreground. The block diagram of the proposed framework is shown in Figure 1. It is worthwhile pointing out that, in this paper, the term "moving object" which excludes shadow refers to a part of foreground, "GMM" refers to the method in [14], and "shadow" refers to moving cast shadow.

### 2.1 KGHM for background modeling

To deduce the KGHM, as in [13], the feature vector  $x$  also works in domain-range space, where the coordinates of pixel are the domain represented by  $s = (x, y)$ , and the RGB color space is the range by  $c = (r, g, b)$ . Therefore,

we represent  $p$  pixels by  $x_i \in IR^5, i = 1, 2, \dots, p$  and  $x_i = (s_i, c_i) = (x_i, y_i, r_i, g_i, b_i)$ . This makes the background represented by a single model. We first construct the PDF of background at time  $t$  by KDE as

$$f(x|b) = n^{-1} \sum_{i=1}^n K_H(x - y_i) \quad (1)$$

where the samples  $y_1, y_2, y_3 \dots y_n$  are the pixels obtained before time  $t$  and they are five-dimensional vectors, and

$$K_H(x) = |H|^{-1/2} K(H^{-1/2}x) \quad (2)$$

where  $K$  is a five-variate kernel function and  $H$  is a symmetric positive definite  $5 \times 5$  bandwidth matrix. We assume the domain component and the range component are independent with each other, then Equation 1 becomes

$$f(x|b) = n^{-1} \sum_{i=1}^n K_{Hs}(s - s_i) K_{Hc}(c - c_i) \quad (3)$$

where  $s_i$  and  $c_i$  are the domain component and range component of  $y_i$ ,  $s$  and  $c$  are components of  $x$ .  $Hs$  and  $Hc$  are the corresponding bandwidth matrices. Obviously, Equation 3 can be rewritten as next formula:

$$f(x|b) = n^{-1} \sum_{i=1}^{CN} K_{Hc}(c - c_i) \left( \sum_{j=1}^{SN_i} K_{Hs}(s - s_j) \right) \quad (4)$$

where  $CN$  is the number of different color values and  $SN_i$  is the number of samples whose color values equal  $c_i$ . Then, we "grid or bin" the domain space and use  $g_j$  to represent the center coordinates of the  $j^{th}$  grid(bin). For simplicity, the width of bin along  $x$  and  $y$  directions is equal. Equation 4 can be approximated by rules similar to the binned kernel density estimators [4] as

$$f(x|b) \approx n^{-1} \sum_{i=1}^{CN} K_{Hc}(c - c_i) \left( \sum_{j=1}^{BN} N_{ij} K_{Hs}(s - g_j) \right) \quad (5)$$

where  $BN$  is the number of bins an image contains, and  $N_{ij} = \sum_{a=1}^{SN_i} \omega_j(s_a, \delta)$ .  $\delta$  is the width of bin, and the weight  $\omega_j(s_a, \delta)$  means that the observed data value  $s_a$  should be contributed to  $g_j$  [4]. Rearrange Equation 5, we get

$$f(x|b) \approx \sum_{j=1}^{BN} \frac{N_j}{n} K_{Hs}(s - g_j) \left( \sum_{i=1}^{CN} \frac{N_{ij}}{N_j} K_{Hc}(c - c_i) \right) \quad (6)$$

where  $N_j = \sum_{i=1}^{CN} N_{ij} = \sum_{i=1}^{CN} \sum_{a=1}^{SN_i} \omega_j(s_a, \delta)$ .

Then, simple binning is used [4]. The corresponding  $\omega_j(s_a, \delta)$  is as follows:

$$\omega_j(s_a, \delta) = \begin{cases} 1, & \text{if } \|s_a - g_j\|_{\infty} < \delta/2, \\ 0, & \text{otherwise;} \end{cases} \quad (7)$$

where  $\|\cdot\|_\infty$  is the infinite norm. Obviously,  $N_j$  is the number of samples which fall in the  $j^{th}$  bin. In the  $N_j$  samples, there are  $N_{ij}$  samples whose range components equal  $c_i$ . Then, Equation 6 can be rewritten as

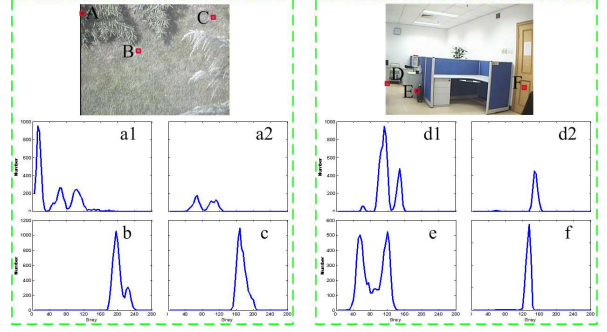
$$f(x|b) \approx \sum_{j=1}^{BN} c_b K_{Hs}(s - g_j) \left( \frac{1}{N_j} \sum_{z=1}^{N_j} K_{Hc}(c - c_{jz}) \right) \quad (8)$$

where  $c_{jz}$  represents the range component of samples whose corresponding domain part falls in the  $j^{th}$  bin and the factor,  $c_b$ , equals  $\frac{N_j}{n}$ , which is a constant allowing for equal area for every bin. The expression,  $N_j^{-1} \sum_{z=1}^{N_j} K_{Hc}(x - c_{jz})$ , can be seen as kernel density estimation for the marginal distribution of range component, only allowing for the samples falling in the  $j^{th}$  grid.

We consider the colors of pixels belonging to a certain bin as a "bin process" compared to "pixel process". Some "bin processes" belonging to different grid are shown in Figure 2 which illustrates that a multi-modal representation is needed for the data. Then, we assume the color values of the samples belonging to the same bin fit Gaussian Mixture distribution, and Equation 8 can be approximated as

$$f(x|b) \approx \sum_{j=1}^{BN} c_b K_{Hs}(s - g_j) \left( \sum_{i=1}^{M_j} \omega_{ji} G_{\sigma_{ji}}(c - \mu_{ji}) \right) \quad (9)$$

where  $M_j$  is the number of Gaussian components in the  $j^{th}$  bin,  $G_\sigma(\cdot)$  is the Gaussian function with variance  $\sigma$ ,  $\mu$  is the mean, and  $\omega_{ji}$  is the weight of the  $i^{th}$  Gaussian in the  $j^{th}$  bin. It is the final expression of KGHM for background, which is a hybrid representation of KDE and GMM. The bandwidth selection of  $K_{Hs}$  is according to the dynamic degree of background. If we set  $\delta = 1$  and  $K_{Hs}(s - g_j)$  an indicator function  $1(\|s - g_j\|_\infty < 0.5)$ , then the density estimation of Equation 9 is equivalent to the traditional GMM [14]. Obviously, the spatial information is fused in two ways: first, KGHM is constructed in grid(bin) level; then, the part  $K_{Hs}(s - g_j)$  is used to model the dependencies between a sample and its neighboring bins. Compared with the KDE method in [13] which also fuses spatial information, our method does not need to store samples, although kernel function is used. This is because the coordinates of every pixel which are described by kernel function don't change over time. This also makes the bandwidth selection for the domain (determined by the dynamic degree of background) is much easier than it for the range. Moreover, the proposed model also does not need to estimate the bandwidth for the range component. In a word, KDE is used for the domain part since the coordinates of each pixel do not change and the bandwidth selection for this part is easier, and GMM is used for the range part since the color distribution of a pixel can be learned adaptively. Therefore, KGHM has the advantages of KDE and GMM for estimating the PDFs.



**Figure 2. Empirical distributions of intensity values for different bins (grids) in indoor and outdoor environments. Histograms a1,b,c,d1,e,f correspond to bins A...F, respectively, and the width of the bins is 4. Histograms a2 and d2 correspond to bins whose width is 2, and they are part of bins A and D, respectively. It illustrates that histograms a2 and d2 which correspond to smaller bins have less peaks than a1 and d1, and that more Gaussians are usually needed for the bins which are situated on the boundary or show dynamic motion, such as A and D.**

## 2.2 Determination of the Gaussian components by Gaussian Merging and Deleting Rules

After fusing spatial information, it is difficult to estimate the fixed number of Gaussians for all bins, even in the static indoor environment. Figure 2 shows histograms of intensity for different bins which have different width, in indoor and outdoor environments. It illustrates that different selection of bin width and different situation of bins will cause the number of Gaussians which are needed to describe the distribution to change. In this section, for consideration of computation, Gaussian merging and deleting rules are used to determine the number of Gaussians, dynamically, instead of other methods, such as reversible jump MCMC [11].

The authors in [15] also used Gaussian merging to improve density estimation. However, they make use of batch learning where all the samples are saved for training. Background modeling is an on-line learning process and all the older samples will be discarded.

When a Gaussian is updated, we will check if it should be merged with others and the merging rules are as follows.

Let two Gaussians which exist in the same model be represented as  $G_1 = (\mu_1, \Sigma_1 = \sigma_1^2 I, \omega_1)$  and  $G_2 = (\mu_2, \Sigma_2 = \sigma_2^2 I, \omega_2)$ , where  $\mu_1, \mu_2$  are the means of corresponding Gaussians,  $\Sigma_1, \Sigma_2$  are covariance matrices,  $\omega_1,$

$\omega_2$  are weights. To merge these two Gaussians, we assume, during training, there are totally  $P$  samples, where  $N$  samples belong to  $G_1$  and  $M$  belong to  $G_2$ . Clearly, the total number of samples for the new Gaussian is  $M + N$ .

The combined mean is:

$$\begin{aligned}\mu_{new} &= \frac{(\sum_{i=1}^N x^i + \sum_{i=1}^M y^i)}{N + M} = \frac{(N/P\mu_1 + M/P\mu_2)}{N/P + M/P} \\ &\approx \frac{1}{\omega_1 + \omega_2}(\omega_1\mu_1 + \omega_2\mu_2)\end{aligned}\quad (10)$$

where  $x^i$  and  $y^i$  are samples from  $G_1$  and  $G_2$ , respectively.

The combined covariance matrix is:

$$\begin{aligned}\Sigma_{new} &= \frac{(\sum_{i=1}^N x^i(x^i)^T + \sum_{i=1}^M y^i(y^i)^T)}{N + M} - \mu_{new}(\mu_{new})^T \\ &\approx \frac{\omega_1\Sigma_1}{\omega_1 + \omega_2} + \frac{\omega_2\Sigma_2}{\omega_1 + \omega_2} + \frac{\omega_1\omega_2(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}{(\omega_1 + \omega_2)^2}\end{aligned}$$

In background modeling, we assume that the covariance matrix is diagonal and that its diagonal components are identical, so

$$\sigma_{new}^2 = \frac{\omega_1\sigma_1^2}{\omega_1 + \omega_2} + \frac{\omega_2\sigma_2^2}{\omega_1 + \omega_2} + \frac{\omega_1\omega_2(\mu_1 - \mu_2)^T(\mu_1 - \mu_2)}{(\omega_1 + \omega_2)^2}$$

is used as approximation, where  $\sigma_{new}^2$  is the diagonal component of  $\Sigma_{new}$ .

The combined weight is:

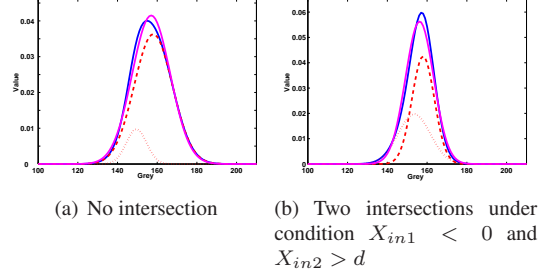
$$\omega_{new} = \frac{N + M}{P} = \frac{N}{P} + \frac{M}{P} \approx \omega_1 + \omega_2 \quad (11)$$

Then, the new Gaussian which is obtained from  $G_1$  and  $G_2$  is denoted as  $G_{new} = (\mu_{new}, \Sigma_{new} = \sigma_{new}^2 I, \omega_{new})$ .

Ideally, merging is performed at a bin when the common area of the two weighted Gaussians divided by any weight exceeds a threshold. However, the computation of common area would be a rather costly procedure. Allowing for diagonal covariance matrix, we construct two one-dimensional weighted Gaussians to represent the originals. These Gaussians are denoted as  $G_{m1} = (0, \sigma_1^2, \omega_1)$  and  $G_{m2} = (d = \|\mu_1 - \mu_2\|, \sigma_2^2, \omega_2)$ , where  $\sigma_1^2$  and  $\sigma_2^2$  are the diagonal components of  $\Sigma_1$  and  $\Sigma_2$ , and  $d = \|\mu_1 - \mu_2\|$  is Euclidian distance of  $\mu_1$  and  $\mu_2$ . As an approximation, the relation between the intersections and the centers of these one-dimensional weighted Gaussians is used to determine whether we should merge the original Gaussians:

1) No intersection. It means that one weighted Gaussian is totally under another, so we merge them.

2) Only one intersection. The intersection point is denoted as  $X_{in}$ . If  $\min(X_{in}/\sigma_1, |d - X_{in}|/\sigma_2) \leq TH_{merge}$ , these two Gaussians will be merged. In our experiments, this one-intersection condition has never happened. This is because some equation should be satisfied strictly.



**Figure 3. Examples of Gaussian merging in background modeling. The dashed red curves are the weighted Gaussians before merging. The blue curves are the Mixed Gaussian distribution before merging. The pink ones show the results after merging.**

3) Two intersections. The intersection points are denoted as  $X_{in1}$  and  $X_{in2}$ , and  $X_{in1} < X_{in2}$ . If  $\min(X_{in1}/\sigma_1, |d - X_{in1}|/\sigma_2) < TH_{merge}$  or if  $\min(X_{in2}/\sigma_1, |d - X_{in2}|/\sigma_2) < TH_{merge}$  or if  $X_{in1} < 0$  and  $X_{in2} > d$  or if  $X_{in1} > d$  or  $X_{in2} < 0$ , merge them. Under other conditions, there is no merging.  $TH_{merge}$  is a threshold for merging and is set to 0.2.

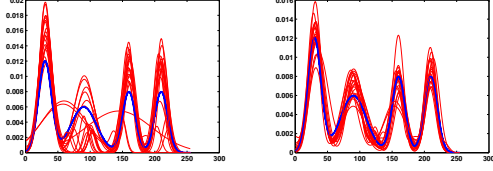
Figure 3 shows some examples of Gaussian merging obtained in the process of background learning. It demonstrates the effectiveness of the approximation.

The Gaussian mixture model which describes the marginal distribution of range component for a bin is updated by the samples belonging to the same bin and is initialized with no Gaussian. If a sample finds no Gaussian to match, then, a new Gaussian centered on the sample with initial weight and variance is added. The updating rules for parameters  $\mu$  and  $\sigma$  are the same as [14]. The prior weight of the  $k^{th}$  Gaussian at time  $t$ ,  $\omega_{k,t}$ , is adjusted as follows

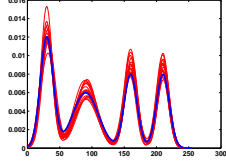
$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(N_{k,t}/M) \quad (12)$$

where  $\alpha$  is the learning rate,  $N_{k,t}$  is the number of samples which match the  $k^{th}$  Gaussian in the corresponding bin, at time  $t$ , and  $M$  is the number of pixels a bin contains in a frame. Therefore, the weights of Gaussians are updated only once despite more samples in a bin. Then, the weights will be checked. If a Gaussian's weight is smaller than a threshold,  $TH_{del}$ , it will be deleted. Also, the weights should be normalized after updating. Figure 4(c) illustrates the results of applying Gaussian merging and deleting rules on a simulated data set with four Gaussians. Figure 4(a) and (b) show the results of traditional GMM with different number of components. Each experiment in Figure 4 is repeated twenty times. After training by the test sequence, the output of the proposed method is better and more stable than GMM. It also illustrates that, to obtain a better re-





(a) Distributions learned by GMM with 4 components. (b) Distributions learned by GMM with 5 components.



(c) Distributions obtained by our method.

**Figure 4. Output distributions by different methods. The blue curves shown in these figures are real distribution.**

sult, the number of Gaussians in GMM should be larger than the real number of components. This is because additional Gaussians are needed to describe outliers.

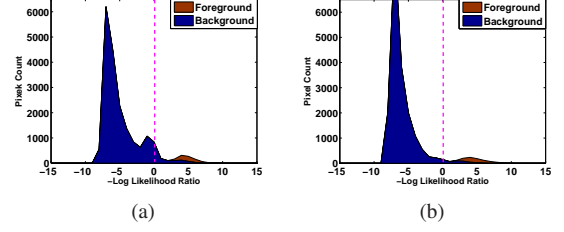
### 2.3 Modeling Foreground with the feedback of tracking

To make use of temporal persistence property of real foreground objects, in this section, the PDF of foreground is constructed based on the assumption that, interesting objects tend to appear in the predicted spatial vicinity which is obtained by a tracker, and tend to maintain consistent colors from frame to frame. This assumption fuses tracking information and is different from that in [13]. It means that, if a bin detects object samples, then the probability of detecting foreground samples with similar colors around another bin where the samples are predicted to appear in the next frame will increase. For considering that, before a bin detects any foreground samples, the probability of observing a foreground pixel of any color is uniform, we model the foreground as

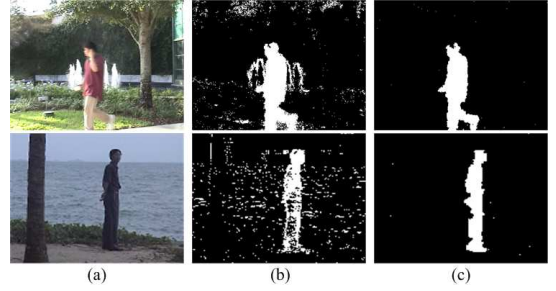
$$f(x|f) = \sum_{j=1}^{BN} c_f K_{Hs}(s - g_j) [\omega_{fj} \gamma + (1 - \omega_{fj}) \psi_j] \quad (13)$$

where  $\omega_{fj}$  is the mixture weight at the  $j^{th}$  bin,  $\gamma$  is a random variable with uniform probability, constant  $c_f$  is a normalization factor which equals  $c_b$ , and  $\psi_j$  is a Gaussian Mixture Model:

$$\psi_j = \sum_{i=1}^{M_j} \omega_{ji} G_{\sigma_{ji}}(c - \mu_{ji}).$$



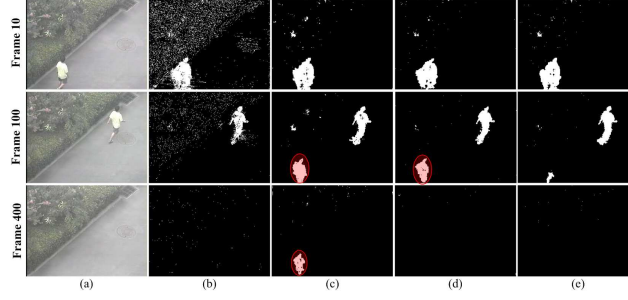
**Figure 6. Histograms of negative log-likelihood ratio values for background and foreground. (a) Histogram based on foreground and background model without tracking information. (b) Histogram obtained with feedback of tracking. The dashed line is the "natural" threshold for the log-likelihood ratio, i.e., zero.**



**Figure 7. Foreground detection results in dynamic scenes. (a) is the original images, (b) shows the results obtained by the traditional GMM, (c) demonstrates the results obtained by the MAX-MRF labeling method based on KGHM.**

Obviously, the part,  $\omega_{fj} \gamma + (1 - \omega_{fj}) \psi_j$  can be seen as the marginal distribution which describes the range component of foreground at  $j^{th}$  bin.  $\omega_{fj}$  is set to 1, at the beginning.

To use tracking information, Kalman filter is used. Every frame, we obtain the speed vector, represented as  $(dx, dy)$ , of a moving object, then, the object's samples which show at the bin whose coordinates are  $(x_b, y_b)$  will update the marginal distribution which describes foreground's range component of another bin where the point situated at  $(x_b + dx, y_b + dy)$  belongs. If we ignore tracking information, the marginal distribution describing the bin at  $(x_b, y_b)$  should be updated. At time  $t$ , if marginal distribution of the  $j^{th}$  bin is updated by the foreground samples, the mixture weight  $\omega_{fj}$  will decrease with a learning rate  $\alpha_w$ , which is set to 0.7, and the foreground samples are used to update the corresponding  $\psi_j$ . The updating rules for  $\psi_j$  are the same as



**Figure 5. Foreground detection by different strategies. (a) original images. (b) detection results by thresholding background model based on KGHM. Under similar foreground detection, the scattered noise is higher. (c) detection results using MAP-MRF estimation without tracking information. Under this condition, a foreground sample updates the marginal distribution at the bin where the sample situates. (d) detection results by [13] which neglects tracking information. (e) results obtained after fusing tracking information based on the KGHM. Obviously, tracking information can improve the segmentation results. In these experiments, the learning rates for background are set to 0.005.**

those for background model. However, the model’s learning rate and the initial weight for newly added Gaussian are much higher than those of background, as the foreground changes far more rapidly than background. To allow stopping objects to become part of background, the minimum of  $\omega_{fj}$  is limited to 0.1 and all samples are used to update the background model, simultaneously. If no updating takes place,  $\omega_{fj}$  will increase with a smaller rate,  $0.1\alpha_w$ , for considering that the interesting object may disappear.

After constructing PDF of foreground, there are several strategies to detect foreground(including shadow) as shown in Figure 5. The fusion of foreground density can increase detection rate of the foreground which persists in time. However, if an object initially in the background moves, this will also make the newly revealed part of background wrongly detected as foreground for a long time as in Figure 5(c) and (d). To alleviate this, if the speed of an object is less than a threshold, no samples of the object will update the foreground model.

The utility in using tracking information can be seen in Figure 5 and 6. Figure 6 shows histogrammed negative likelihood ratio based on KGHM with or without tracking information. It illustrates that, after fusing tracking information, the interclass variance of background reduces and the variance between clusters increases. Figure 7 shows foreground detection results in highly dynamic scenes <sup>1</sup>by the MAP-MRF labeling method [13] based on the models described above. It illustrates that, for the traditional GMM, neglect of spatial information will cause a lot of false positive pixels when dealing with highly dynamic scenes.

<sup>1</sup>These two videos are downloaded from <http://server.cs.ucf.edu/vision/temp/yaser> and, [http://perception.i2r.a-star.edu.sg/bk\\_model/bk\\_index.html](http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html), respectively.

## 2.4 Modeling moving cast shadow

After obtaining PDFs of background and foreground, we can detect foreground objects. However, the detection usually includes shadow which will cause problems, such as object merging and shape distortion. A large part of shadow removal methods is used as an isolated module after foreground detection, such as [2]. Under this condition, the results of shadow removal are highly dependent on the output of foreground detection. The proposed framework can alleviate this problem. In this section, to incorporate shadow removal in a probabilistic framework, we will construct the PDF of shadow.

The construction of the shadow’s PDF is based on the assumption that, for a given bin, the shadow cast by different moving objects is relatively similar [9]. In other words, the shadow value tends to converge to one or more of states in a Gaussian mixture model. Thus, the model of shadow is represented as:

$$f(x|sh) = \sum_{j=1}^{BN} c_{sh} K_{Hs}(s - g_j) \left( \sum_{i=1}^{M_j} \omega_{ji} G_{\sigma_{ji}}(c - \mu_{ji}) \right)$$

where  $f(x|sh)$  is the PDF of shadow and  $c_{sh}$  is a constant which also equals  $c_b$ . The PDF of shadow has the same form as that of background. The part,  $\sum_{i=1}^{M_j} \omega_{ji} G_{\sigma_{ji}}(c - \mu_{ji})$ , is the marginal distribution which describes the range component of shadow at the  $j^{th}$  bin. The samples which can be used to update the model must satisfy [2]:

$$\begin{aligned} 0 < \frac{I_k^V(x, y)}{B_k^V(x, y)} < 1 \\ \wedge (I_k^S(x, y) - B_k^S(x, y)) \leq \tau_S \\ \wedge |I_k^H(x, y) - B_k^H(x, y)| \leq \tau_H \end{aligned} \quad (14)$$

where  $I_k^H(x, y)$ ,  $I_k^S(x, y)$  and  $I_k^V(x, y)$  are the hue, saturation and intensity of a pixel located at  $(x, y)$  of frame  $k$ .  $B_k^H(x, y)$ ,  $B_k^S(x, y)$  and  $B_k^V(x, y)$  are the hue, saturation and intensity of a background pixel at  $(x, y)$ .  $\tau_S, \tau_H$  are parameters which need to be set beforehand. Equation 14 means that the intensity of shadow sample is smaller than the corresponding background and that shadows lower the saturation of points. The equation,  $0 < \frac{I_k^V(x, y)}{B_k^V(x, y)} < 1$ , can be seen as a preclassifier and we can refine it through global and tracking information as in [8], which helps distinguish shadow from dark objects as in Figure 8(b). The updating rules for shadow are the same as those for background.

The rules described above need a background image as reference. However, unlike pixel-wise GMM which can use the mean of the Gaussian with maximum weight as background reference, the KGHM can not obtain background image directly. The background reference is constructed as:

$$B_{t+1}(x, y) = \begin{cases} I_0(x, y) & \text{if } t = 0; \\ (1 - \beta_1)B_t(x, y) + \beta_1 I_t(x, y), & \text{else if } f(x|b) < f(x|f); \\ (1 - \beta_2)B_t(x, y) + \beta_2 I_t(x, y), & \text{otherwise.} \end{cases} \quad (15)$$

where  $B_t(x, y)$  is the background value of pixel  $(x, y)$  at time  $t$ ,  $I_t(x, y)$  is the image value, and  $\beta_1 = 0.0001, \beta_2 = 0.1$ . This formula means that, if a pixel is more prone to be background, a higher learning rate is used to update the corresponding pixel of background image.

## 2.5 Labeling of moving objects and non-moving objects based on MAP-MRF

In real scenes, moving object detection can be seen as a two-category classification problem, moving objects of interest or non-moving objects. Obviously, the non-moving objects should include shadow and background. We approximate the PDFs of them as follows:

$$f(x|m) = f(x|f) \quad (16)$$

$$f(x|nm) = \max(f(x|sh), f(x|b)) \quad (17)$$

where  $f(x|m)$  is the PDF of moving objects and  $f(x|nm)$  the PDF of non-moving objects. Then, an MAP-MRF labeling method is used, as it can enforce spatial context. After deduction as in [13], the MAP-MRF estimate of this classification problem is equivalent to maximize next formula,

$$L = \sum_{i=1}^p \ln\left(\frac{f(x_i|m)}{f(x_i|nm)}\right)l_i + \sum_{i=1}^p \sum_{j=1}^p \lambda(l_i l_j + (1 - l_i)(1 - l_j)) \quad (18)$$

where  $p$  is the number of samples in a frame,  $l_i$  is the label of the  $i^{th}$  sample, moving objects of interest or non-moving objects, and  $\lambda$  is a positive constant. To maximize Equation

18, a graph with a four-neighborhood system is constructed as in [13]. Then, the max-flow algorithm needed for graph cuts computation in [1] is used. After segmentation of moving objects, the detected samples are fed back to update the PDF of moving objects.

## 3 Experimental results

The experiments in this paper include scenes of fountain, ocean surface (as shown in Section 2.2), swaying tree and simulated nominally moving camera. They are carried out on a 3.0 GHZ Intel Pentium 4 processor with 1 GB RAM. The speed is about 6 fps for a frame size of  $320 \times 240$  without optimization. The bin width is set to four in our experiments. The bandwidth matrix  $Hs$  for these models is parameterized as a diagonal matrix with equal variance which is set to 4.

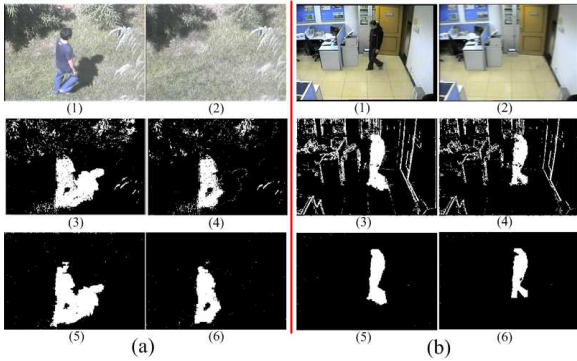
The sequence as in Figure 8(a) shows the scene where the wind caused the trees to move randomly. In this scene, the shadow is also distinctive. This sequence was manually segmented to generate ground truth. It is evident that orderless movement of neighboring "background" will deteriorate the performance of traditional GMM. As shown in (4) and (6) of Figure 8(a), we also can find that the contour of shadow has been removed and that majority of hollows are filled by our method. This shadow contour is caused, because the intensity value changes not abruptly at the edge of the evident shadow. Figure 8(b) shows an indoor scene, where the nominally moving camera was simulated by moving the original pictures left and right(motion distance is about 8 pixels). The shadow here is insignificant. It shows that a slight movement will also cause substantial degradation in performance of GMM, especially on the neighbor of edges. Figure 9 shows the per-frame moving object detection rates according to precision [13] and recall [13] for the video of Figure 8(a). The results in Figure 9 are shown for three different learning rates of traditional GMM. The shadow removal method for GMM is the same as [2]. The detection accuracy is consistently higher than the traditional GMM with shadow removal method in [2].

These figures demonstrate that our method can deal with dynamic scenes and cast shadow effectively after fusing spatial information and incorporating PDF of shadow.

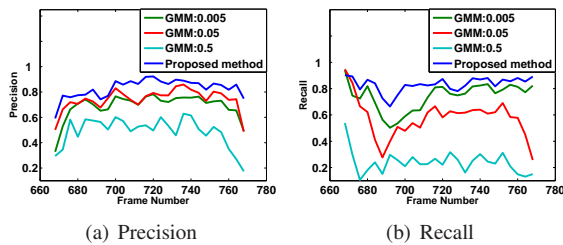
## 4 Conclusions and discussions

In this paper, a probabilistic framework for moving object segmentation in dynamic scenes has been proposed. Under this framework, we unify background modeling, moving object detection and shadow removal by constructing probability density functions(PDFs) of background, shadow and foreground based on KGHM which fuses spatial information. Also, the number of Gaussian components





**Figure 8. Moving object detection in dynamic scenes. (1) original images. (2) background images obtained by Equation 15. (3) results obtained by GMM. (4) the results obtained by GMM and the shadow removal method in [2]. (5) results obtained by MAX-MRF labeling method without PDF of shadow. (6) results of the proposed method.**



**Figure 9. Per-frame detection rates for our method and GMM with shadow removal method in [2]. The learning rates for GMM are 0.5, 0.05 and 0.005, respectively.**

in this model is determined dynamically by Gaussian merging and deleting rules. Furthermore, the feedback of tracking is used to refine the PDF of foreground. Quantitative evaluation and comparison with existing method demonstrate improved performance for moving object detection in dynamic scenes.

Here, the KGHM gives a promising solution to integrate the spatial information (KDE for coordinates) and its corresponding color or texture information (GMM for intensity or texture components) together, which can effectively characterize the texture which has different types of texture(color) details at different local components. Therefore, this comprehensive model is suitable to many applications, such as biometrics and visual surveillance. Our future work will try this model for face recognition.

## Acknowledgement

This work is funded by research grants from the National Basic Research Program of China (2004CB318110), the National Science Foundation (60605014, 60332010, 60335010 and 2004DFA06900), and the CASIA Innovation Fund for Young Scientists. The authors also thank the anonymous reviewers for their valuable comments.

## References

- [1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1222–1239, 2001.
- [2] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with HSV color information. *Proc. IEEE Int'l Conf. Intelligent Transportation Systems*, pages 334–339, 2001.
- [3] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using non-parametric kernel density estimation for visual surveillance. *Proc. IEEE*, 2002.
- [4] P. Hall and M. P. Wand. On the accuracy of binned kernel density estimators. *J. Multivariate Analysis*, 1995.
- [5] T. Kanade, R. Collins, and A. Lipton. Advances in cooperative multi-sensor video surveillance. *Proceedings of DARPA*, 1998.
- [6] K. Karmann, A. Brandt, and R. Gerl. Using adaptive tracking to classify and monitor activities in a site. *Time Varying Image Processing and Moving Object Recognition*, 1990.
- [7] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. on Image Processing*, 11:1459–1472, 2004.
- [8] Z. Liu, K. Huang, T. Tan, and L. Wang. Cast shadow removal combining local and global features. *The Seventh International Workshop on Visual Surveillance*, 2007.
- [9] N. Martel-Brisson and A. Zaccarin. Moving cast shadow detection from a gaussian mixture shadow model. *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2:643–648, 2005.
- [10] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. *ICCV*, pages 1305–1312, 2003.
- [11] S. Richardson and P. Green. On bayesian analysis of mixtures with an unknown number of components. *Proc. Roy. Stat. Soc. B*, 59:731–792, 1997.
- [12] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. *ECCV*, 2000.
- [13] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Trans. Parttern Analysis and Machine Intelligence*, pages 1778–1792, 2005.
- [14] C. Stauffer and W. Grimson. Learning pattern of acitivity using real-time tracking. *IEEE Trans. Parttern Analysis and Machine Intelligence*, 22:747–757, 2000.
- [15] N. Ueda, R. Nakano, Y. Ghahramani, and G. Hinton. SMEM algorithm for mixture models. *Neural Comput*, pages 2109–2128, 2000.