Author manuscript, published in "The Eighth International Workshop on Visual Surveillance - VS2008, Marseille : France (2008)"

inria-00325769, version 1 - 30 Sep 2008

Dynamic Background Subtraction Based on Local Dependency Histogram *

Shengping Zhang, Hongxun Yao, Shaohui Liu
Harbin Institute of Technology
92 West Dazhi Street, Harbin, 150001, China
{spzhang, yhx, shaohl}@vilab.hit.edu.cn

Abstract

Traditional background subtraction methods perform poorly when scenes contain dynamic backgrounds such as waving tree, spouting fountain, illumination changes, camera jitters, etc. In this paper, a novel and effective dynamic background subtraction method is presented with three contributions. First, we present a novel local dependency descriptor, called local dependency histogram (LDH), to effectively model the spatial dependencies between a pixel and its neighboring pixels. The spatial dependencies contain substantial evidence for dynamic background subtraction. Second, based on the proposed LDH, an effective approach to dynamic background subtraction is proposed, in which each pixel is modeled as a group of weighted LDHs. Labeling the pixel as foreground or background is done by comparing the new LDH computed in current frame against its model LDHs. The model LDHs are adaptively updated by the new LDH. Finally, unlike traditional approaches which use a fixed threshold to define whether a pixel matches to its model, an adaptive thresholding technique is also proposed. Experimental results on a diverse set of dynamic scenes validate that the proposed method significantly outperforms traditional methods for dynamic background subtraction.

1. Introduction

Moving objects detection and segmentation from a video sequence is one of the essential tasks in object tracking and video surveillance. A common approach for this task is background subtraction, which first builds an adaptive statistical background model, and then pixels that are unlikely to be generated by this model are labeled as foreground. Despite a large number of background subtraction methods have been proposed in the literature over the past few decades, the task remains challenging when the scenes to be modeled contain dynamic backgrounds such as waving tree, spouting fountain, illumination changes, camera jitters, etc. A robust background subtraction method should work well in these scenes since they are common phenomena in real world. It is always desirable to achieve very high accuracy in the detection of moving objects. The performance of background subtraction depends mainly on the background modeling technique used.

Early approaches operated on the premise that the color of a pixel over time in a static scene could be modeled by a single Gaussian distribution. Wren et al. [15] modeled the color of each pixel with a single three-dimensional Gaussian. Its mean and variance were learned from pixel observations in previous frames. Once the pixel-wise background model was built, the likelihood of each incident pixel generated by this model could be computed. Based on the likelihood, the pixel was labeled as background or foreground. However, a single-Gaussian model is unsuited to most outdoor situations since repetitive object motion, shadows or reflectance often cause multiple pixel colors to belong to the background at each pixel. To overcome the limitations of single-Gaussian model, the mixture of Gaussians (MOG) approach was used to model the complex, non-static scenes [12]. An incident pixel was compared to every Gaussian distribution in the pixel's model and, if a match (defined by a fixed threshold) was found, the mean and variance of the matched Gaussian distribution was updated, otherwise a new Gaussian distribution with the mean equal to the current pixel color and some initial variance was introduced into the mixture. Each pixel was labeled depending on whether the matched distribution represented the background process. Many authors have proposed improvements and extensions to this algorithm. In Ref. [3], new update algorithms for learning mixture models were presented. In Ref. [18], not only the parameters but also the number of components of the mixture was adapted for each pixel.

The Gaussian-based methods have an assumption that the pixel color values over time could be modeled by one

*This work is supported by the National Natural Science Foundation of China (Grant No 60775024) and New Century Excellent Talents in University (NCET-05-03-34).

inria-00325769, version 1 - 30 Sep 2008

or multiple Gaussian distributions, however, this assumption does not always hold in the real world. To deal with the limitations of parametric methods, a nonparametric approach to background modeling was proposed in [1]. The method utilized a general nonparametric kernel density estimation technique for building a statistical representation of the scene. The probability density function for pixel intensity was estimated directly from the observation data without any assumptions about the underling distributions. In [5], a quantization/clustering technique to construct a nonparametric background model was presented. The background was encoded on a pixel by pixel basis and samples at each pixel were clustered into the set of codewords. A similar approach in [16] was proposed, which combined color and texture features by a multi-layer model.

In [4, 9], each pixel was modeled with a Kalman filter. The method can adapt to changes in illumination, but performs poorly in complex dynamic scenes. This approach was used in the automatic traffic monitoring application presented in [6]. In [8], the dynamic background was modeled by an autoregressive moving average (ARMA) model. A robust Kalman filter algorithm was used to iteratively estimate the intrinsic appearance of the dynamic scenes.

Hidden Markov Models (HMMs) have also been used to model pixel intensity [10, 13]. In these approaches, the pixel intensity variations were represented as discrete states corresponding to scene modes. In [10], the approach was used in a traffic monitoring application where the pixel states represented the road, vehicles and shadows. In [13], HMM states corresponded to global intensity modes each modeled with a single Gaussian distribution. The model was capable of handling sudden illumination changes.

Although the methods mentioned above demonstrated their success cases, their performance will notably deteriorate in the presence of dynamic backgrounds such as waving tree, spouting fountain, illumination changes, camera jitters, etc. There are two causes for this. First, these methods model the background model of a pixel exploiting only its intensity information and ignoring the useful dependencies existing in intensities of neighboring pixels. In dynamic scenes, although some pixels significantly changes over time , they should be considered as background. However, the background models of these methods can not effectively model such changes since they exploit the pixel's intensity information lonely. Second, these methods use a fixed threshold to define whether a pixel matches to its model, which also will result in the notable performance deterioration of these methods since the extents of dynamic changes of different pixels are distinctly different. Using a fixed threshold can not model such difference and it will result in the performance deterioration of these methods.

In order to effectively model the dynamic scenes, in this paper, we present a dynamic background subtraction method that has three contributions. First, we present a novel local dependency descriptor, called Local Dependency Histogram (LDH), which is computed over the region centered on a pixel. The LDH effectively extracts the spatial dependency statistics of the center pixel, which contain substantial evidence for labeling the pixel in dynamic scenes. Second, based on the proposed LDH descriptor, we present a novel dynamic background subtraction method, in which each pixel is modeled as a group of weighted LDHs. Labeling the pixel as foreground or background is done by comparing the new LDH computed in current frame against its model LDHs. The model LDHs are adaptively updated by the new LDH. Finally, unlike traditional approaches which define whether a pixel matches to its model using a fixed threshold, an adaptive thresholding technique is also proposed. The proximity value computed using histograms intersection between the LDH and its model LDH over time is modeled by a Gaussian distribution. The match is defined as the proximity value within 2.5 standard deviations of the Gaussian distribution. The Gaussian distribution is adaptively updated using the proximity value.

The rest of the paper is organized as follows: Section 2 describes the proposed local dependency histogram (LDH). The proposed approach to dynamic background subtraction based on LDH is presented in Section 3. The qualitative and quantitative experimental results and analysis are shown in Section 4, followed by some conclusions in Section 5.

2. Local Dependency Histogram

Let R be an $(2N+1) \times (2N+1)$ region. (For simplicity, we assume that the region is square.) The color space is quantized into M levels $C_M = \{0, \ldots, M-1\}$. Let P discretely and regularly index the region lattice, $P = \{(x, y) | -N \leq x \leq N, -N \leq y \leq N\}$. For a pixel $p = (x, y)$, let $C(p) \in C_M$ denote its color level. The center pixel of region R is $\tilde{p} = (0, 0)$. For pixel pair, $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ we define two types of distances: 1) $d(p_i, p_j) = \max\{|x_i - x_j|, |y_i - y_j|\}$, in which the pixel pair can be along any directions, 2) $d'(p_i, p_j) = \max\{|x_i - x_j|, |y_i - y_j|\}$, in which the pixel pair is confined to be only along the horizontal or vertical direction. The given distance set is denoted by $D_L = \{d_0, \ldots, d_{L-1}\}$. For the center pixel $\tilde{p}$, we define L direct neighboring sets as follows:

$$P_l = \{\, p \mid d(p, \tilde{p}) = d_l \,\}, \qquad l = 0, \ldots, L-1\,. \quad (1)$$

We denote the pixels that are not in the L direct neighboring sets but in the region R as indirect neighboring set $\bar{P} = P - \bigcup_{l=0}^{L-1} P_l$. To compute the local dependency histogram, we define two quantities:

$$h_{C(\tilde{p}),l} = \#\{\, p \mid C(p) = C(\tilde{p}), p \in P_l \,\}\,, \quad (2)$$

inria-00325769, version 1 - 30 Sep 2008

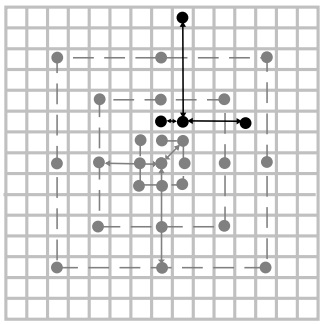


Figure 1. Illustration of spatial dependencies of the center pixel with given distance set $D_3 = \{1, 3, 5\}$. The direct neighboring sets of the center pixel are denoted by gray points. Note that some direct neighboring pixels are denoted by dashed for conveniences. The direct dependencies are denoted by gray arrows which can be along any directions. The indirect neighboring set are denoted by black points and the indirect spatial dependencies are denoted by black arrows which are confined to be only along the horizontal or vertical direction.

$$H_{u,l} = \#\{(p_i, p_j) | C(p_i) = u, C(p_j) = u, d'(p_i, p_j) = d_l, p_i \in \bar{P}, p_j \in \bar{P}\}, \quad (3)$$

where $u = 0, \ldots, M-1, l = 0, \ldots, L-1$ and $\#$ denotes the number of elements of in the set. The $h_{C(\tilde{p}),l}$ is the total number of pixels that are at distance d_l from $\tilde{p}$ along any directions and have the same color with $\tilde{p}$. It models the direct dependencies between $\tilde{p}$ and its direct neighboring pixels. We call this dependency as direct local dependency. The $H_{u,l}$ is the total number of pixel pairs that have the same color u and are d_l apart only along the horizontal or vertical direction in the set $\bar{P}$. It models the indirect dependencies of $\tilde{p}$ between its indirect neighboring pixels. We call this dependency as indirect local dependency. These two dependency statistics can be integrated as:

$$H_{u,l} = \begin{cases} H_{u,l} + h_{C(\tilde{p}),l} & \text{if } u = C(\tilde{p}) \\ H_{u,l} & \text{else}, \end{cases} \quad (4)$$

where $u = 0, \ldots, M-1, l = 0, \ldots, L-1$. Then, the local dependency histogram H is obtained by arranging the matrix $(H_{u,l})_{M \times L}$ as a vector in row-major order. The illustration of the spatial dependencies of the center pixel is shown in Fig. 1.

3. Dynamic background subtraction based on Local Dependency Histogram

In this section, we introduce our approach to dynamic background subtraction based on local dependency histogram (LDH). The algorithm can be divided into three phases, background modeling, background update and foreground detection, respectively described in Section 3.1, 3.3 and 3.4. The proposed adaptive thresholding technique is presented in Section 3.2.

3.1. Background Modeling

There are several advantages of using LDH as statistical descriptor for dynamic background subtraction. First, it explicitly models the spatial dependencies of the center pixel from two aspects: the direct dependencies between the center pixel and its direct neighboring pixels computed by Eq. (2) and the indirect dependencies of the center pixel between its indirect neighboring pixels computed by Eq. (3). These two kinds of dependencies contain substantial evidence for modeling the background model of the center pixel in dynamic scenes. The former models the dynamic motion of the center pixel due to non-periodic motion. For example, the center pixel in previous frame will be a neighboring pixel in current frame. The latter models the dynamic motion occurring in indirect neighboring pixels, which is also very important for labeling the center pixel[2, 7]. For example, the center pixel will not be labeled as background if its indirect neighboring pixels are labeled as foreground. These two kinds of dependencies statistics are then integrated into the histogram statistic, which effectively models the complex dependencies between the pixel and its spatial context. Second, it is not sensitive to noisy since the color space is quantized into less levels. Finally, the computation cost is low, since the pixel pairs in indirect neighboring set are confined to be only along the horizontal or vertical direction.

For a given pixel, let R be an $(2N+1) \times (2N+1)$ square region centered on the pixel. In the following, we describe the background subtraction procedure for this pixel, and the procedure is identical for each pixel. Since we use LDH computed over the region R as the feature vector of the pixel, we can consider the feature vectors of the pixel over time as a histogram process. At time t, the background of the pixel is modeled by a group of adaptive LDHs $\{h_{t,0}, \ldots, h_{t,K-1}\}$, where K is the number of model histograms. The first $B(\leq K)$ model histograms have been identified as representing the background process in last time instant. Each model histogram has a weight between 0 and 1 such that K weights sum to 1. The weight of the kth model histogram is denoted by $\omega_{t,k}$. It denotes the probability that this model histogram belongs to the background

inria-00325769, version 1 - 30 Sep 2008

process. Initially, the K model histograms are assigned by the LDH computed at $t = 1$ with weight $1/K$.

3.2. An Adaptive Thresholding Method

The LDH of the pixel computed at current frame $t + 1$, is denoted by h. It is compared against all K model histograms using a proximity measure. The histogram intersection is used to measure the proximity value of two LDHs as follows:

$$\rho(h_1,\, h_2) = \frac{\sum_{i=0}^{M-1} \min(h_{1,i}, h_{2,i})}{\max(\sum_{j=0}^{M-1} h_{1,j}, \sum_{j=0}^{M-1} h_{2,j})}\,, \quad (5)$$

where h_1 and h_2 are two LDHs, and M is the number of histogram bins. The denominator is the maximal value of the sums of all elements in two histograms. It is used to normalize the histogram intersection since the sums of all elements in two different LDHs are different.

K proximity values $\rho_{t,0}, \ldots, \rho_{t,K-1}$ between h and K model histograms are obtained by Eq. (5). Unlike traditional approaches which define whether the pixel matches to its model using a fixed threshold, an adaptive thresholding method is also proposed. We consider the K proximity values over time as K proximity values processes and each such process is modeled by a Gaussian distribution $\rho_{t,k} \sim N(\mu_{t,k}, \sigma^2_{t,k}), k = 0, \ldots, K-1$. Initially, for each Gaussian distribution the mean μ is assigned by the proximity value computed at time $t = 2$ and the σ^2 is assigned by a low initial value σ^2_{init}. The histogram h matching to the model histogram $h_{t,k}$ is defined as the proximity value $\rho_{t,k}$ within 2.5 standard deviations of the kth Gaussian distribution:

$$\frac{(\rho_{t,k} - \mu_{t,k})^2}{\sigma^2_{t,k}} < 2.5^2\,. \quad (6)$$

3.3. Background Update

The background update consists of three aspects: updating the K model histograms, updating the K Gaussian distributions and updating the background process.

If none of the K model histograms matches to h using Eq. (6), the model histogram with the lowest weight is replaced with h and assigned a low initial weight. In our experiments, a value of 0.01 was used. The other model histograms and the K Gaussian distributions keep invariant.

If a model histogram $h_{t,k}$ matches to h, the corresponding Gaussian distribution of the model histogram is updated as follows:

$$\mu_{t+1,k} = (1-\gamma)\mu_{t,k} + \gamma\rho_{t,k}\,, \quad (7)$$

$$\sigma^2_{t+1,k} = (1-\gamma)\sigma^2_{t,k} + \gamma(\rho_{t,k} - \mu_{t,k})^2\,, \quad (8)$$

where

$$\gamma = \alpha_g \frac{1}{\sqrt{2\pi}\sigma_{t,k}} e^{-\frac{(\rho_{t,k}-\mu_{t,k})^2}{2\sigma^2_{t,k}}}\,, \quad (9)$$

α_g is the Gaussian learning rate that controls the update speed of the Gaussian distributions.

In all matched model histograms, the best matching model histogram $h_{t,k}$ is selected with the highest proximity value. The $h_{t,k}$ is adapted with the new data by updating its bins as follow:

$$h_{t+1,k} = \alpha_b h + (1-\alpha_b)h_{t,k}\,, \quad (10)$$

where α_b is a background learning rate. The weights of all K model histograms are updated as follows:

$$\omega_{t+1,k} = \alpha_w M_{t,k} + (1-\alpha_w)\omega_{t,k}, \qquad (k = 0, \ldots, K-1)\,, \quad (11)$$

where α_w is a weight learning rate and $M_{t,k}$ is 1 for the best matching model histogram and 0 for the others. The adaptive speed of the background model is controlled by the learning rate α_b and α_w.

All of the model histograms are not necessarily produced by the background process. The weight of the model histogram is used to decide whether the model histogram models the background process or not. All of the model histograms are sorted in decreasing order according to their weights, and the first B model histograms are selected as the background histograms as follows:

$$B = \arg\min_b \Big(\sum_{k=0}^{b-1} \omega_k > T_B\Big), \qquad T_B \in [0, 1]\,, \quad (12)$$

where T_B is a threshold for measuring the minimum portion of the data that should be accounted for by the background.

3.4. Foreground Detection

It should be noted that the B background histograms identified using Eq. 12 in Section 3.3 will be used in next time instant. In current time instant, foreground detection is done before updating the background model. It uses the current B background histograms which were identified in the update step at last time instant. The LDH h is compared against the current B background histograms using the same match definition as in the update algorithm. If the match is found for at least one background histogram, the pixel is labeled as background, otherwise, it is labeled as foreground.

4. Experimental Results and Analysis

In order to confirm the effectiveness of the proposed method for dynamic scenes, we conduct experiments on a

inria-00325769, version 1 - 30 Sep 2008

Figure 2. Experimental results on waving tree sequence. Morphological operators were not used in the results. The top row are the original images: 10^{th}, 246^{th}, 248^{th}, 252^{th} and 254^{th} frames. The second row are the results obtained by Mixture of Gaussians method (MOG). The third row are the results obtained by the proposed method, and the fourth row are the masked original images.

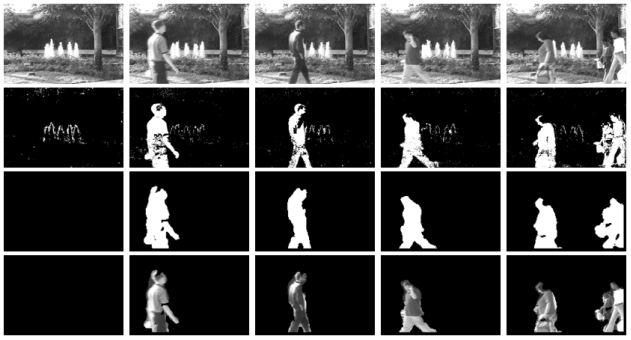


Figure 3. Experimental results on fountain sequence. The top row are the original frames: 10^{th}, 597^{th}, 877^{th}, 1113^{th} and 1816^{th} frames. The second row are the results obtained by Mixture of Gaussians method (MOG). The third row are the results obtained by the proposed method, and the fourth row are the masked original images.

variety of sequences presented in previous literature. The widely used Mixture of Gaussian method (MOG) is used to compare with the proposed method. The comparison is done based on qualitative evaluation by looking at processed images provided by the algorithms, and quantitative evaluation in terms of true positive ratio and false positive ratio.

4.1. Qualitative Evaluation

Qualitative results on four sequences of dynamic scenes are presented in this section. The first sequence is the waving tree sequence presented in [14]. The tree branches heavily waves in the presence of a strong wind. Fig. 2 shows the comparative results. The first row are the original images, the second row shows the detected foreground by MOG. The detected results by the proposed method are shown in the third row. The fourth row are the masked original images. It is stressed that no morphological operators or median filters were used in the presentation of these results. As shown in Fig. 2, since the dynamic motions caused by waving tree do not repeat exactly, it causes substantial performance degradation of the MOG, which detected a large number of background pixels as foreground and also labeled a huge amount of foreground pixels as background on the inner areas of the moving person. The proposed method dramatically outperforms the MOG, and achieves very high accuracy in the detection of moving person.

Fig. 3 shows some results on fountain sequence from [11], which involves three sources of dynamic motion: 1) The spouting fountain, 2) the swaying tree branches above, and 3) the shadow of the trees branches on the grass below. The MOG performs poorly with labeling a large number of background pixels as foreground due to the dynamic motions, especially the spouting fountain in the scene. The proposed method manages the situation relatively well and suppresses most false detection by the MOG.

In Fig. 4, results on car sequence presented in [17] are shown. This is a very difficult scene from the background modeling point of view since it involves fast moving car, heavily swaying vegetation and large area of illumination changes. The MOG outputs a huge amount of background pixels as foreground. Although the proposed method also misses some foreground pixels, the overall performance of the proposed method is better than the performance of the MOG.

Fig. 5 shows results on moving camera sequence from [11], which involved a camera mounted on a tall tripod. The wind caused the tripod to sway back and forth causing motion of the camera. The second row shows the detected foreground by MOG, and it is evident that the motion of the camera causes substantial degradation in performance, despite a five-component mixture model and a relatively high learning rate of 0.05. The proposed method detects the moving car and person accurately.

As shown in the first columns of Fig. 2–Fig. 5, the MOG performs poorly and labels a huge number of background pixels as foreground at the beginning of the sequences which do not include foreground objects. On the other hand, the proposed method handles dynamic motions

inria-00325769, version 1 - 30 Sep 2008

Table 1. The parameter values of the proposed method for four test sequences.

Parameter	N	M	D_L	K	α_w	α_b	α_g	T_B	σ^2_{init}
Value	9	64	$\{1,3,5\}$	4	0.01	0.01	0.01	0.8	0.025

immediately and achieves accurate detection at the beginning of the sequences. There are two causes for this: First, the proposed method uses LDH as statistical features which effectively models the spatial dependencies of neighboring pixels. The spatial dependencies provide the substantial evidence for labeling the center pixel and they are exploited to sustain high levels of detection accuracy at the beginning of the sequences. However the MOG exploiting only single pixel's color can not accurately detect the foreground objects at the beginning of the sequences since the MOG need to take longer time to train the background models than the proposed method. Second, the MOG defines whether the pixel match to its model using a fixed threshold, which is unsuited to dynamic scenes since the extents of dynamic motions of different pixels are distinctly different. A fixed threshold can not effectively model such difference and it will cause performance degradation at the beginning of the sequences. The proposed method uses an adaptive thresholding technique, which can achieve accurate detection even if at the beginning of the sequences.

The parameter values of the proposed method are given in Table. 1. We did not change the parameter values for four test sequences, although better results could be obtained by customizing the values for each sequence.

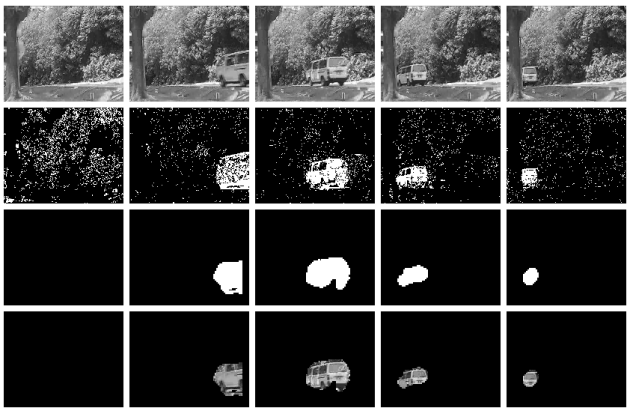


Figure 4. Experimental results on car sequence. The top row are the original frames: 10^{th}, 202^{th}, 208^{th}, 216^{th} and 220^{th} frames. The second row are the results obtained by Mixture of Gaussians method (MOG). The third row are the results obtained by the proposed method, and the fourth row are the masked original images.

4.2. Quantitative Evaluation

The performance of the proposed method is also evaluated quantitatively in terms of true positive ratio (TPR) and false positive ratio (FPR):

$$\text{TPR} = \frac{\text{true positives}}{\text{number of foreground pixels in ground truth}}, \tag{13}$$

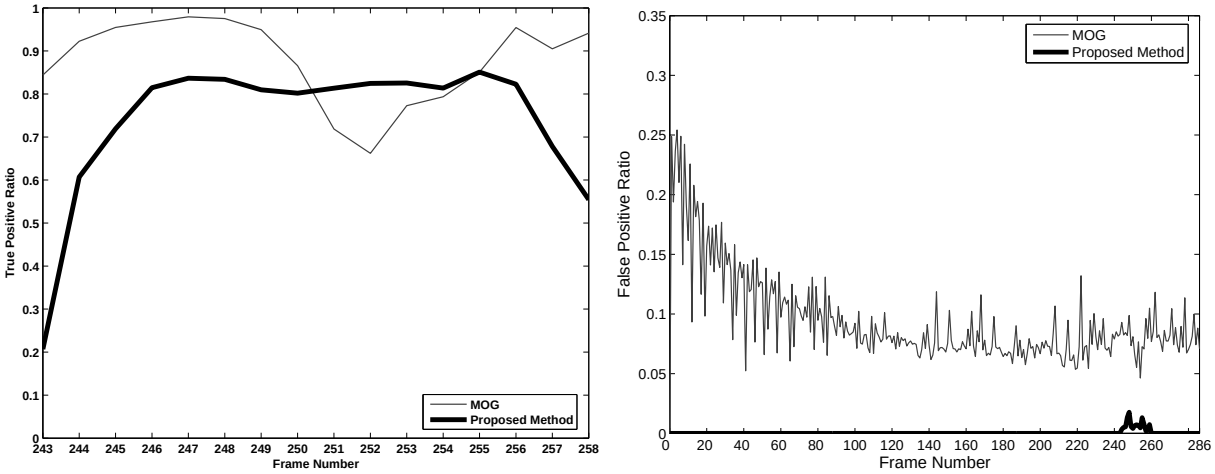
$$\text{FPR} = \frac{\text{false positives}}{\text{number of background pixels in ground truth}}, \tag{14}$$

where true positives are the number of foreground pixels that are correctly detected, false positives are the number of background pixels that are detected as foreground, and ground truth is the correct detection result and was obtained by manual segmentation.

We first perform quantitative evaluation on waving tree sequence, which has total 287 frames and the person moves across the field of view during 243^{th} to 258^{th} frames. The TPR and FPR are shown in Fig. 6. As shown in Fig. 6(b), the proposed method gives lower false positives ratio than the MOG for all 287 frames. Especially for the first 242 frames where there are no moving objects, the proposed method outputs zero false positives and is notable superior to the MOG. In the case of true positive ratio, as shown in Fig. 6(a), compared to the MOG the proposed method performs better for 251^{th}–255^{th} frames. For the rest of the 11 frames, the proposed method is inferior to the MOG. It should be noticed that, for the proposed method, most of the missing foreground pixels occur on the contour areas of the moving objects (as seen in Fig. 2). This is because spatial dependency features are extracted from the pixel neighboring. In most applications, the accurate contour information is not need. According to the overall performance shown in Fig. 6, the proposed method outperforms the MOG.

In the moving camera sequence, the scene is empty for the first 276 frames, after which two objects (first a person and then a car) move across the field of view. The sequence contained an average camera motion of approximately 14.66 pixels[11]. Fig. 7 shows its quantitative evaluation results. As shown in Fig. 7(a), the proposed method achieves the higher true positive ratio than the MOG. In the case of false positive ratio (shown in Fig. 7(b)), the proposed method outperforms the MOG for the first 276

inria-00325769, version 1 - 30 Sep 2008



(a) (b)

Figure 6. Quantitative evaluation results on waving tree sequence. (a) True positive ratio and (b) false positive ratio.

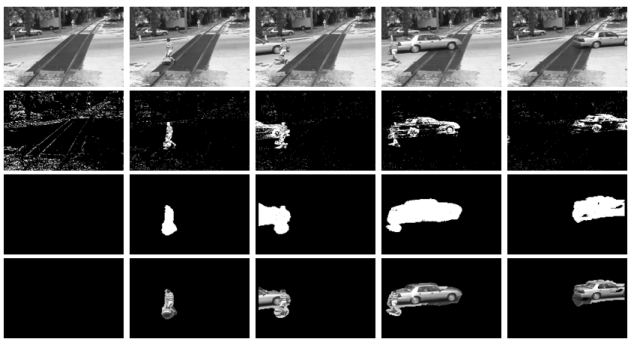


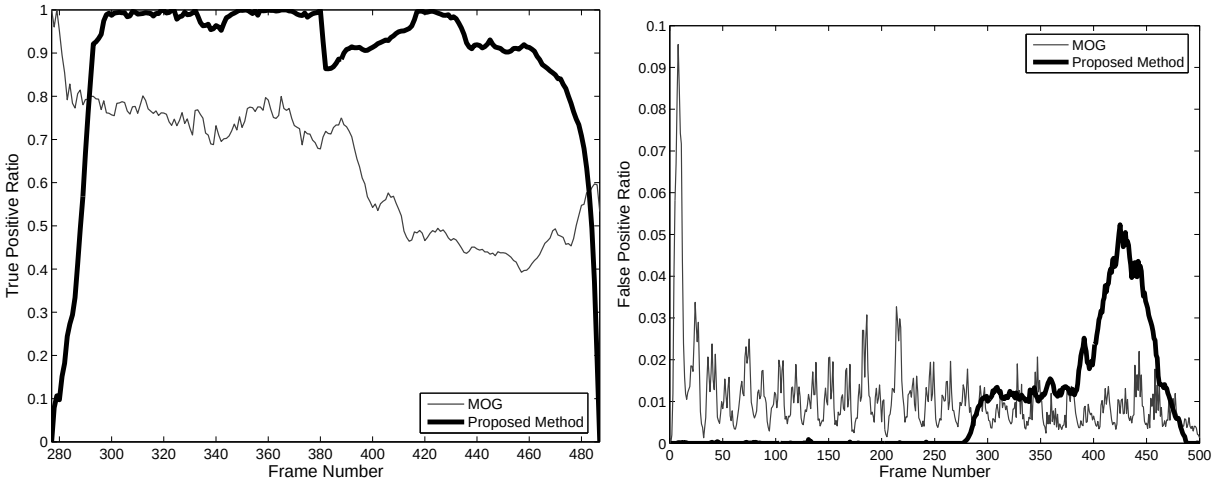
Figure 5. Experimental results on moving camera sequence. The top row are the original frames: 10^{th}, 379^{th}, 396^{th}, 421^{th} and 452^{th} frames. The second row are the results obtained by Mixture of Gaussians method (MOG). The third row are the results obtained by the proposed method, and the fourth row are the masked original images.

frames and outputs zero false positives. When the moving objects move across the scene after 276^{th} frame, the proposed method outputs higher false positive ratio than the MOG. This is because that the proposed method labels a large number background pixels as foreground on the contour areas of the moving objects (see Fig. 5). The overall performance of the proposed method is superior to the MOG.

We also measured the speed of the proposed method on waving tree sequence. The image resolution is 160×120 pixels. We used a standard PC with a 3.0 GHz processor and 1 GB of memory in our experiments. The proposed method achieved a frame rate of 15 fps. This makes the method well-suited to applications that require real-time processing.

5. Conclusions

In this paper, we present a dynamic background subtraction method that has three contributions. First, we present a novel local dependency descriptor, called Local Dependency Histogram (LDH), which is computed over the region centered on a pixel. The LDH effectively extracts the spatial dependency statistics of the center pixel which contain substantial evidence for labeling the pixel in dynamic scenes. Second, based on the proposed LDH descriptor, we present a novel dynamic background subtraction method, in which each pixel is modeled as a group of weighted LDHs. Labeling the pixel as foreground or background is done by comparing the new LDH computed in current frame against its model LDHs. The model LDHs are adaptively updated by the new LDH. Finally, unlike traditional approaches which define whether a pixel matches to its model using a fixed threshold, an adaptive thresholding technique is also proposed. The proximity value computed using histograms intersection between the LDH and its model LDH over time is modeled by a Gaussian distribution. The match is defined as the proximity value within 2.5 standard deviations of the Gaussian distribution. The Gaussian distribution is adaptively updated using the proximity value. Experimental results on a variety of dynamic scenes validate that the



(a) (b)

Figure 7. Quantitative evaluation results on moving camera sequence. (a) True positive ratio and (b) false positive ratio.

inria-00325769, version 1 - 30 Sep 2008

proposed method significantly outperforms the widely used Mixture of Gaussian (MOG).

References

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, July 2002.

[2] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[3] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Proc. European Workshop Advanced Video Based Surveillance Systems*, 2001.

[4] K. Karmann and A. Brandt. Moving object recognition using an adaptive background memory. *Cappellini V. (ed.) Time-Varying Image Processing and Moving Object Recognition*, 1990.

[5] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. *Proc. Int. Conf. Image Processing*, pages 3061–3064, 2004.

[6] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. *Proc. Int. Conf. Pattern Recognition*, pages 126–131, 1994.

[7] S. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995.

[8] M. Mason and Z. Duric. Using histogram to detect and tracking objects in color video. *Proc. Applied Imagery Pattern Recognition Workshop*, pages 154–159, 2001.

[9] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. *Proc. Int. Conf. Recent Advances in Mechatronics*, pages 193–199, 1995.

[10] J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. *Proc. European. Conf. Computer Vision*, pages 336–350, 2000.

[11] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, November 2005.

[12] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[13] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Buhmann. Topology free hidden markov models: Application to background modeling. *Proc. Int. Conf. Computer Vision*, pages 294–301, 2001.

[14] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. *Proc. Int. Conf. Computer Vision*, 1:255–261, 1999.

[15] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[16] J. Yao and J. Odobez. Multi-layer background subtraction based on color and texture. *Proc. Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[17] W. Zhang, X. Z. Fang, X. K. Yang, and Q. M. J. Wu. Spatiotemporal gaussian mixture model to detect moving ojbects in dynamic scenes. *Jouranl of Electronic Imaging*, 16(2):023013–1–6, November 2007.

[18] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. *Proc. Int. Conf. Pattern Recognition*, pages 28–31, 2004.