Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 782101, 14 pages
doi:10.1155/2010/782101

Research Article

Local Histogram of Figure/Ground Segmentations for Dynamic Background Subtraction

Bineng Zhong,[1] Hongxun Yao,[1] Shaohui Liu,[1] and Xiaotong Yuan[2]

[1] *Department of Computer Science and Engineering, Harbin Institute of Technology, No.92, West Da-Zhi Street, Harbin, Heilongjiang 150001, China*
[2] *National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing 100080, China*

Correspondence should be addressed to Bineng Zhong, bnzhong@gmail.com

Received 23 October 2009; Revised 22 April 2010; Accepted 9 June 2010

Academic Editor: Irene Y. H. Gu

We propose a novel feature, local histogram of figure/ground segmentations, for robust and efficient background subtraction (BGS) in dynamic scenes (e.g., waving trees, ripples in water, illumination changes, camera jitters, etc.). We represent each pixel as a local histogram of figure/ground segmentations, which aims at combining several candidate solutions that are produced by simple BGS algorithms to get a more reliable and robust feature for BGS. The background model of each pixel is constructed as a group of weighted adaptive local histograms of figure/ground segmentations, which describe the structure properties of the surrounding region. This is a natural fusion because multiple complementary BGS algorithms can be used to build background models for scenes. Moreover, the correlation of image variations at neighboring pixels is explicitly utilized to achieve robust detection performance since neighboring pixels tend to be similarly affected by environmental effects (e.g., dynamic scenes). Experimental results demonstrate the robustness and effectiveness of the proposed method by comparing with four representatives of the state of the art in BGS.

1. Introduction

Background subtraction (BGS) has attracted significant attention due to its wide variety of applications, including intelligence video surveillance, human machine interfaces, and robotics. Much progress has been made in the last two decades. However, designing robust BGS methods is still an open issue, especially considering various complicated variations that may occur in dynamic scenes, for example, trees waving, water rippling, moving shadow, illumination changes, camera jitters, and so forth. To solve them, most top-performing methods rely on more sophisticated features, more elaborate modeling techniques, prior information on the scenes and foreground objects, more costly post processing schemes (e.g., Graph Cuts on Markov Random Field), and more higher-level feedbacks (e.g., detection or tracking). In the literatures, for a scene, we actually can get a lot of output via a number of BGS algorithms using different features and modeling strategies. Since each kind of BGS algorithm has its strength and weakness and is particularly applicable for handling a certain type of variation, many methods often use sequential coarse-to-fine frameworks to fuse the output of a number of BGS algorithms. However, when a foreground pixel is not detected by coarse level subtraction due to some reasons, for example, similar color, those methods will not classify this pixel as foreground. The following question naturally arises: instead of using sequential coarse-to-fine fusion frameworks, is there another more powerful way for fusing the output of a number of BGS algorithms to achieve more robust BGS results in dynamic scenes? Our answer is yes.

In this paper, we propose an approach that uses local histogram of figure/ground segmentations to fuse a set of candidate solutions that are produced by simple BGS algorithms in order to get a final robust and accurate BGS result, especially under dynamic scenes. More specifically, for one incoming video frame, we first obtain a set of candidate figure/ground segmentations via fast and simple BGS algorithms. Then, we represent each pixel in the video frame as a local histogram of figure/ground segmentations via combing

these proposal solutions. Finally, the background model of each pixel is constructed as a group of weighted adaptive local histograms of figure/ground segmentations, which capture apparent coocrurence statistics of neighboring pixels.

Our method has the following advantages. (1) We can use multiple complementary BGS algorithms to build background models for a scene. This avoids the pitfalls of purely single BGS approaches. (2) The proposed feature, local histogram of figure/ground segmentations, fuses the output of a number of BGS algorithms to encode spatial correlation between neighboring pixels. This avoids a basic assumption sharing by most BGS algorithms: there exists a common underlying low-level visual property (e.g., intensities, colors, edges, gradients, textures, and optical flow) which is shared by the consecutive pixels in the same position, and can thus be extracted and compared to the background model. This assumption, however, may be too restrictive, especially under difficult conditions such as dynamic scenes. The proposed method does not require the temporal continuity of the background images, but the correlation of image variations at neighboring pixels. Therefore, we can robustly detect foreground objects in dynamic scenes, as illustrated by our results in Section 5.

The rest of the paper is organized as follows. Section 2 reviews related work in the BGS literature. The local histogram of figure/ground segmentations is then described in Section 3. The BGS approach based on local histogram of figure/ground segmentations is presented in Section 4. Experimental results are given in Section 5. Finally, we conclude this work in Section 6.

2. Related Work

One popular technique is to model each pixel color in a video frame with a Gaussian distribution [1]. This model does not work well in the case of dynamic scenes. To deal with this problem, Gaussian Mixture Model (GMM) [2] is used to model each pixel. But it cannot adapt to the case where the background has quick variations [3]. Numerous improvements of the original method developed by Stauffer and Grimson [2] have been proposed over the recent years and a good survey of these improvements is presented in [4].

Rather than extending the GMM, a number of nonparametric approaches have been proposed to model background distribution in complex environments where background statistics cannot be described parametrically. In W4 system [5], the background scene is statically modeled by the minimum and maximum intensity values and maximal temporal derivative for each pixel recorded over some period. A nonstatistical clustering technique to construct a background model is presented in [6]. The background is encoded on a pixel-by-pixel basis and samples at each pixel are clustered into the set of codewords. Elgammal et al. [7] are among the first to utilize the kernel density estimation (KDE) technique to model the background color distribution, which has been successful applied in BGS literature. Another significant contribution of this work is the incorporation of spatial constraints into the formulation of foreground classification. In the second phase of their approach, pixel values that could be explained away by distributions of neighboring pixels are reclassified as background, allowing for greater resilience against dynamic backgrounds. In [8], the background and foreground models are first constructed via KDE technique separately, which are then used competitively in a MAP-MRF decision framework. Mittal and Paragios [9] propose the use of variable bandwidths for KDE to enable modeling of arbitrary shapes of the underlying density in a more natural way. Parag and Elgammal [10] use a boosting method (RealBoost) to choose the best feature to distinguish the foreground for each of the areas in the scene. However, one key problem with kernel density estimation techniques is their high computational requirement due to the large number of samples needed to model the background. A Bayesian framework that incorporates spectral, spatial, and temporal features to characterize the background appearance is proposed in [11]. Under this framework, the background is represented by the most significant and frequent features, that is, the principal features, at each pixel.

Some authors model the background using texture features. Heikkilä and Pietikäinen [12] propose an approach based on the discriminative LBP histogram. However, simple grayscale operations make LBP rather sensitive to noise and it is also not so efficient on uniform regions. Yao and Odobez [13] propose a multiple layer background model which makes use of the LBP texture feature and color feature. In [14], the background is firstly divided into three types of regions—flat, sketchable and textured region according to a primal sketch representation. Then, the three types of regions are modeled, respectively, by Mixture of Gaussians, image primitives and LBP histograms. Finally, the geometry information obtained from camera calibrations is used to further reduce false alarms.

Some approaches treat pixel value changes as a time series and consider a predictive model to capture the most important variation based on past observations. In [15, 16], an autoregressive model is proposed to capture the properties of dynamic scenes. Monnett et al. [17] model the background as a dynamic texture, where the first few principal components of the variance of a set of background images comprise an autoregressive model. In [18], a Hidden Markov Model approach is adopted.

A number of attempts have been made to utilize statistics of neighborhoods for BGS. Seki et al. [19] propose a BGS method based on the cooccurrence of image variations, which can be regarded as narrowing the background image variations by estimating the background image pattern in each image block from the neighboring image patterns in the input image. In [20], scene is coarsely represented as the union of pixel layers and foreground objects are detected by propagating these layers using a maximum-likelihood assignment. However, the limitations of the method are high-computational complexity and the requirement of an extra offline training step. Ko et al. [21] have developed a BGS scheme that analyzes the temporal variation of intensity or color distributions, instead of either looking at temporal variation of point statistics, or the spatial

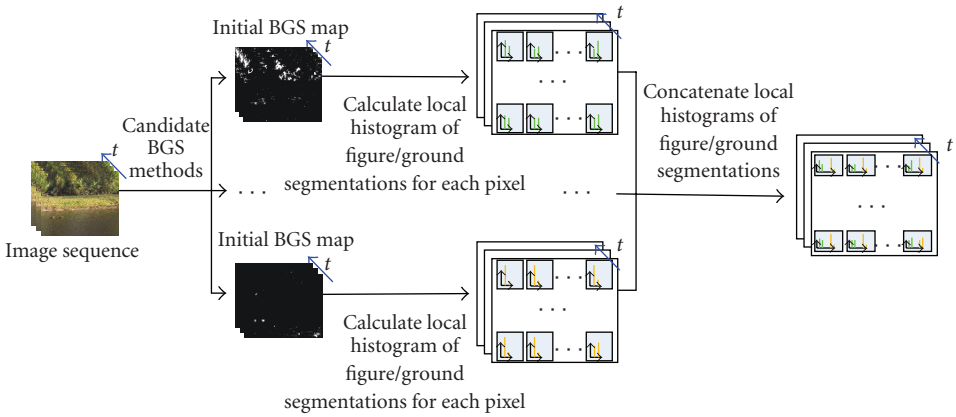

Figure 1: The process of constructing local histogram of figure/ground segmentations to form a final representation for each pixel. In the figure, t denotes a frame number.

variation of region statistics in isolation. Dalley et al. [22] introduce a new image generation model that takes into account the spatial uncertainty of dynamic background textures. In their model, they allow pixels to be generated from nearby Gaussians. Mahadevan and Vasconcelos [23] view BGS as a problem of saliency detection: background points are those considered not salient by suitable comparison of object and background appearance and dynamics. Other methods (e.g., [24]) firstly use BGS to get a set of candidate foreground pixels; then, use foreground analysis to remove false alarm pixels of detected foreground regions.

Other efforts dealing with background modeling include motion-based approach [25], region-based algorithms [26, 27], hierarchical method [28], and methods using edge features [27, 29]. Cevher et al. [30] present a method to directly recover background subtracted images using the compressive sensing theory when the objects of interest occupy a small portion of the camera view, that is, when they are sparse in the spatial domain.

3. The Proposed Feature

In this section, we describe the proposed feature, local histogram of figure/ground segmentations, whose goal is to combine several candidate solutions that are produced by simple BGS algorithms to get a more reliable and robust feature for BGS. Figure 1 illustrates the procedure for representing each image pixel as a local histogram of figure/ground segmentations. For one incoming video frame, we first obtain a set of candidate BGS maps via several BGS algorithms. Then, for each initial BGS map, we calculate a local histogram of figure/ground segmentations computed on a neighboring region centered on each pixel. Finally, these local histograms of figure/ground segmentations of each initial BGS map are concatenated together to form a final representation for each pixel.

Below we give a detailed description about each component in this feature extraction framework.

3.1. Initial Figure/Ground Segmentations. In this paper, to instantiate the proposed feature, we incorporate the KDE-based method of [7] and the LBP-based method of [12] to get initial figure/ground segmentations. Due to the incorporation of spatial constraints into the formulation of foreground classification, the KDE-based method [7] can effectively adapt to smooth behaviors and gradual variations in the background. However, there are still some problems which lead to poor performance when infrequent motions occur, such as ripples or trees rustling periodically (but not constantly) due to wind gusts (please see Figure 3). Furthermore, when a foreground pixel is not detected by pixel level subtraction due to similar color, the method will not classify this pixel as foreground. Instead of using only the pixel color or intensity information to make the decision, Heikkila and Pietikainen [12] have utilized discriminative texture features (LBP histogram) in dealing with the BGS problem. The texture-based method often gives good results for textured regions but is useless in textureless regions. Moreover, simple grayscale operations make LBP rather sensitive to noise and it is also not so efficient on uniform regions.

The motivation of embedding both KDE-based method (using color feature) and LBP-based method (using texture feature) in our feature extraction framework for BGS is to fuse multiple complementary features. Since each kind of feature has an interesting property, which is particularly applicable for handling a certain type of variation, we want to efficiently exploit it in order to make more reliable the final fusion procedure. For instance, texture features may be considered for obtaining invariance to textured regions, while they might not be very suitable for textureless regions. On the other hand, color information should overcome texture feature's limitation.

3.2. Local Histogram of Figure/Ground Segmentations. The main goal of using local histogram of figure/ground segmentations is to make up for deficiencies in each individual BGS algorithm, thus achieving a better overall BGS performance than each single BGS algorithm could provide.

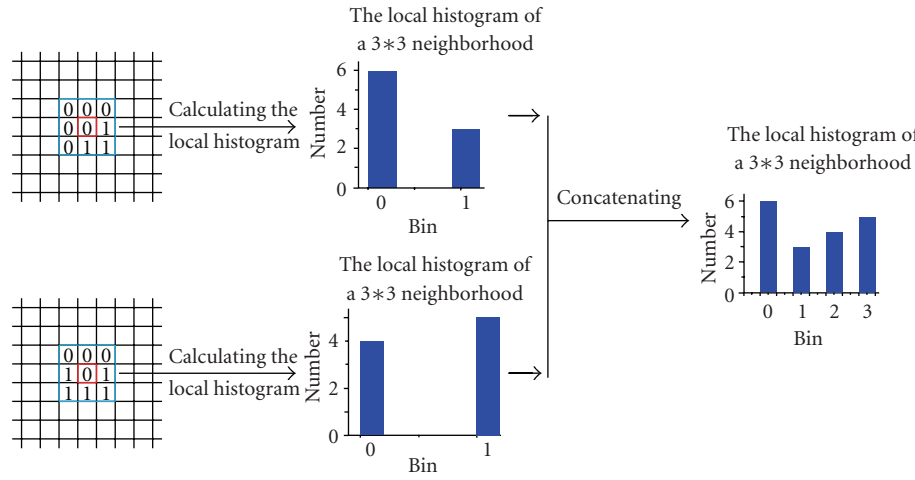

Figure 2: One simple example of constructing local histogram of figure/ground segmentations to form a final representation for a pixel using a 3∗3 neighborhood.

Based on the initial figure/ground segmentations, we represent a pixel in a video frame as the feature of local histogram of figure/ground segmentations via the following steps. First, for each initial BGS map, we calculate a local histogram of figure/ground segmentations computed over a squared fixed-size $N \times N$ neighborhood around each pixel. For efficient calculation, integral histogram [31] is used here. Then, these local histograms of the figure/ground labels of each initial BGS map are concatenated together to form the final representation of each pixel. Specifically, let S denote the number of initial BGS maps. The preliminary feature extraction step thus yields to S (2-bin) histograms. The S histograms are then concatenated together to form a final 2S-bin histogram, which is then normalized to sum to one, so that it is also a probability distribution. Figure 2 shows the procedure for representing an image pixel as a local histogram of figure/ground segmentations using a 3∗3 neighborhood.

To the best of the authors' knowledge, none of the earlier studies have utilized discriminative local histogram of figure/ground segmentations in dealing with the BGS problem. Only some hierarchical coarse-to-fine strategies may have been considered. In this paper, we propose an approach that uses discriminative local histogram of figure/ground segmentations to capture background statistics.

4. Background Subtraction (BGS)

In this section, we introduce background modeling mechanism based on local histograms of figure/ground segmentations described above. The goal is to construct and maintain a statistical representation of the scene that the camera sees.

We consider the local histograms of figure/ground segmentations of a particular pixel over time as a pixel process, and model the background for this pixel as a group of weighted adaptive local histograms of figure/ground segmentations, $\{H_{1,t}, H_{2,t}, \ldots, H_{n,t}\}$, where n is the number of model histograms and t is the current frame number. Each model histogram has a weight between 0 and 1 and all the n weights sum up to one. The weight of the iTh model histogram is denoted by $w_{i,t}$.

At the initialized stage, each bin of the n local histograms is set as 0. The weight of each histogram is set as $1/n$. Then, the BGS procedure continues in the following iterative fashion until the end of video:

(i) foreground detection.

(ii) background updating.

Below we give some detailed descriptions about these two components, and the whole BGS algorithm is summarized finally.

Foreground Detection. At the beginning phase of detection, we sort the model histograms in decreasing order according to their weights and the first B model histograms are chosen as the background model:

$$B = \arg\min_{b} \left(\sum_{i=1}^{b} w_{i,t} > T_n \right), \tag{1}$$

where T_n is a measure of the minimum portion of the data that should be accounted for by the background. Actually, the weight $w_{i,t}$ of each model histogram encodes the accumulation of supporting evidence for the background distributions. We are interested in the local histograms which have the most supporting evidence over the time. Equation (1) takes the "best" histograms until a certain portion T_n of the recent data has been accounted for. An incoming histogram V of the given pixel is checked against the existing n model histograms until a match is found. In our paper, the similarity between two histograms V_1 and V_2 is calculated by the Bhattacharya distance:

$$D_B(V_1, V_2) = \sum_{i=1}^{K} \sqrt{V_{1i} V_{2i}}, \tag{2}$$

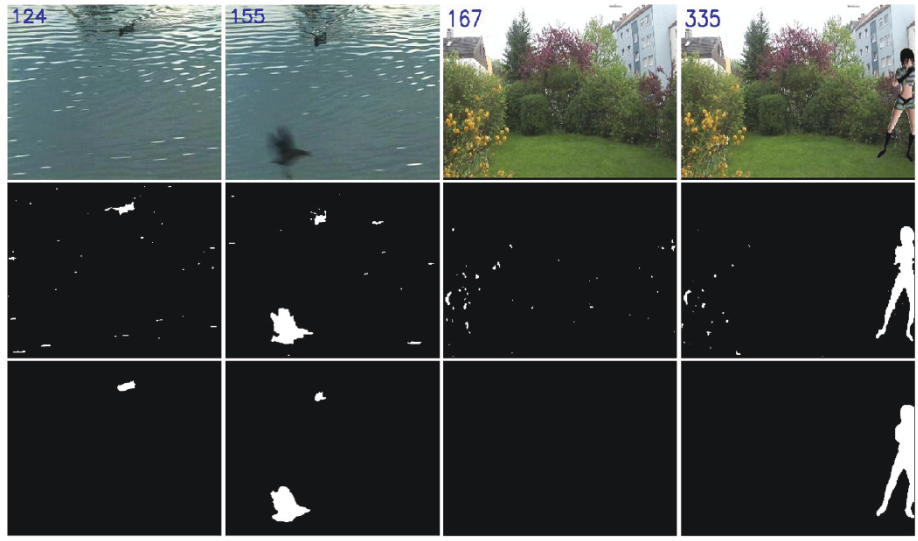

Figure 3: Comparison results of the KDE algorithm and its variation using local histogram of figure/ground segmentations on the two dynamic scenes. The first two columns are from a scene contains ripples in the water. The last two columns are from a scene contains heavily swaying trees. The first row contains original video frame. The second and third row contains the detection results of the KDE algorithm and its variation, separately.

Initialization:
(1) Initialize candidate BGS algorithms.
(2) Initialize the local histograms, their corresponding weights and the rest parameters.
for $t = 2$ **to the end of the video**
(1) Generate a set of proposal BGS map (i.e., proposal solutions) via a heterogeneous set of candidate BGS algorithms.
(2) Construct local histogram of figure/ground segmentations for each candidate BGS algorithms.
(3) Concatenate local histograms of figure/ground segmentation from candidate BGS algorithms to form a final representation for each pixel.
(4) Detect foreground based on the concatenated histograms of figure/ground segmentations.
(5) Update background model of each candidate BGS methods.
(6) Update background model of the concatenated histograms of figure/ground segmentations.
end for

Algorithm 1: Local histogram of figure/ground segmentations for dynamic background subtraction.

where K is the number of histogram bins. Please note that the larger the $D_B(V_1, V_2)$, the higher the probability of matching. Other similarity measures like $L2$ distance, Chi square distance or log-likelihood statistic could also be used. If the similarity is larger than a threshold T_s for at least one background model, the pixel is classified as background. Otherwise, the pixel is labeled as foreground.

Background Updating. In the background updating phase, if none of the n model histograms match the current histogram V, the model histogram with lowest weight is replaced with the current histogram V and a low prior weight β. In our experiments, a value of $\beta = 0.05$ is used and a match is defined as the similarity above a threshold T_s. The weights of the n model histograms at time $t + 1$ are adjusted with the new data as follows:

$$w_{i,t+1} = (1 - \alpha)w_{i,t} + \alpha M_{i,t+1}, \tag{3}$$

where α is the learning rate and $M_{i,t+1}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized. The bigger the weight, the higher the probability of being a background histogram. The adaptation speed of the background model is controlled by the learning rate parameterα. The bigger the learning rate, the faster the adaptation is. The unmatched model histograms remain the same. The model histogram which matches the new observation is updated as follows:

$$H_{i,t+1} = (1 - \alpha)H_{i,t} + \alpha V. \tag{4}$$

Finally, a summary of our local histogram of figure/ground segmentations-based BGS algorithm is described as Algorithm 1.

Table 1: The parameter values of the five BGS algorithms.

Method	Parameter Values
GMM [2]	$K = 5$ (the number of Gaussian components), $T = 0.8$ (the minimum portion of the background model), $\alpha = 0.01$ (learning rate), and $f = 2.5$ (a match is defined as a pixel value within f standard deviations of a distribution).
LBP [12]	$\text{LBP}_{P,R} = \text{LBP}_{6,2}$ (P equally spaced pixels on a circle of radius R), $R_{\text{region}} = 9$ (defines the region for histogram calculation), $K = 5$ (the number of LBP histograms), $T_B = 0.8$ (the minimum portion of the background model), $T_P = 0.65$ (the threshold for the proximity measure), $\alpha_b = 0.01$ (learning rate for updating histogram), and $\alpha_w = 0.01$ (learning rate for updating weight).
KDE [7]	$N = 100$ (the number of samples for each pixel), $W = 50$ (time window for sampling), $T = 10e - 8$ (the probability threshold for a pixel to be a foreground), $\alpha = 0.3$ (the parameter determining the relative pixel values considered as shadowed), SDEstimationFlag = 1 (estimate suitable kernel bandwidth to each pixel), and UseColorRatiosFlag = 1 (use normalized RGB for color).
Bayesian Model [8]	$R * G * B * X * Y = 26 * 26 * 26 * 21 * 31$ (the number of bins used to approximate the background/foreground model), $T = -5$ (the log-likelihood ratio threshold), and $\alpha = 0.01$ (learning rate).
Ours	$n = 3$ (the number of model histograms for each pixel), $T_n = 0.7$ (the minimum portion of the background model), $T_s = 0.65$ (the histogram similarity threshold), $\alpha = 0.01$ (learning rate), and $N \times N = 9 \times 9$ (the size of local squared region).

5. Experiments

Our algorithm is implemented using C++, on a computer with Intel-Pentium Dual 2.00 GHz processor. The running time of the whole BGS algorithm is determined by the slowest candidate BGS method and the fusion time since all candidate BGS methods are run in parallel. It achieves the processing speed of 10 fps at the resolution of 160×120 pixels (the running time could be reduced substantially using multiple cores). For performance evaluation, we compare our approach against four representatives of the current state of the art in BGS—the widely used Gaussian mixture model of [2], the texture-based method of [12], the nonparametric kernel density estimator of [7], and the Bayesian model of Sheikh and Shah [8]. In the rest of our experiments, we refer to the four compared algorithms as GMM, LBP, KDE, Bayesian Model separately. We have tested the five BGS algorithms using dozens of challenging video sequences from the existing literatures as well as our own collections. Both qualitative and quantitative comparisons are done to evaluate the five BGS algorithms.

For comparisons, we acknowledge the fact that the BGS results of the four compared BGS algorithms may be worse than is reported in the corresponding papers; this could be because our parameter values were not tuned per each video sequence. However, our study is still valid for comparison due to the following reasons. First, we used the typical parameter values given in the papers. Second, only the fusion algorithm is different between our method and the compared methods (e.g., LBP and KDE), and everything else is kept constant. This allows us to isolate the BGS methods, which provide initial figure/ground segmentations, to make sure that it is the cause of the performance difference. In our experiments, the significant parameter values of the five algorithms are listed in Table 1. Please refer to [2, 7, 8, 12] for more details about these methods. It must be emphasized that, after the construction of the background models, one can use any postprocessing scheme (e.g., morphological processing or Graph Cut smooth) to give more fine-tuned results. Thus, the informativeness of the results could be obscured by the varying abilities of these operations. Therefore, when comparing these algorithms, we use only the morphological operations.

5.1. Efficacy of Local Histogram of Figure/Ground Segmentations for Background Modeling. In this subsection, we illustrate the efficacy of local histogram of figure/ground segmentations for dynamic background modeling by using two image sequences including ripples in the water [34] and heavily swaying trees [35], as shown in Figure 3.

We compare the performance of the KDE algorithm with its alternative modification using the local histogram of figure/ground segmentations. More specifically, the variation of the KDE algorithm takes the initial figure/ground segmentations obtained by the KDE algorithm as input to a BGS algorithm, in which each pixel's background model is constructed as a group of weighted adaptive 2-bin local histograms of figure/ground segmentations. As is clearly shown in Figure 3, the KDE algorithm generates many false detections due to the quick variations in the dynamic scenes. On the other hand, by making use of local histogram of figure/ground segmentations, the variation of the KDE algorithm has almost no false detections in spite of the dynamic motions.

To more clearly illustrate why local histogram of figure/ground segmentations contains substantial evidence for dynamic BGS, we plot in Figure 4(b) the evolving curves of the intensity values, background probabilities obtained by KDE, initial figure/ground labels obtained by KDE, the distances between the current local histogram of figure/ground segmentations and the corresponding background histogram (obtained by the modeling procedure described in Section 4), and the final labels obtained

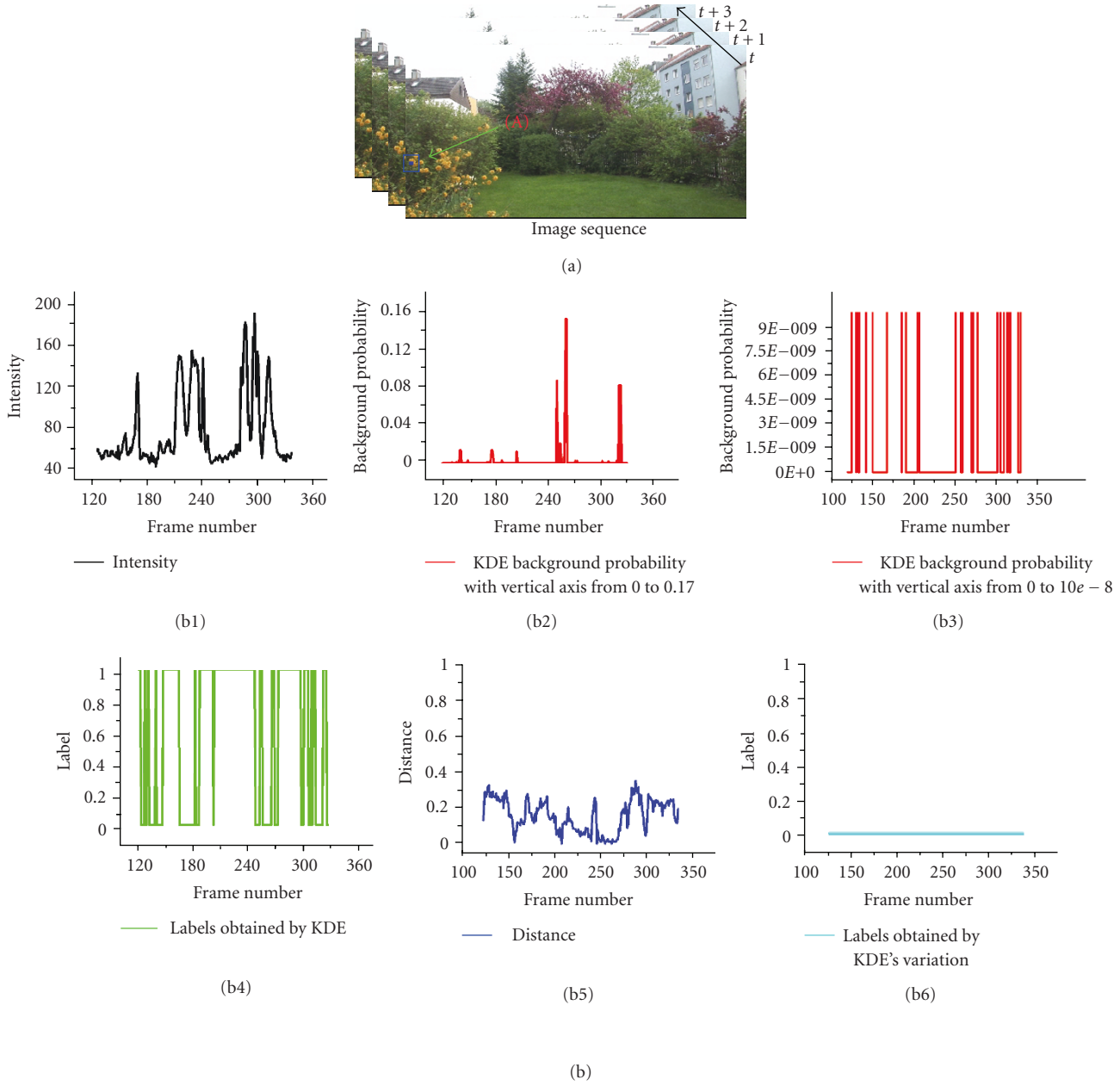

FIGURE 4: Evolving curves of the intensity values, background probabilities obtained by KDE, initial figure/ground labels obtained by KDE, the distances between the current local histogram of figure/ground segmentations and the corresponding background histogram, and the final labels obtained by KDE's variation for a dynamic pixel A. (a) Outdoor scene with blue rectangle showing the location of the sample pixel A. (b) The corresponding curves mentioned above, for the dynamic pixel A, respectively.

by KDE's variation separately, for the pixel A shown in Figure 4(a). In Figure 4(b5), it is obvious to see that the distance obtained by our method is no larger than 0.35, that is, the fluctuation of the distance distribution is relatively compact and small. Thus, comparing to KDE (please see Figures 4(b2) and 4(b3)), this property significantly eases the selection of parameter values for judging a pixel is foreground or not. Therefore, we can get more robust and stable results as shown in Figure 4(b6). On the other hand, for the other four curves mentioned above, relatively high fluctuation of the corresponding value appears due to the quick variations and the nonperiodic motions of the dynamic pixels. In particular, let us take a look at the KDE background probability curve (Figure 4(b3)) with vertical axis from 0 to $10e-8$ to further explain the label fluctuation phenomenon caused by KDE. It is well known that setting the probability threshold for a pixel to be a foreground is a tradeoff between sensitiveness and accuracy, that is, the smaller the value the less false positive and more false negative. In this case, the label curve (Figure 4(b4)) obtained by KDE still drastically and frequently changes, even though the probability threshold is set as small as $10e-8$.

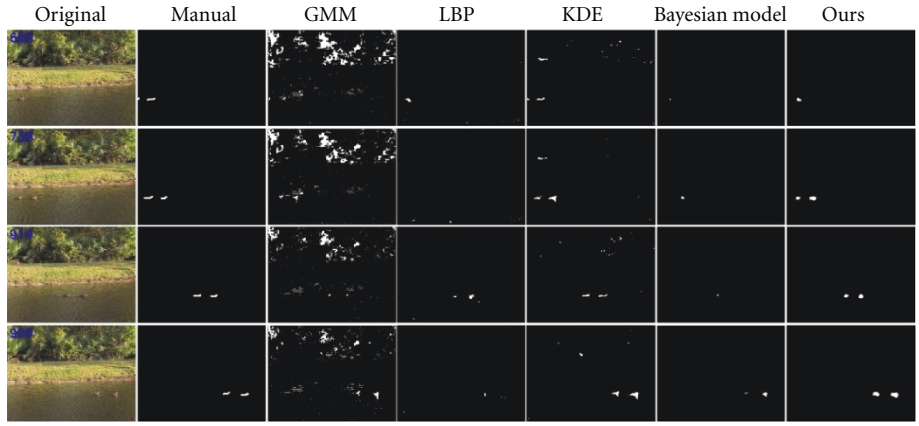

Figure 5: Qualitative comparison results of the five BGS algorithms on the Ducks sequence. The first column contains the original video frames. The second column contains the corresponding ground truth frames. The last five columns contain the detection results of the GMM, LBP, KDE, Bayesian Model and our method, respectively.

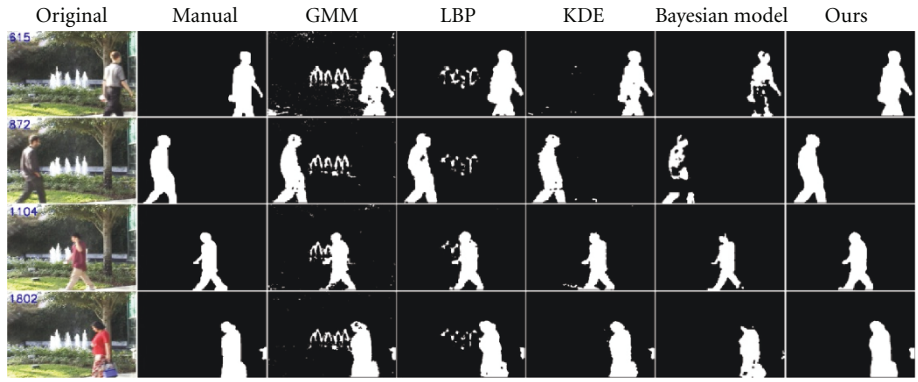

Figure 6: Qualitative comparison results of the five BGS algorithms on the Fountain sequence. The first column contains the original video frames. The second column contains the corresponding ground truth frames. The last five columns contain the detection results of the GMM, LBP, KDE, Bayesian Model and our method, respectively.

Based on the above analysis, the efficacy of local histogram of figure/ground segmentations for background modeling is verified.

5.2. Qualitative Comparisons. Qualitative comparison results of the five BGS algorithms on several challenging video sequences of dynamic scenes are presented in this subsection.

Figure 5 shows qualitative comparison results of the five BGS algorithms on the *Ducks* sequence from [36]. The *Ducks* sequence is from an outdoor scene that contains two ducks swimming on a pond, with dynamic background composed of subtle illumination variations along with ripples in the water and heavily swaying trees. This is a very difficult scene from the background modeling point of view. The upper part of the scene contains heavily swaying trees. This leads to the failure of classical background modeling methods that rely only on the pixel color information (i.e., GMM). Since some simple statistics of neighborhoods have been considered in KDE, the results obtained by KDE have greatly improved. However, there are still some false foreground pixels under this difficulty condition, due to the quick variations and the nonperiodic motions of the waving trees. The challenges in the lower part of the scene are that the background is composed of subtle illumination variations along with ripples in the water and the color of the ducks and background is similar. The LBP performs well on the upper part of the scene but generate some false background and foreground pixels in the lower part of the scene, which is textureless. The reason is that simple grayscale operations make LBP rather sensitive to noise, even using the modified version of LBP with the thresholding constant α' set to 3, as suggested by the original study. Since Bayesian Model constructs the entire background/ foreground model with a single adaptive binned kernel density estimation using quantized feature space (i.e., $R * G * B * X * Y$), it generates some false background pixels in the lower part of the scene where the color of the ducks and background is similar. Our method gives good results because it explicitly considers the meaningful correlation between pixels in the spatial vicinity and uses multiple complementary features to build background models for scenes.

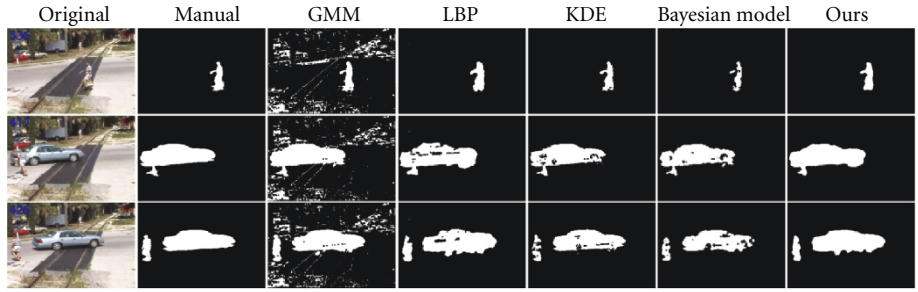

Figure 7: Qualitative comparison results of the five BGS algorithms on the Camera Jitter sequence. The first column contains the original video frames. The second column contains the corresponding ground truth frames. The last five columns contain the detection results of the GMM, LBP, KDE, Bayesian Model, and our method, respectively.

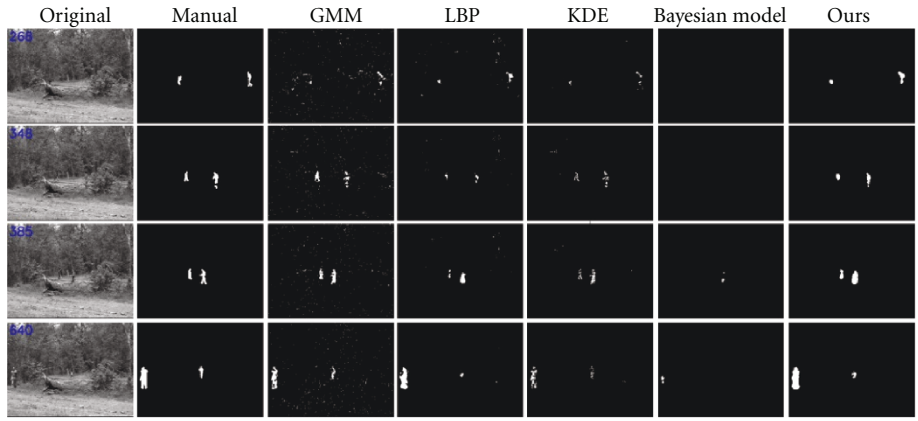

Figure 8: Qualitative comparison results of the five BGS algorithms on the Jump sequence. The first column contains the original video frames. The second column contains the corresponding ground truth frames. The last five columns contain the detection results of the GMM, LBP, KDE, Bayesian Model, and our method, respectively.

Figure 6 shows qualitative comparison results of the five BGS algorithms on the *Fountain* sequence from [36]. The *Fountain* sequence contains three sources of dynamic motion: (1) the fountain, (2) the tree branches oscillate, and (3) the shadow of the trees branches on the grass below. It is obviously to see that, for GMM and LBP, most of the false foreground pixels occur on the background areas occupied by the fountain. GMM generates large number of false foreground pixels due to the nonperiodic motions of the fountain. The reason for the failure of LBP is that it does not work very robustly on flat image areas such as fountain, where the gray values of the neighboring pixels are very close to the value of the center pixel. KDE generates some false foreground pixels on shadow areas. Bayesian Model generates some false background pixels on image areas where the color of foreground and background is similar. It can be seen that our method has almost no false detections in spite of the dynamic motions.

Figure 7 shows qualitative comparison results of the five BGS algorithms on the *Camera Jitter* sequence from [36]. The *Camera Jitter* sequence contains average camera jitter of about 14.66 pixels. Since the nominal motion of the camera do not repeat exactly, GMM handles this difficulty condition poorly, that is, producing many false foreground pixels. While the rest methods manage the situation relatively well, due to considering the meaningful correlation between pixels in the spatial vicinity.

Figure 8 shows qualitative comparison results of the five BGS algorithms on the *Jump* sequence from [37]. The challenges in the *Jump* sequence are that the background is composed of waving trees and the color of the two moving persons and background is similar. This is a very difficult scene from the background modeling point of view. Benefitting from fusion the output of a number of complementary BGS algorithms, the proposed method performs much more robust than the other four, while the GMM, LBP or KDE generates some false detections under this difficulty condition, due to the quick variations of the waving trees. It also can be seen that the Bayesian Model produces some false background pixels on image areas where the color of foreground and background is similar.

In Figure 9, we show the results of our method using other four dynamic outdoor sequences. The first sequence is from Wallflower [32] which contains heavily swaying

Figure 9: Some detection results by our method. The first row contains the original video frames. The second row contains the corresponding detection results.

trees. The other three dynamic outdoor sequences are from our own collections, which include large-area waving leaves and ripples in the water. The challenges in these four dynamic scenes are that the backgrounds are continuously changing and have quick variations. Our method successfully handles these situations and the moving objects are detected correctly.

5.3. Quantitative Comparisons. In order to provide a quantitative perspective about the quality of foreground detection with our approach, we manually mark the foreground regions in every frame from the *Ducks*, *Fountain*, *Camera Jitter* and *Jump* sequence to generate ground truth data, and make comparison between the five BGS algorithms. In the most BGS work, quantitative evaluation is usually done in terms of the number of false negatives (the number of foreground pixels that were missed) and false positives (the number of background pixels that were marked as foreground). However, it is found that when averaging the measures over various environments, they are not accurate enough. In this paper, a new similarity measure presented by Li et al. [11] is used to evaluate the detection results of foreground objects. Let A be a detected foreground region and B be the corresponding ground truth, the similarity between A and B is defined as

$$S(A,B) = \frac{A \bigcap B}{A \bigcup B}, \tag{5}$$

$S(A,B)$ varies between 1 and 0 according to their similarity. If A and B are the same, $S(A,B)$ approaches 1, otherwise 0 if A and B have the least similarity. It integrates the false positive and false negative in one measure.

The corresponding quantitative comparison is reported in Figure 10. For the *Ducks* and *Jump* Sequence, our method outperforms the comparison methods. In the case of *Fountain* and *Camera Jitter* sequence, our method is comparable to KDE and it outperforms the rest algorithms. It should be noticed that, for our method, most of the false detections occur on the contour areas of the foreground objects. This is because the meaningful correlation between pixels in the spatial vicinity is exploited. That is why the performance of KDE and our method is comparable in the *Fountain* and *Camera Jitter* sequences, in which the objects of interest occupy a large portion of the camera view. In these two sequences, the number of errors of our method caused by contour inaccuracy may be more than that of KDE caused by dynamic scenes at some video frames. According to the overall results, the proposed method outperforms the comparison methods for the used test sequences in most cases. The reason for the superior performance is that our algorithm is able to handle dynamic scenes via local histogram of figure/ground segmentations. Rather than relying only one BGS algorithm and taking the risk that that algorithm is suboptimal for handling every type of variations, our local histogram of figure/ground segmentations-based approach fuses the output of a number of complementary BGS algorithms and reaps the advantage of encoding spatial correlation between neighboring pixels.

5.4. Sensitivity to Parameters. Since our method has relatively many parameters, there naturally arise the following questions. (1) How sensitive our method is to small changes of its parameter values? (2) How easy or difficult is it to obtain a good set of parameter values? To answer these questions, we calculate the similarity measures for different parameter configuration. Because of a huge amount of different combinations, only one parameter is varied at a time. The measurements are made for several image sequences. The results for the *Fountain* sequence of Figure 6 are plotted in Figure 11, in which the final similarity measure is achieved by averaging the similarity measures obtained from all frames. Obviously, for all parameters, a good value can be chosen across a wide range of values. The same observation is identical for all the test sequences. This property significantly eases the selection of parameter values. Furthermore, the experiments have shown that a good set of parameters for a sequence usually performs well also for other sequences (please see Figures 5–9).

5.5. When Does the Overall Approach Break Down? Finally, one requirement of our algorithm is that there must exist at least one BGS algorithm that produces an accurate enough suggestion in a particular region, thus one would naturally

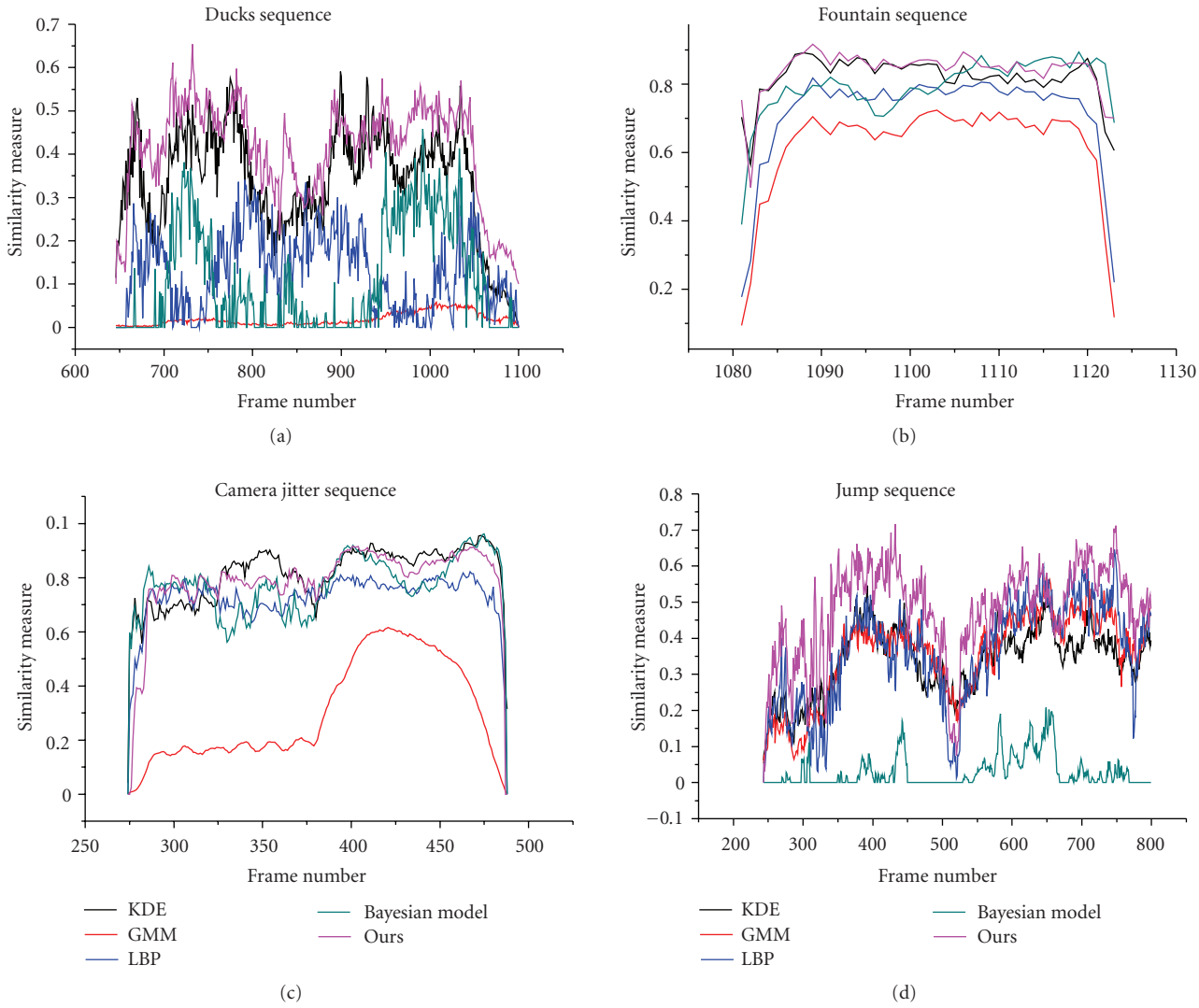

Figure 10: Quantitative comparison results of the five BGS algorithms on the Ducks, Fountain, Camera Jitter and Jump sequence. The similarity measure used to evaluate the detection results of foreground objects is defined in (5).

miss an experimental exposition of the limitations of the approach. In what scenario the overall approach breaks down? What happens if all complementary BGS algorithms fail?

To explore the performance of the proposed approach under these conditions, we perform a set of experiments in which the challenges included in the testing sequences are cast shadow and light Switch. Figure 12 shows some failed BGS examples of our methods. Like most of the other methods, our method is not capable of handling the cast shadow and light switch problem. This is because the two complementary BGS algorithms (i.e., the KDE method of [7] and the LBP method of [12]) embedding in our method are both unable to handle these cases. To address these cases, one potentially useful idea that is not investigated in the paper, is to utilize some higher level processing that could be used to detect sudden changes in background.

6. Conclusion

We explore in this paper the method of fusing a set of candidate solutions that are produced by simple BGS algorithms in order to get a final robust and accurate BGS result, especially under dynamic scenes. Our study shows that the local histogram of figure/ground segmentations is helpful to achieve robust moving objects detection in dynamic scenes such as waving trees, ripples in water, illumination changes, camera jitters. The key contribution of this work is to formulate the fusion of a set of candidate solutions inside a novel feature extraction framework. To instantiate this framework, we incorporate the KDE method of [7] and the LBP method of [12] to get initial figure/ground segmentations. The background model of each pixel is constructed as a group of weighted adaptive local histograms of figure/ground segmentations, which describe the structure properties of the surrounding region. Therefore,

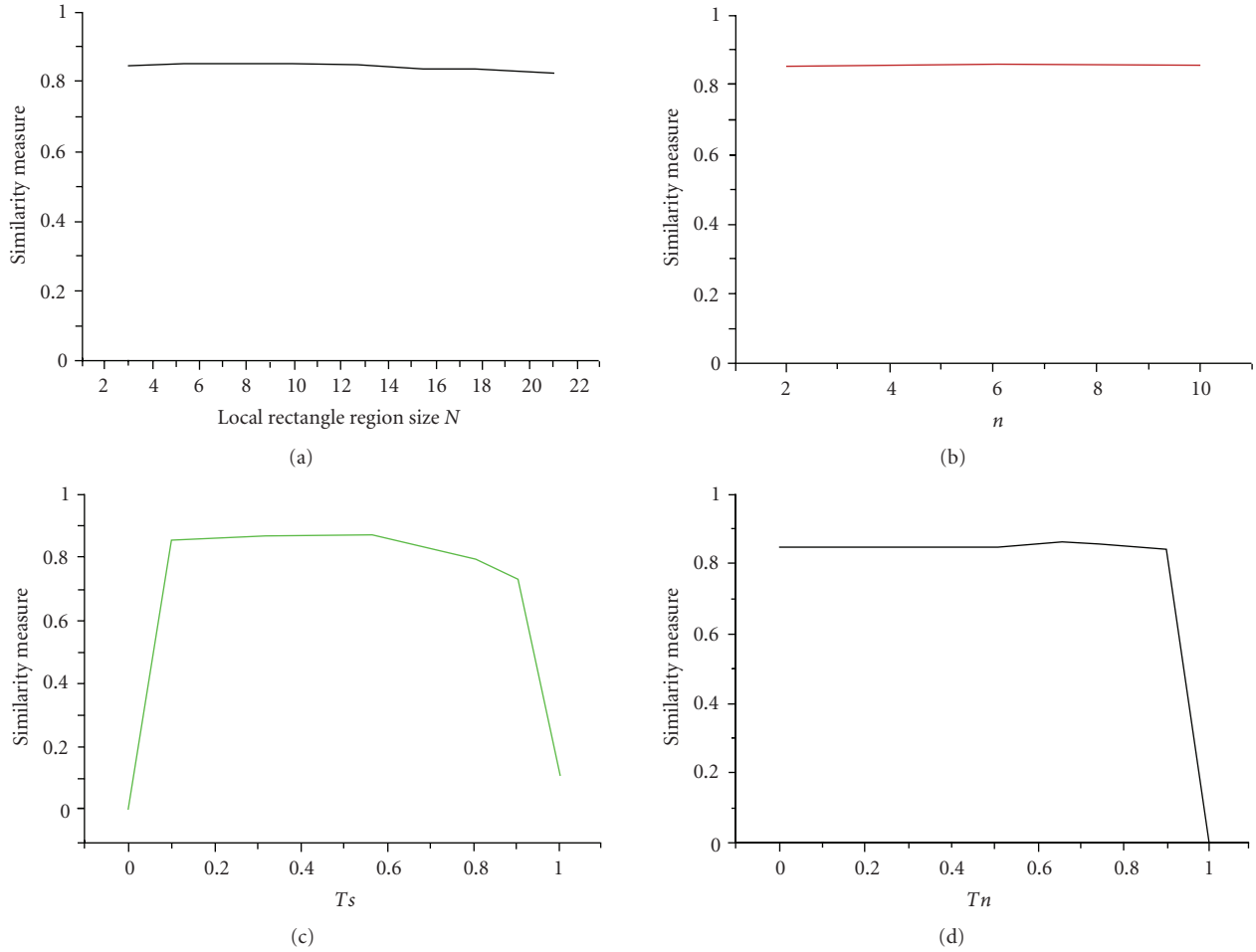

Figure 11: The similarity measures as a function of different parameter values for the Fountain sequence of Figure 6. While one parameter is varied, other parameters are kept fixed at the values given in Table 1. For each parameter configuration, the final similarity measure is achieved by averaging the similarity measures obtained from all frames.

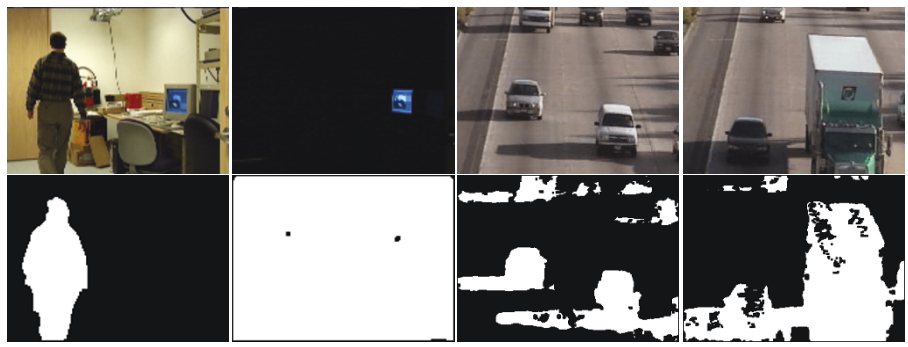

Figure 12: Some failed detection results by our method. The first row contains the original video frames. The second row contains the corresponding detection results. Challenges in the first two and the rest columns are light switch and cast shadow. The two image sequences are from [32, 33], respectively.

our method cannot only use multiple complementary BGS algorithms to build background models for scenes, but also exploit the spatial correlation between neighboring pixels. Extensive comparison experiments on several challenging video sequences demonstrate the advantage of the proposed method over four representatives of the current state of the art in background subtraction.

Currently, the proposed method has relatively many parameters. This may be a weakness. However, at the same time, it allows the user extensive control over method

behavior. Moreover, the experimental results have shown a proper set of parameters can be easily found for a given application scenario.

Acknowledgments

The authors thank Professor Ahmed Elgammal, Professor Yaser Sheikh, and Dr. Valtteri Takala for kindly sharing their code and data. This work is supported by Natural Science Foundation of China (nos. 60775024 and 60803147), National Basic Research Program of China (no. 2009CB320906).

References

[1] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "P finder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[2] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[3] O. Javed, K. Shafique, and M. Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Proceedings of the IEEE Workshop on Motion and Video Computing*, pp. 22–27, 2002.

[4] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection—a survey," *Recent Patents on Computer Science*, vol. 1, no. 3, pp. 219–237, 2008.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809–830, 2000.

[6] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[7] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric Model for Background Subtraction," in *Proceedings of the European Conference on Computer Vision (ECCV '00)*, vol. 2, pp. 751–767, June 2000.

[8] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778–1792, 2005.

[9] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 302–309, July 2004.

[10] T. Parag, A. Elgammal, and A. Mittal, "A framework for feature selection for background subtraction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 1916–1923, June 2006.

[11] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.

[12] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[13] J. Yao and J.-M. Odobez, "Multi-layer background subtraction based on color and texture," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.

[14] W. Z. Hu, H. F. Gong, S.-C. Zhu, and Y. T. Wang, "An integrated background model for video surveillance based on primal sketch and 3D scene geometry," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[15] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 1, pp. 255–261, September 1999.

[16] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 1, pp. 44–50, October 2003.

[17] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 1305–1312, October 2003.

[18] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1291–1296, 2002.

[19] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 2, pp. 65–72, June 2003.

[20] K. A. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 746–751, 2008.

[21] T. Ko, S. Soatto, and D. Estrin, "Background subtraction on distributions," in *European Conference on Computer Vision (ECCV '08)*, October 2008.

[22] G. Dalley, J. Migdal, and W. E. L. Grimson, "Background subtraction for temporally irregular dynamic textures," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '08)*, January 2008.

[23] V. Mahadevan and N. Vasconcelos, "Background subtraction in highly dynamic scenes," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[24] Y.-L. Tian, M. Lu, and A. Hampapur, "Robust and efficient foreground analysis for real-time video surveillance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 1182–1187, June 2005.

[25] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 774–780, 2000.

[26] T. Matsuyama, T. Ohya, and H. Habe, "Background subtraction for non stationary scenes," in *Proceedings of the Asian Conference on Computer Vision (ACCV '00)*, pp. 622–667, 2000.

[27] M. Mason and Z. Duric, "Using histograms to detect and track objects in color video," in *Proceedings of the Applied Imagery Pattern Recognition Workshop*, pp. 154–159, 2001.

[28] Y.-T. Chen, C.-S. Chen, C.-R. Huang, and Y.-P. Hung, "Efficient hierarchical method for background subtraction," *Pattern Recognition*, vol. 40, no. 10, pp. 2706–2715, 2007.

[29] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, "Detection and location of people in video images using adaptive fusion of color and edge information," in *Proceedings of the International Conference on Pattern Recognition (ICPR '00)*, vol. 4, pp. 627–630, 2000.

[30] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, pp. 155–168, October 2008.

[31] F. Porikli, "Integral histogram: a fast way to extract histograms in Cartesian spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 829–837, June 2005.

[32] http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm.

[33] http://cvrr.ucsd.edu/aton/shadow/.

[34] http://projects.cwi.nl/dyntex/.

[35] http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/#testvideo.

[36] http://www.cs.cmu.edu/~yaser/.

[37] http://www.cs.rutgers.edu/~elgammal/Research/BGS/research_bgs.htm.