Video Object Segmentation and Tracking Framework With Improved Threshold Decision and Diffusion Distance

Shao-Yi Chien, *Member, IEEE,* Wei-Kai Chan, Yu-Hsiang Tseng, and Hong-Yuh Chen

***Abstract*—Video object segmentation and tracking are two essential building blocks of smart surveillance systems. However, there are several issues that need to be resolved. Threshold decision is a difficult problem for video object segmentation with a multibackground model. In addition, some conditions make robust video object tracking difficult. These conditions include nonrigid object motion, target appearance variations due to changes in illumination, and background clutter. In this paper, a video object segmentation and tracking framework is proposed for smart cameras in visual surveillance networks with two major contributions. First, we propose a robust threshold decision algorithm for video object segmentation with a multibackground model. Second, we propose a video object tracking framework based on a particle filter with the likelihood function composed of diffusion distance for measuring color histogram similarity and motion clue from video object segmentation. The proposed framework can track nonrigid moving objects under drastic changes in illumination and background clutter. Experimental results show that the presented algorithms perform well for several challenging sequences, and our proposed methods are effective for the aforementioned issues.**

***Index Terms*—Diffusion distance (DD), particle filter, smart camera, surveillance, threshold decision, tracking.**

I. Introduction

In RECENT years, content analysis in smart surveillance has become increasingly important, and emerging systems [1]–[3] are being studied and deployed in real environments. In future surveillance networks, content analysis engines embedded in smart cameras will play an important role [4]. In embedded content analysis algorithms, video object segmentation and tracking get the most attention as they are critical building blocks for other smart surveillance functions.

Manuscript received February 5, 2012; revised May 30, 2012 and September 9, 2012; accepted November 27, 2012. Date of publication January 24, 2013; date of current version May 31, 2013. This work was supported by the National Science Council, under Grant NSC 100-2221-E-002-090-MY3. This paper was recommended by Associate Editor S. Battiato.

S.-Y. Chien and H.-Y. Chen are with the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: sychien@cc.ee.ntu.edu.tw; keane@media.ee.ntu.edu.tw).

W.-K. Chan is with MediaTek Inc., Hsinchu 30078, Taiwan (e-mail: f93943041@ntu.edu.tw).

Y.-H. Tseng is with MStar Semiconductor Inc., Hsinchu 302, Taiwan (e-mail: b92901020@ntu.edu.tw).

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

Digital Object Identifier 10.1109/TCSVT.2013.2242595

Several video object segmentation algorithms have been proposed under various environmental assumptions. In [5]–[8], several simple and efficient video object segmentation algorithms are proposed. However, the proposed algorithms cannot address dynamic backgrounds because only one background layer is employed in their background model. However, in [9]–[11], multilayered, complex background models are employed to address dynamic backgrounds; the large memory requirements of these models result in implementation bottlenecks, especially in embedded systems, such as smart cameras [12]. Performance comparisons of most existing video object segmentation algorithms are conducted with artificial datasets [13] that indicate that algorithms with multilayer background models (multimodels) outperform those with single background models (unimodels). Vosters *et al.* [14] proposed a more complex algorithm, consisting of an eigenbackground and statistical illumination model, which can address sudden changes in illumination; however, this algorithm is too complex to be integrated into existing smart camera platforms. A threshold decision algorithm for deciding appropriate threshold values is also very important since the segmentation results deeply rely on the threshold value. In [15], we proposed a threshold decision method for single background models; however, it still cannot successfully and robustly determine the threshold values for conditions with dynamic background.

For video object tracking, the data association of segmentation blobs [16] is highly dependent on the quality of segmentation results. Gradient descent-based methods [17], [18] search for the most likely object candidate regions with gradient descent optimization techniques. However, they suffer from local minima problems, and it is difficult for them to address objects that have large motions. In [19], the Kalman filter is employed to predict object motion and track objects; however, it may fail for objects that have random motions. Particle filter [20], [21] is a more robust methodology for object tracking, and it can address large and random motions more effectively; however, the features employed for object modeling and the distance measurements used to decide the weights of the particles, which are essential for making these algorithms effective, have to be appropriately selected and designed. For object modeling, color, gradient, edge, texture, and motion are usually the features employed. However, several defects may appear when these features are used as the object model. For example, models with color features

may fail to address appearance variations due to changes in the illumination of the environment. On the other hand, models with gradient, edge, and texture features are not robust when the gradient, edge, or texture feature is not obvious or when the target under tracking is a nonrigid object since these features are usually variant along with the nonrigid object motion. Moreover, the background clutter problem, in which the objects and the background are similar in color, gradient, edge, or texture, also has to be overcome. Furthermore, the computational complexity should be taken into account since particle filter is a more computationally intensive algorithm.

In this paper, a video object segmentation and tracking framework for smart visual surveillance cameras is proposed with two major contributions. First, we propose a robust threshold decision algorithm for video object segmentation with a multibackground model. The proposed algorithm can determine an appropriate optimal threshold value for our proposed multibackground model; hence, it can enable good performance for conditions with dynamic backgrounds without threshold tuning by developers. In addition, it is based on a mechanism that is different from that of background subtraction-based video object segmentation, which can prevent possible error propagations. Second, we propose a video object tracking framework based on a particle filter with diffusion distance (DD) for measuring color histogram similarity and motion clues from video object segmentation. For better tracking of nonrigid objects, we include color histogram in our object model as it is more stable for nonrigid moving objects. In addition, cross-bin distance measurements for the color histogram can address appearance variations resulting from changes of illumination more effectively, and DD is used for such measurements in our proposal as it is more suitable for parallel implementation and hardware realization. Further to this, 1-D color histogram is used instead of 3-D color histogram in order to reduce computational complexity. Finally, with the motion clue from video object segmentation, our proposed method can address the background clutter problem more effectively with existing information, and no other computation for extracting motion information is required. Note that the proposed segmentation and tracking framework are feasible for embedding in smart cameras in visual surveillance networks (an example is given in [22]).

The remainder of this paper is organized as follows. The proposed threshold decision for video object segmentation is presented in Section II. In Section III, the proposed video object tracking framework with DD and motion clue from segmentation is presented. Experimental results demonstrating the performance of the proposed algorithms are given in Section IV. Finally, this paper is concluded in Section V.

II. Proposed Threshold Decision for Video Object Segmentation

The video object segmentation algorithm introduced in the following section is based on our previous work [23], in which a multibackground registration scheme was proposed to model complex and dynamic backgrounds. To make it fully automatic for variant conditions, in this section, an automatic threshold decision technique that can automatically and precisely determine the threshold values for dynamic backgrounds is proposed and presented. The segmentation algorithm is also designed with a low memory requirement for background storage.

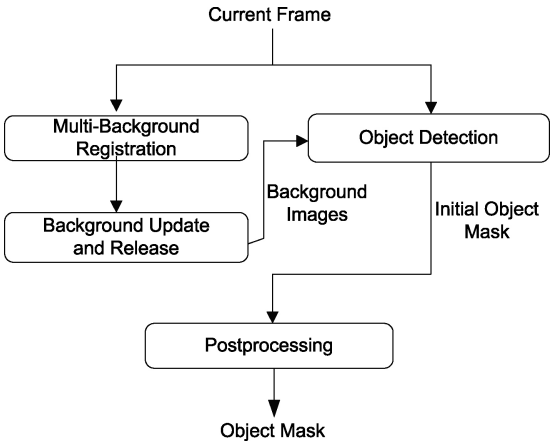


Fig. 1. Block diagram of the proposed video segmentation algorithm.

A. Video Object Segmentation With Multibackground Registration

Our segmentation method is based on an online multilayer background modeling technique called multibackground registration (MBReg), whose block diagram is shown in Fig. 1. The key concept in this algorithm is the fact that it models the background with N layers of background images instead of a single background layer. For each pixel position, the corresponding pixel in each layer of the background image represents one possible background pixel value. In Fig. 1, the background model is established and maintained in the MBReg and background update and release blocks. In the MBReg block, each input pixel of the current frame $CurFrm(i, j, t)$, where (i, j) is the pixel position and t is the time index, is compared with the corresponding background pixels in the multibackground image BImg(i, j, t−1, k), where $k \in [1, N]$, and a matching flag, match(i, j, k), is recorded by the following equation:

$$\text{match(i, j, t, k)} = \begin{cases} 1 & \text{if } \text{BD(i, j, t, k)}_{\text{y}} \le \text{BDth(i, j, t, k)}_{\text{y}} \\ & \&\text{BD(i, j, t, k)}_{\text{u}} \le \text{BDth(i, j, t, k)}_{\text{u}} \\ & \&\text{BD(i, j, t, k)}_{\text{v}} \le \text{BDth(i, j, t, k)}_{\text{v}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

in which $k \in$ built background layers, YUV color space is employed, BD(i, j, t, k) denotes the background difference, which can be calculated using the following equations:

$$\text{BD(i, j, t, k)}_{\text{y}} = |CurFrm(i, j, t)_{y} - \text{BImg(i, j, t}-1, \text{k)}_{\text{y}}| \tag{2}$$

$$\text{BD(i, j, t, k)}_{\text{u}} = |CurFrm(i, j, t)_{u} - \text{BImg(i, j, t}-1, \text{k)}_{\text{u}}| \tag{3}$$

$$\text{BD(i, j, t, k)}_{\text{v}} = |CurFrm(i, j, t)_{v} - \text{BImg(i, j, t}-1, \text{k)}_{\text{v}}| \tag{4}$$

and BDth(·) is the background threshold value for each color channel of each background pixel.

In the background update and release block, an unmatched background counter, CntSNO(i, j, t, k), and a weighting coefficient, Wgt(i, j, t, k), are maintained to record the duration

when a background pixel is unmatched to the input pixel and the confidence of each background pixel

$$\mathrm{CntSNO}(i,j,t,k) = \begin{cases} 0 & \text{if } \mathrm{match}(i,j,t,k)=1 \\ \mathrm{CntSNO}(i,j,t-1,k)+1 & \text{otherwise} \end{cases} \quad (5)$$

$$\mathrm{Wgt}(i,j,t,k) = \begin{cases} \mathrm{Wgt}(i,j,t-1,k)+1 & \text{if } \mathrm{match}(i,j,t,k)=1 \\ \mathrm{Wgt}(i,j,t-1,k)-1 & \text{if } \mathrm{CntSNO}(i,j,t,k) > \mathrm{BDF}(i,j,t,k) \\ \mathrm{Wgt}(i,j,t-1,k) & \text{otherwise} \end{cases} \quad (6)$$

where BDF is the background decaying factor. With the unmatched background counter and weighting coefficient, the background model can be updated or released with the following equations:

$$\mathrm{BImg}(i,j,t,k)_y = \begin{cases} \mathrm{UpdBckgnd}(i,j,t,k)_y & \text{if } \mathrm{match}(i,j,t,k)=1 \\ 0(\text{release}) & \text{if } k \in \text{Built background layers} \\ & \quad \& \ \mathrm{Wgt}(i,j,t,k) < \mathrm{RELth}(i,j,t,k) \\ 0(\text{release}) & \text{if } \mathrm{Wgt}(i,j,t,k)=0. \end{cases} \quad (7)$$

$$\mathrm{UpdBckgnd}(i,j,t,k)_y = \frac{\mathrm{BImg}(i,j,t-1,k)_y \times \mathrm{Wgt}(i,j,t,k) + \mathrm{CurFrm}(i,j,t)_y}{\mathrm{Wgt}(i,j,t,k)+1} \quad (8)$$

$$\text{Built background layers} = \{k \mid \mathrm{Wgt}(i,j,t,k) \geq \mathrm{RegInth}(i,j,t,k)\} \quad (9)$$

where RELth(i, j, t, k) is the background release threshold, and RegInth(i, j, t, k) is the background registration threshold. These two thresholds are used to control the background update speed. Note that the update schemes for the U and V channels are the same and are thus omitted here. An unmatched input pixel can still be placed in the background with the weighting parameter set to zero if there is still an empty background layer (i.e., #{Built background layers} $< N$) for background initialization and learning.

After the MBReg and background update and release blocks, in the object detection block in Fig. 1, the current frame and the background images are compared in order to generate the initial object mask as follows:

$$\mathrm{Ini\_Object\_Mask}(i,j,t) = \begin{cases} 0 & \exists k \in \text{Built background layers}, \\ & \mathrm{BD}'(i,j,t,k)_y \leq \mathrm{BDth}(i,j,t,k)_y \\ & \&\mathrm{BD}'(i,j,t,k)_u \leq \mathrm{BDth}(i,j,t,k)_u \\ & \&\mathrm{BD}'(i,j,t,k)_v \leq \mathrm{BDth}(i,j,t,k)_v \\ 1 & \text{otherwise} \end{cases} \quad (10)$$

where $\mathrm{BD}'(\cdot)$ is calculated using (2), (3), and (4) but with the updated background BImg(i, j, t, k). A match indicates that the current pixel belongs to the background, and the detection result is negative; otherwise, the detection result is positive. In the postprocessing block in Fig. 1, the object mask is denoised by means of binary morphological operations.

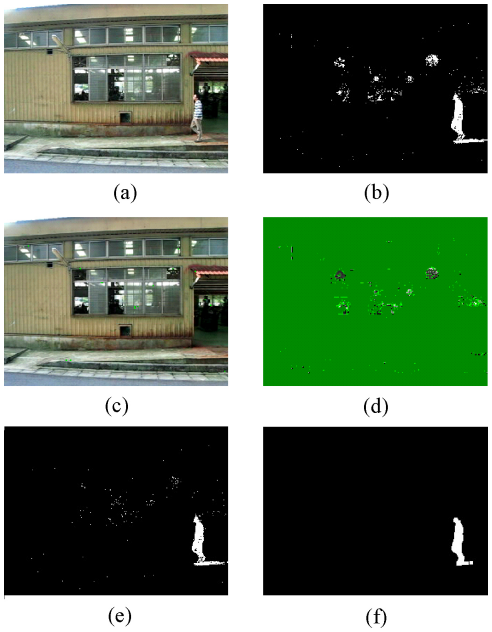


Fig. 2. Example of MBReg and segmentation, in which the test sequence is *Fans*. (a) Current frame of *Fans*. (b) Initial object mask with the original single background registration-based segmentation [15]. (c) First background layer BImg(i, j, 0). (d) Second background layer BImg(i, j, 1). (e) Ini_Object_Mask(i, j) with MBReg-based segmentation ($N = 4$). (f) Final object mask.

Note that approaches such as those in [8] and [15] that have been proposed to remove the shadows and ghosts of the object masks can still be employed here; however, the related discussions are beyond the scope of this paper.

An example demonstrating the proposed segmentation flow is shown in Fig. 2. In the figure, a video sequence with several rotating fans in the background region is employed. The current frame of the sequence $CurFrm(i, j)$ is shown in Fig. 2(a). The background images, BImg(i, j, 0) and BImg(i, j, 1), produced by the MBReg and background update and release blocks are shown in Fig. 2(c) and (d). The initial object mask, Ini_Object_Mask(i, j), produced by the object detection block, is shown in Fig. 2(e), and the final object mask, produced after postprocessing, is shown in Fig. 2(f), where morphological opening and closing operations are applied. The initial object mask generated by the original single background registration segmentation [15] is also shown in Fig. 2(b) for comparison. It shows that the multiple (varying) values of each pixel on the rotating fans will be registered on different layers of the corresponding position of the proposed background model, which is demonstrated in Fig. 2(c) and (d). As a result, the fans are not segmented as foreground objects. Consequently, the result shown in Fig. 2(e) is better than that shown in Fig. 2(b).

B. Proposed Threshold Decision

There are several threshold values in the proposed MBReg-based video object segmentation algorithm. They include $\mathrm{BDth}(i,j,k,t)_y$, $\mathrm{BDth}(i,j,k,t)_u$, $\mathrm{BDth}(i,j,k,t)_v$, $\mathrm{BDF}(i,j,k,t)_v$, $\mathrm{RELth}(i,j,k,t)$, and $\mathrm{RegInth}(i,j,k,t)$. In our empirical study, it was discovered that these thresholds

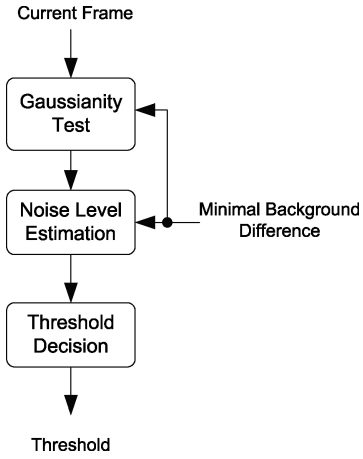


Fig. 3. Block diagram of the proposed threshold determination algorithm.

can be set globally for all the positions and background layers without degrading the quality of the generated object masks. That is, the indices i, j, and k can be removed from these thresholds. In addition, among these thresholds, BDF(t), RELth(t), and RegInth(t) are used to control the update speed of the background model. The parameters are not sensitive in the proposed algorithm and can thus be set as constants or adjusted according to the requirements of different scenarios. However, the thresholds $\mathrm{BDth_y}(t)$, $\mathrm{BDth_u}(t)$, and $\mathrm{BDth_v}(t)$ used in (1) and (10) are critical parameters that are highly correlated with the camera noise and should thus be temporally adjusted in order to address changes in illumination. In [15], we proposed a threshold decision method for the single background registration-based video segmentation algorithm. However, this algorithm cannot successfully and robustly determine the thresholds for conditions in which there are dynamic backgrounds.

In this paper, we propose an improved threshold decision algorithm for our video object segmentation algorithm with multiple backgrounds. In addition to the ability to address dynamic backgrounds, the threshold decision algorithm required must have several desirable characteristics. First, it must be able to determine the optimal thresholds without any user input. Second, since background subtraction-based video object segmentation may cause an error propagation problem while updating the per-pixel background model, a threshold determination algorithm that is based on a different mechanism is required here. Furthermore, the quantization effect of digital systems also has to be taken into consideration.

To meet these requirements, we proposed the threshold decision algorithm outlined in Fig. 3. The algorithm is based on the assumption that the camera noise is in the zero-mean Gaussian distribution, and the camera noise is the only factor affecting the optimal thresholds. The proposed algorithm consists of three sections: Gaussianity test, noise level estimation, and threshold decision. Note that this proposed threshold decision algorithm is based on a mechanism that is different from that of the per-pixel background subtraction algorithm.

1) *Gaussianity Test:* Before measuring the parameters of camera noise, the background of the input frame should be automatically indicated because it is very difficult to correctly measure the camera noise from the foreground. Since we assume that the camera noise is Gaussian distributed, the minimal background difference (BDmin) of the background should also be Gaussian distributed:

$$\mathrm{BDmin(i,\,j,\,t)_y} = \mathrm{CurFrm(i,\,j,\,t)_y} - \mathrm{BImg(i,\,j,\,t,\,k^*)_y}, \quad (11)$$

where

$$k^* = \arg\min_k \quad \{|CurFrm(i,\,j,\,t) - \mathrm{BImg}(i,\,j,\,t,\,k)| \mid k \in \text{Built background layers}\},$$

and $\mathrm{BDmin(i,\,j,\,t)_u}$ and $\mathrm{BDmin(i,\,j,\,t)_v}$ can be calculated similarly. With this property, the Gaussianity test [24], which can indicate if a group of values is Gaussian distributed or not, can be used to find the background. First, the frame is divided into a number of nonoverlapped $M_b \times N_b$ blocks. The Gaussianity test is then applied to each block to determine if the minimal background differences in the block are Gaussian distributed or not. The Gaussianity test can be shown as the following equations:

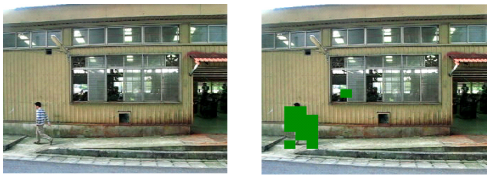
$$I_r = \frac{1}{M_b \times N_b} \sum_{m=1}^{M_b} \sum_{n=1}^{N_b} [\mathrm{BDmin(m,\,n)}]^{\mathrm{r}} \quad (12)$$

$$H(I_1, I_2, I_3, I_4) = I_3 + I_4 - 3I_1(I_2 - I_1^2) - 3I_2^2 - I_1^3 - 2I_1^4. \quad (13)$$

1) Gaussian: $|H(I_1, I_2, I_3, I_4)| < \mathrm{Gth}$.
2) Non-Gaussian: $|H(I_1, I_2, I_3, I_4)| \geq \mathrm{Gth}$.

where the smaller the H value, the closer the distribution of $\mathrm{BD_{min}}$ is to the Gaussian distribution, and Gth is the threshold value for binarizing the decision. If the minimal background differences in a block are Gaussian distributed, the block belongs to the background region because the (minimal) difference between the current frame and the background images is only caused by noise; that is, no foreground objects exist in this block to cause additional differences. Note that in [15], the frame differences are obtained by subtracting the current frame from the previous frame. However, that method cannot address dynamic background cases, since the frame differences in dynamic background regions are obviously not Gaussian distributed, and the frame differences obtained in dynamic background regions cannot be used to estimate the noise level.

An example of the Gaussianity test is illustrated in Fig. 4. Only blocks whose minimal background differences are Gaussian distributed are shown in Fig. 4(b), where the green blocks are non-Gaussian-distributed blocks. It demonstrates that the Gaussianity test can roughly distinguish the background from the foreground. Parameter Gth is set as a constant because the H values in (13) are significantly different for the foreground blocks and background blocks. Note that although the Gaussianity test can already distinguish between the background and the foreground and can give rough object masks, but it cannot give pixel-wised segmentation results. The function of the Gaussianity test is to determine the optimal threshold without any information given by users, and also to make the threshold decision procedure independent of the per-pixel segmentation procedure in order to avoid error propagation.



(a) (b)

Fig. 4. Illustration of the Gaussianity test. (a) Input frame of sequence *Fans*. (b) Result of the Gaussianity test.

2) *Noise Level Estimation and Threshold Decision:* The optimal threshold $\mathrm{BDth_y}$ can be expressed as in the following equation:

$$\mathrm{BDth_y^*} = \max\{|\mathrm{BDmin(i,j)_y}| \mid (\mathrm{i,j}) \in \text{Background blocks}\} \tag{14}$$

where the background blocks are indicated by the Gaussianity test described in the previous section. Note that $\mathrm{BDth_y^*}$ and $\mathrm{BDmin(i,j)_y}$ are all random variables. Since the minimal background differences are Gaussian distributed, and the quantization effect of digital systems is considered, the distribution of the minimal background differences in the digital domain should be

$$P_{yd} = \begin{cases} \int_{-0.5}^{0.5} \frac{1}{\sigma_y\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma_y^2}} dy = \int_{0}^{0.5} \frac{2}{\sigma_y\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma_y^2}} dy \\ \qquad\qquad\qquad\qquad \text{for } d = 0 \\ \int_{d-0.5}^{d+0.5} \frac{1}{\sigma_y\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma_y^2}} dy + \int_{-d-0.5}^{-d+0.5} \frac{1}{\sigma_y\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma_y^2}} dy = \\ \quad \int_{d-0.5}^{d+0.5} \frac{2}{\sigma_y\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma_y^2}} dy \quad \text{for } d = 1, 2, \ldots, 255 \end{cases}, \tag{15}$$

where $P_{yd} = P(|\mathrm{BDmin_y} = \mathrm{d}|)$. The expected value of the optimal threshold should be

$$\begin{aligned} E(\mathrm{BDth_y^*}) &= \sum_{d=0}^{255} d \cdot P(\mathrm{BDth_y^*} = \mathrm{d}) \\ &= \sum_{d=0}^{255} d \cdot \left[\binom{Ns}{1} P(|\mathrm{BDmin_y}| = \mathrm{d}) \right. \\ &\qquad \left. P(|\mathrm{BDmin_y}| \le \mathrm{d})^{\mathrm{Ns}-1} \right] \\ &= \sum_{d=0}^{255} d \cdot \left\{ (Ns) P_{yd} \left[\sum_{m=0}^{d} P_{ym} \right]^{Ns-1} \right\} \end{aligned} \tag{16}$$

where N_s is the number of pixels belonging to the background blocks.

Nevertheless, the parameter σ_y is difficult to derive from the minimal background differences because the minimal background differences are already quantized as integers in digital systems. That is

$$\sigma_y' = \sqrt{\frac{\sum_{(i,j)\in \text{Background blocks}} [\mathrm{BDmin_y(i,j)}]^2}{Ns}} \tag{17}$$

is different from σ_y. By substituting σ_y with different values, the expected optimal threshold $E(\mathrm{BDth_y^*})$ can be derived using (15) and (16), and the associated σ_y' is also calculated to form a σ_y'–$E(\mathrm{BDth_y^*})$ mapping curve. In our implementation, noise level estimation uses the minimal background difference values of all the background blocks indicated by the Gaussianity test to calculate the value of σ_y' using (17), and threshold decision determines the optimal threshold using the mapping curve, which causes only a simple table look-up operation since the curve is precomputed offline. The same procedure is also employed to derive $E(\mathrm{BDth_u^*})$ and $E(\mathrm{BDth_v^*})$.

TABLE I
MEMORY REQUIREMENT FOR MODELING EACH PIXEL

Algorithm	Memory Requirement (Bytes)
Kernel [9]	900
MoG with 10 Gaussians [10]	200
MoG with 4 Gaussians [10]	80
Codebook [11]	112
MBReg with 4 layers	51
MBReg with 4 layers and global thresholds	27

C. Memory Reduction in Background Modeling

In this section, the memory requirements of our proposed method (MBReg) and that of previous works [9]– [11] are compared. In Table I, the memory requirements for modeling each pixel is shown (refer to [11]). The kernel-based background model requires 3 B for each sample and would require 900 B if 300 samples are required for each pixel position [9], [11]. The mixture-of-Gaussian (MoG) background model is a well-known method for foreground object segmentation [10]. It requires five floating-point numbers for each pixel to store RGB means, a variance, and a weight parameter. It would require 200 B for 10 Gaussians, and 80 B for four Gaussians. In the codebook model [11], five floating-point numbers and four 16-bit integers are required for each codeword. It is claimed that four codewords are needed on average, which results in a total of 112 B. In our method, 3 B are required for store $\mathrm{BImg_y}$, $\mathrm{BImg_u}$, and $\mathrm{BImg_v}$; 20 b for Wgt; 1 B for CntSNO; 3 B for $\mathrm{BDth_y}$, $\mathrm{BDth_u}$, and $\mathrm{BDth_v}$; 3 B for RegInth, RELth, and BDF; and two extra bits to flag the status of each pixel in a single layer. That is, 102 b are required for a single layer, and 51 B for four layers. Note that the performance of our approach with four background layers should be similar to that of the codebook model with four codewords. Moreover, since the threshold values are set as global parameters for all positions and layers, the required memory size is further reduced to 27 B for each four-layer pixel. Compared with the MoG model with 4 Gaussians, a reduction rate of 66% is achieved with our proposed multibackground model. The advantage is quite significant. For a 720×576 30 f/s video, for example, memory savings of $720\times576\times(80-27) = 21\,980\,160$ B can be achieved.

III. VIDEO OBJECT TRACKING

The video object tracking process can be modeled with the hidden Markov model (HMM) shown in Fig. 5. In this model, system state, X_t, is the object state to be estimated at time t

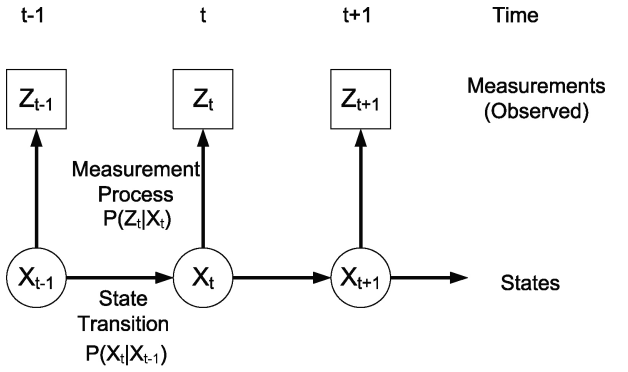


Fig. 5. Illustration of system model. The video object tracking problem can be modeled with a hidden Markov model (HMM).

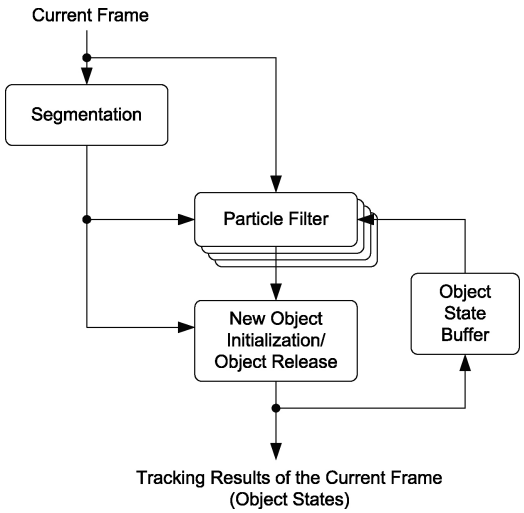


Fig. 6. Block diagram of the proposed object tracking algorithm.

(e.g., object position, object velocity, object size, etc). Measurement, Z_t, is the observation from the current frame. Video object tracking is an inverse problem; that is, object states are inferred from the measurements (features extracted from video sequences) such as color, texture, and gradient. The model in Fig. 5 can be specified further using the following equations:

$$X_t = f(X_{t-1}) + n_x, \tag{18}$$

$$Z_t = g(X_t) + n_z, \tag{19}$$

where $f(\cdot)$ is the state transition function, $g(\cdot)$ is the measurement function, and n_x and n_z are noise.

Our proposed video object segmentation and tracking framework are depicted in Fig. 6. The object masks derived from the video object segmentation algorithm described in Section II are employed as a part of the measurement. We utilize the particle filter framework to infer the object state from the measurements as the particle filter framework can better track objects with random motion and large motion. For measurements other than the object masks, we also utilize the color histogram feature in our target representation since it is a more robust global feature for tracking nonrigid moving objects. The most important contribution of this framework is that we propose the use of Diffusion Distance (DD) as a cross-bin color histogram distance function to improve the tracking performance, especially when the object being tracked has appearance variations due to changes in illumination. The outputs of particle filter are the updated object states. Those object masks that have large area but are not included by any trackers belong to new incoming objects. The trackers for these new objects are initialized with the object masks and added to the object state buffer, as shown in Fig. 6.

A. Particle Filter

The particle filter is a sampling, importance measuring, and resampling (SIR) framework to estimate the underlying posterior distribution of the system state [21], [25]. In the particle filter, the posterior distribution of the system state, which is the object state here, is approximated by

$$p(X_t|Z_{1:t}) \approx \sum_{k=1}^{N} w_{tk}\delta(X_t - S_{tk}) \tag{20}$$

were N is the number of particles, S_{tk} is the kth particle (i.e., an assumption of the object state), w_{tk} is the weight of the kth particle, and the posterior distribution can be approximated with the weighted sum of these particles. Note that the particles are generated according to a resampling function determined by the weights [20], [25]. The formula for calculating the weight of the particles can be approximated with the following likelihood function:

$$w_{tk} \propto p(Z_t|X_t = S_{tk}) \tag{21}$$

which means that if the assumption (particle) is more likely to be the true object state, the particle is given a greater weight. The normalized weight is

$$\pi_{tk} = \frac{(w_{tk})}{\sum_{j=1}^{N} w_{tj}}. \tag{22}$$

The estimated state O_t can then be derived using the following equation:

$$O_t = \sum_{k=1}^{N} \pi_{tk} S_{tk}. \tag{23}$$

Among these steps, the importance measuring step [i.e., the calculation of the likelihood function $p(Z_t|X_t = S_{tk})$ in (21)] is the most important since it defines the "objective function" to be optimized. That is to say, it is the similarity function to be maximized through the optimization process for finding the most likely state (position and size) of the object. We employ the color histogram and motion clue from segmentation to calculate the likelihood function.

B. Color Histogram Distance: Bin-to-Bin Versus Cross-Bin

In video object tracking, different kinds of features, such as color, texture, and gradient [20], [26], can be utilized to measure the importance of particles, where the distance functions, which are defined for those features, between the tracked object of the previous frame and each assumption

(particle) in the current frame are calculated. For tracking nonrigid objects, a color histogram is usually utilized since the stability of a color histogram is better than that of other features under the nonrigid motion. However, a color histogram suffers from appearance variations due to changes in illumination. Therefore, a histogram distance calculation scheme that is robust to changes of illumination is critical for object tracking.

There are two kinds of histogram distances: cross bin and bin-by-bin. Assume there are two histograms, $P = \{p_i\}$ and $Q = \{q_i\}$, where p_i and q_i are the corresponding normalized bin values, respectively. P is the color histogram of the tracked object of the previous frame, and Q is the color histogram of one assumption (particle) in the current frame. The bin-by-bin method only compares values in corresponding histogram bins; that is, it only compares p_i and q_i for all i, but not p_i and q_j for $i \neq j$. In contrast, the cross-bin method may contain terms that compare noncorresponding histogram bins. Moreover, the cross-bin method also takes the distance between two bins into account, while the bin-by-bin method does not.

As is expected, the bin-by-bin histogram distance is sensitive to changes in illumination. Therefore, in contrast to previous works that utilize bin-to-bin color histogram distance, such as Bhattacharyya distance (BhD), we propose to utilize cross-bin histogram distance to improve the tracking performance under changes in illumination. We also propose the use of DD instead of earth mover's distance (EMD), employed by some image analysis systems, as the cross-bin distance function. In the following sections, the three distance functions are presented, followed by the computation reduction method and the proposed likelihood function (weight decision function).

1) *Bhattacharyya Distance:* Many visual tracking algorithms that employ a color histogram in their target representation utilize BhD [17], [20], [26]. The BhD can be computed with the following equation:

$$\mathrm{BhD}(\mathrm{P}, \mathrm{Q}) = -\ln(\mathrm{BC}(\mathrm{P}, \mathrm{Q})) \tag{24}$$

where BC(P, Q) is called the Bhattacharyya Coefficient between histogram P and histogram Q, and

$$\mathrm{BC}(\mathrm{P}, \mathrm{Q}) = \sum_{\mathrm{i}=1}^{\mathrm{M}} \sqrt{\mathrm{p_i q_i}} \tag{25}$$

where M is the total number of histogram bins. From the above equations, it can be seen that the BhD only considers the values of corresponding bins. Therefore, it is sensitive to local histogram changes or bin shifting caused by changes in illumination.

2) *Earth Mover's Distance:* EMD is used in many computer vision and image analysis application domains [27], [28]. It can also be used for visual tracking.

From the view-point of the transportation problem (see Fig. 7), the EMD seeks the minimal effort needed to transport the mass to the holes, as illustrated in Fig. 7, where the amount of mass in source i is p_i and the capacity of hole j is q_j. The distance d_{ij} (effort to transport a single unit of mass) between a source i and a hole j can be specified according to different applications, and the flow (amount of mass that is moved)

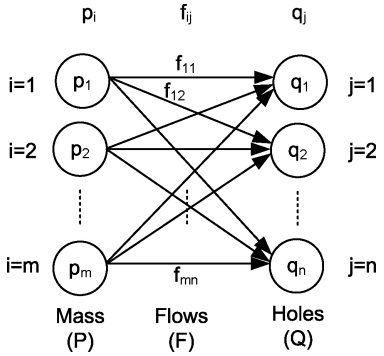


Fig. 7. Illustration of transportation problem and EMD.

between a source i and a hole j is f_{ij}. The EMD is defined as the minimal effort that is required to move all of the mass to fill all of the holes. That is

$$\mathrm{EMD}(\mathrm{P}, \mathrm{Q}) = \min_{\mathrm{F}} \frac{\sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{d_{ij} f_{ij}}}{\sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{f_{ij}}} \tag{26}$$

where $F = \{f_{ij}\}$ subject to the following constraints:

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n \tag{27}$$

$$\sum_{j=1}^{n} f_{ij} = \text{Amount } p_i \text{ of Source } i \tag{28}$$

$$\sum_{i=1}^{m} f_{ij} \leq \text{Capacity } q_j \text{ of Hole } j \tag{29}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} p_i, \sum_{j=1}^{n} q_j\right). \tag{30}$$

By considering one histogram P as the mass and the other histogram Q as the hole, we can use the EMD to measure the distance between two histograms. For the normalized histogram, we also have $m = n$ and

$$\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} q_j. \tag{31}$$

To calculate the EMD, we can use the simplex method or the EMD-L_1 algorithm [27]. However, although the EMD is one of the cross-bin distance functions that can calculate the global distance between two histograms, the complexity for EMD is not acceptable. Moreover, it is difficult to achieve a parallel implementation of the EMD or realize it in hardware.

3) *Diffusion Distance:* DD was originally used in shape matching and interest point matching [29]. We propose using if for visual tracking. The DD simulates the heat diffusion process. Here, for simplicity, it can be computed using the following equations, with a multiscale histogram pyramid:

$$\mathrm{DD}(\mathrm{P}, \mathrm{Q}) = \sum_{\mathrm{l}=1}^{\mathrm{L}} \left(\sum_{\mathrm{i}=1}^{\mathrm{M_i}} |\mathrm{d_i(i)}| \right) \tag{32}$$

where L is the number of pyramid layers (number of different scales) and M_i, is the bin number of histograms

$$d_1(i) = p_i - q_i \tag{33}$$

$$d_l(i) = [d_{l-1}(i) * \phi(i, t_d)] \downarrow_2 \quad l = 2, \ldots, L \tag{34}$$

$$\phi(i, t_d) = \frac{1}{(2\pi)^{\frac{1}{2}} t_d} e^{\frac{-i^2}{2t_d^2}} \tag{35}$$

where ϕ is the Gaussian kernel, t_d is a variable that specifies the time of heat diffusion or the standard deviation for the Gaussian kernel, and "$\downarrow_2$" denotes half-size down sampling.

With the DD defined as in the above equations, the histograms are compared in different scales. For example, if two histograms are originally 256-bin in layer 1 ($l = 1$), after the Gaussian filtering and down sampling, we can compare the two histograms in their 128-bin, 64-bin, and 32-bin versions (depending on the total number of layers, L). This means that if two bins are not corresponding bins in the 256-bin version, they may belong to the same bin in the higher layers, such as the 32-bin version. Note that if one bin of one histogram and one bin of the other histogram are originally near each other in bin distance, the histogram distance is less since the bin distance would be canceled by (34) in higher layers. From this intuition, it is obvious that DD is also a cross-bin distance function, and the distance between two bins is embedded in the diffusion process of the above equations.

From the above introduction to BhD, EMD, and DD, it can seen that the EMD and DD are both cross-bin distance functions. To show why cross-bin histogram distances (EMD and DD) are more robust under changes in illumination, several examples of histogram pairs are illustrated in Fig. 8. In histogram pair A, the histogram is shifted by two bins from (A.1) to (A.2), which could be the same object but under a slight change in illumination, caused by a change in object position. Histogram B are apparently two different but overlapped histograms. However, if we use BhD to measure the distances of the histogram pairs, we will get a result that the distance of histogram pair A is larger than that of histogram pair B, which is not correct. In contrast, if we use EMD and DD, the results are correct since they are cross-bin histogram distance metrics that consider the distance between bins.

In histogram pairs C and D, histogram (C.2) is also a shifted version of histogram (C.1), and histogram (D.2) is apparently different from histogram (D.1). In BhD, the results of the histogram distance measures for C and D are infinite because there is no overlap for both histogram pairs. In EMD, it is interesting that the distances of histogram pairs C and D are equal in this case. On the other hand, DD is the only distance measurement that can tell that the distance of pair C is less than that of pair D. These two cases are summarized in Table II, which implies that, for visual tracking, DD and EMD could be the better metrics that can correctly tell us which candidate region (particle) is more likely to be the true target because they can derive the global distance between two histograms.

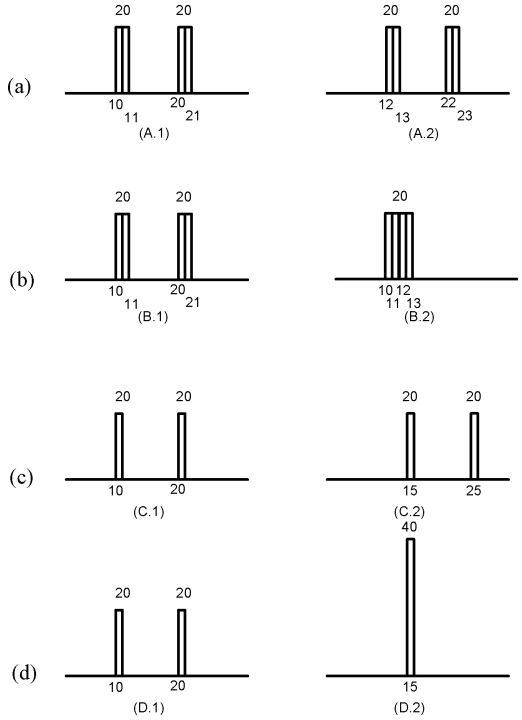


Fig. 8. Examples of histogram comparisons. Case I: histogram pairs A and B. Case II: histogram pairs C and D.

TABLE II
HISTOGRAM DISTANCE COMPARISON WITH THE EXAMPLES IN FIG. 8

	Answer	Bhattacharyya Distance	Earth Mover's Distance	Diffusion Distance
Case I	A < B	A > B	A < B	A < B
Case II	C < D	N/A	C = D	C < D

C. Computation Reduction for Histogram Distance Measurements

In color histogram distance measurements, the computation complexity depends on the number of bins in the histogram. For example, a RGB 3-D color space (we call it 3-D histogram here) with 256 bins in each dimension has in total $256 \times 256 \times 256$ histogram bins. The computation complexity grows acutely for this 3-D histogram. The corresponding complexities for BhD, EMD, and DD are shown in Table III. It can be seen that with the 3-D histogram, the complexities are $O(n^3)$, $O(n^6)$, and $O(n^3)$ for BhD, EMD, and DD, where n is the number of bins in each color space dimension. To reduce the computation time, we propose using three 1-D histograms instead of one 3-D histogram. For example, instead of using one YUV 3-D histogram, we propose to use one Y 1-D histogram, one U 1-D histogram, and one V 1-D histogram. This significantly reduces the computational complexity, as shown in Table III. The complexity for DD now becomes $O(n)$, while that for EMD becomes $O(n^2)$. This indicates that the complexity of DD is lower than that of EMD. Finally, the histogram distance value with three 1-D histograms is

$$D = \sqrt{D_Y^2 + D_U^2 + D_V^2} \tag{36}$$

TABLE III
COMPUTATION COMPLEXITY OF THREE ALGORITHMS

	Bhattacharyya Distance	Earth Mover's Distance	Diffusion Distance
3-D	$O(n^3)$	$O(n^6)$	$O(n^3)$
1-D	$O(n)$	$O(n^2)$	$O(n)$

N is the bin number of the histogram in one color dimension.

where D_Y, D_U, and D_V are the 1-D histogram distances in Y histogram, U histogram, and V histogram, respectively.

The recent trend in computation platforms is that many-core or multicore computing platforms have now become mainstream. For such platforms, DD is recommended since the computation of DD is more easily implemented in parallel. Moreover, if some specific hardware modules can be designed to accelerate the cross-bin histogram distance measurements, the DD would also be preferred since it is more suitable for hardware realization.

D. Motion Clue From Video Object Segmentation

In addition to changes in illumination, background clutter (i.e., the background and target having similar appearances, such as similar color histograms in this case) is another issue to be solved in visual tracking. In our segmentation and tracking framework, we include the motion information from video object segmentation to enhance the tracking performance under background clutter conditions, as shown in Fig. 6. The proposed particle weight decision function (likelihood function) with both motion clue and color histogram distance is

$$w_{tk} = W(D_{tk}) + \alpha \times (\text{Count}_{\text{tk}})/(\text{Area}_{\text{tk}}) \tag{37}$$

where α determines the importance of the motion clue, Count_{tk} is the number of pixels that are marked as "foreground" in the candidate region corresponding to particle k, and Area_{tk} is the area of the candidate region of particle k. In (37), $(\text{Count}_{\text{tk}})/(\text{Area}_{\text{tk}})$ is used as the confidence level at which there is at least one foreground object (moving object) inside the candidate region. We employ this simple clue $(\text{Count}_{\text{tk}})/(\text{Area}_{\text{tk}})$ from video object segmentation to improve the tracker's performance under background clutter with no extra computation for extracting motion clue. The $W(D_{tk})$ term in (37) is from the histogram comparison results:

$$W(D_{tk}) = e^{-\frac{D_{tk}}{K_t}} \tag{38}$$

where D_{tk} is the result of histogram distance measurements in (36), and K_t is used to control sensitivity to this distance function.

IV. EXPERIMENTAL RESULTS

In this section, the performance of our proposed video object segmentation and tracking framework with the proposed threshold decision algorithm and DD is examined. We first look at the video object segmentation performance then the object tracking performance.

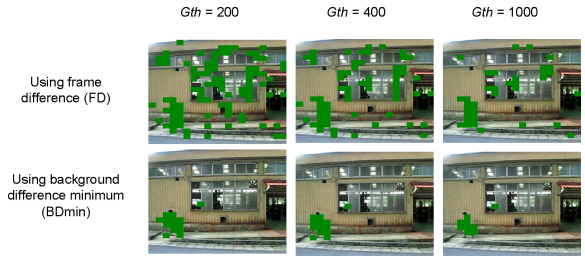


Fig. 9. Results after Gaussianity test. The image blocks detected as foreground regions are shown with green blocks, while the background blocks remain unchanged. It can be seen that the proposed method with BDmin can provide more precise detection results, which can make noise level estimation more stable and robust.

A. Segmentation and Threshold Decision Performance

First, let us look at the performance of the proposed Gaussianity test. Compared with [15], the proposed threshold decision is viable in conditions with dynamic backgrounds. In Fig. 9, the sequence *Fans* with several rotating fans in the background region is employed. The image blocks indicated as foreground after the Gaussianity Test are shown with green blocks, while the background blocks are shown with the original pixel values. This shows that by using frame difference, as in [15], the rotating fans are detected as foregrounds. This problem can get worse if there are many dynamic background regions in the scene leading to not enough samples for noise level estimation and threshold decision. In contrast, our improved method with minimal background difference can still detect foreground objects in the dynamic background, as shown in Fig. 9. More reliable and robust noise level estimation and threshold values can thus be derived with our improved threshold decision algorithm. Moreover, the experimental results in Fig. 9 also show that Gth is actually not a sensitive parameter in this system.

Several test sequences were used to test our proposed segmentation algorithm [13], [30]. The five challenging sequences depicted in Fig. 10 were used for objective evaluation. *IPPR_data1*, *IPPR_data2*, and *IPPR_data3* are challenging test sequences with large noise and compression artifacts [31]; *Bootstrap* and *Darkening* are artificial datasets with dynamic background (waving tree) and changes in illumination [32]. The performance of our segmentation algorithm is shown in Table IV [33], where

$$\text{Recall} = \frac{\#\{\text{Corrected Detected Foreground Pixels}\}}{\#\{\text{Foreground Pixels in Ground Truth}\}} \tag{39}$$

$$\text{Precision} = \frac{\#\{\text{Corrected Detected Foreground Pixels}\}}{\#\{\text{Detected Foreground Pixels}\}} \tag{40}$$

$$F1 = 2 \times \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{41}$$

Note that the threshold was determined using the proposed threshold decision algorithm in this paper. It can be seen that the performance of our algorithm is similar to that of the representative pixel-wised segmentation algorithm with MoG [10], while the memory requirement for our algorithm is much less, as shown in Table I. It proves that our algorithm is indeed a memory-efficient algorithm.

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE PROPOSED SEGMENTATION ALGORITHM (MBREG) AND MIXTURE OF GAUSSIAN (MOG)

Test Sequence	MoG [10]			MBReg		
	Recall	Precision	F1	Recall	Precision	F1
IPPR_data1 [31]	0.602493	0.521882	0.559298	0.685805	0.829457	0.750822
IPPR_data2 [31]	0.664775	0.757593	0.708155	0.595183	0.728440	0.655103
IPPR_data3 [31]	0.716903	0.810027	0.760625	0.772868	0.727437	0.749464

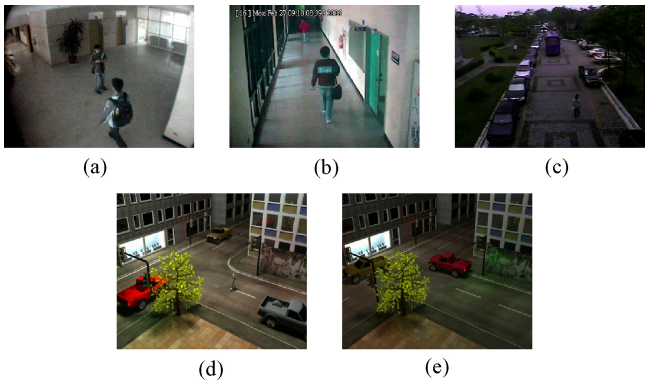


Fig. 10. Test sequences for objective evaluation. (a) *IPPR_data1* [31]. (b) *IPPR_data2* [31]. (c) *IPPR_data3* [31]. (d) *Bootstrap* [32]. (e) *Darkening* [32].

The precision-recall curves (PR curves) for sequences *IPPR_data2*, *Bootstrap* and *Darkening* are shown in Fig. 11(a)–(c), respectively, where MBG-x denotes the proposed MBReg-based algorithm with x-layer background, and MOG-x denotes the MoG-based algorithm with x Gaussians. This shows that the performance of our proposed algorithm with three or more background layers can be similar in performance to that of MoG-based algorithms. When compared with the PR curves in [13], it also shows that the performance of our proposed algorithm is similar to that of the state-of-the-arts, but with much less memory requirements. Moreover, a comparison of Fig. 11(a) and Table IV shows that the proposed threshold decision algorithm can provide an optimal threshold value automatically. Several specially designed video sequences with highly dynamic backgrounds are also provided in [34] for subjective evaluation of the proposed algorithm.

B. Tracking Performance

In this section, we look at the performance of the overall segmentation and tracking framework. First, we test its reliability under changes in illumination, and then, we examine its robustness to background clutter.

In order to test the performances of different histogram distance measurements, the information from segmentation was not included in the weight decision function, i.e., α in (37) was set to 0. The results of comparison of BhD, EMD, and DD are shown in Figs. 12–15 and Tables V and VI, where several specially designed video sequences with drastic changes in illumination and background clutter conditions were tested. Note that only one object existed in all these sequences.

In Fig. 12, the target is a nonrigid object (human) walking through a white lighted region, a dim region, a yellow lighted region, and another dim region in this sequence. The tracker with 3-D color histogram and BhD (first column) almost lost track at (f), while the tracker with 1-D color histogram and BhD (second column) lost track at (d). This indicates that a tracker with BhD tends to lose tracking under these kinds of lighting condition changes. On the other hand, with cross-bin distance, such as EMD and DD, the trackers (fourth column and third column) can adapt to lighting condition changes and track the target correctly. Note that the target model learning scheme is similar to those employed in particle filter frameworks with color histogram as target representation [20], [26].

Similarly, in Figs. 13 and 14, it can be seen that the trackers with EMD and DD outperform those with BhD. In Fig. 14, there are acute changes in illumination. With BhD, the trackers lose tracking along the object trajectory. Even for EMD, there is tracking loss at (e). The only tracker that tracked the target without any track loss was the tracker with 1-D histogram and DD.

The quantitative results of these three sequences and another two sequences are shown in Tables V and VI. In Table V, the performances of these four trackers are evaluated by dividing the number of frames with correct trackers by the total number of frames having the object in ground truth. It was observed that the tracker with 3-D color histogram and BhD lost tracking for two sequences, while the tracker with 1-D color histogram and BhD lost tracking for three sequences. Meanwhile, the tracker with 1-D color histogram and EMD lost tracking for only one sequence, while the tracker with 1-D color histogram and DD successfully tracked all five sequences. In Table VI, the error distance between tracked object center and the center in ground truth is shown. It shows that the tracker with the minimal error distance is either the tracker with DD or the tracker with EMD. From the above results, we can conclude that DD is a better distance metric than the bin-to-bin histogram distance (i.e., the BhD) under changes in illumination. The tracker with DD for color histogram matching can track objects with nonrigid motion robustly under drastic changes in illumination.

In addition to changes in illumination, the tracker may drift due to background clutter. In Fig. 15, by setting α in (37) from 0 to 0.4, the background clutter problem is solved by including motion clue from video object segmentation. (In (g), there is a background clutter problem because the color histogram in the background region is too similar to that of the target.) This shows a simple but efficient method to solve the background clutter problem.

Our proposed tracking system was also tested with more complex cases as shown in Fig. 16, where four more video

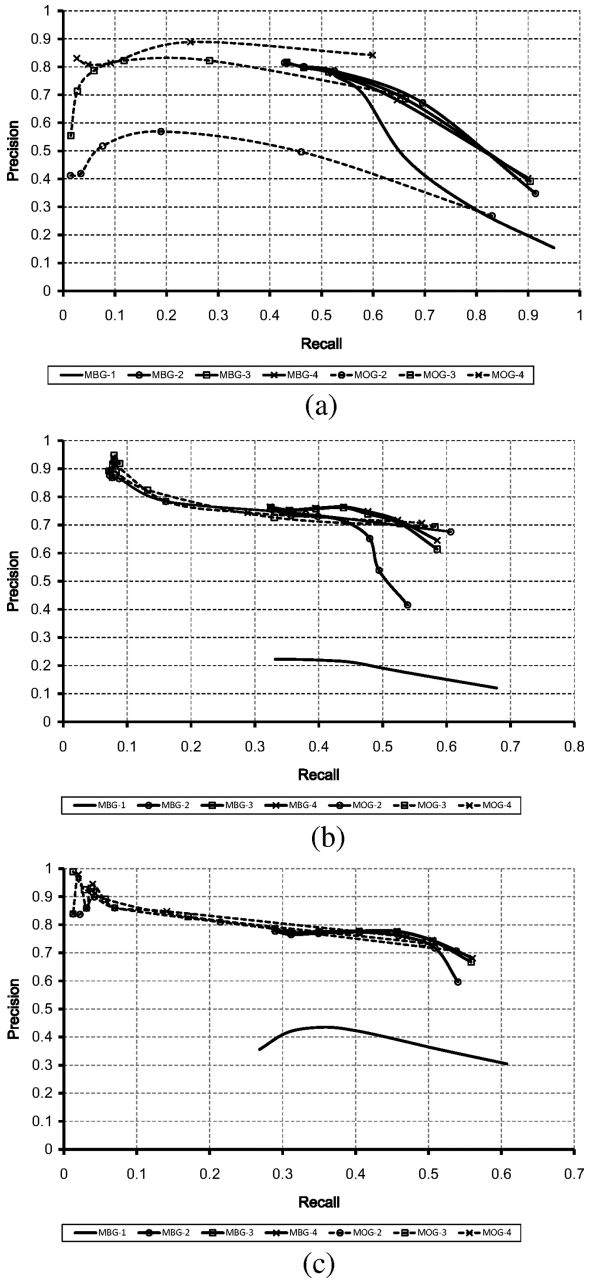


Fig. 11. Precision-Recall curves for three test sequences. (a) *IPPR_data2* [31]. (b) *Bootstrap* [32]. (c) *Darkening* [32].

TABLE V

TRACKING RESULTS FOR SOME TEST SEQUENCES. (FRAMES TRACKED/TOTAL FRAMES WITH OBJECTS)

Color Histogram Distance Measurement	3-D BhD	1-D BhD	1-D DD	1-D EMD
Through Red Light	395/395	242/395	395/395	395/395
Through White Light	58/92	13/92	92/92	66/92
Shadow	440/440	343/440	440/440	440/440
Flicking Light	145/145	78/145	145/145	145/145
Dark Hall	313/334	334/334	334/334	334/334
Average	96%	72%	100%	98%

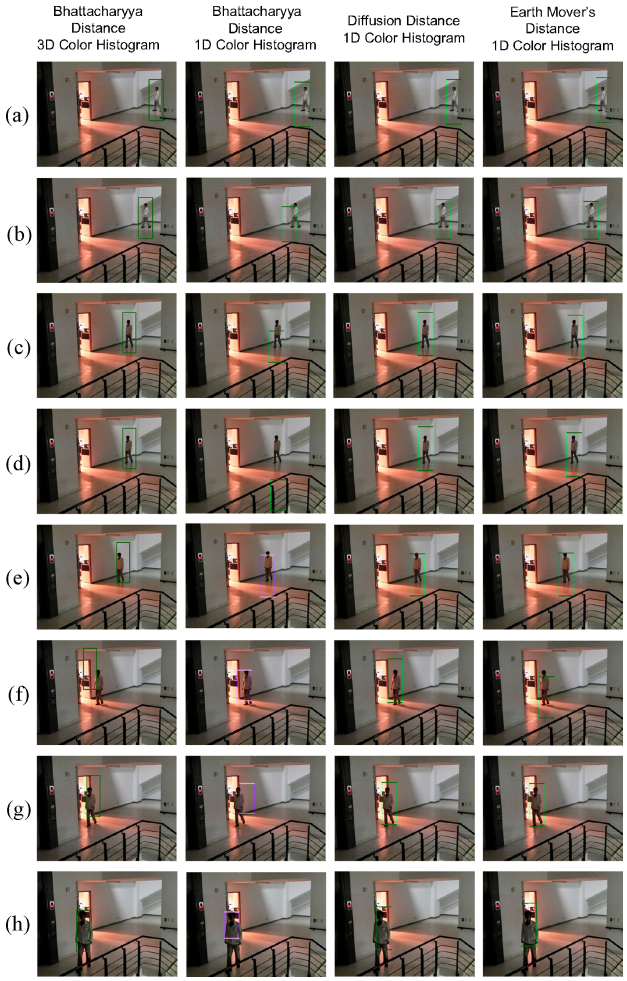


Fig. 12. Comparison of three distance measurements using the sequence *Through Red Light*. The first column is 3-D histogram and BhD, the second column is 1-D histogram and BhD, the third column is 1-D histogram and DD, and the fourth column is 1-D histogram and EMD. (a) Frame no. 8. (b) Frame no. 40. (c) Frame no. 110. (d) Frame no. 112. (e) Frame no. 149. (f) Frame no. 218. (g) Frame no. 249. (h) Frame no. 340.

TABLE VI

POSITION ERRORS FOR SOME TEST SEQUENCES

Color Histogram Distance Measurement	3-D BhD	1-D BhD	1-D DD	1-D EMD
Through Red Light	16.876	124.248	**11.305**	17.685
Through White Light	36.164	107.538	**4.382**	18.371
Shadow	19.722	17.005	10.057	**9.467**
Flicking Light	5.207	11.881	**4.909**	5.809
Dark Hall	23.117	18.439	**7.796**	8.806

clips are provided on our website [34]. Two of the video clips are from the View 007 sequence of PETS 2010 benchmark [35], while the other two video clips were captured from a surveillance camera deployed in our building. The results show that the proposed algorithm can address objects with varying sizes, as shown in Fig. 16(a), and it can also address some object occlusion and crossover cases, as shown in Fig. 16(b). Note that the object maintenance and complex motion model for a complete object tracking system are not included here since the proposed framework is designed to be embedded

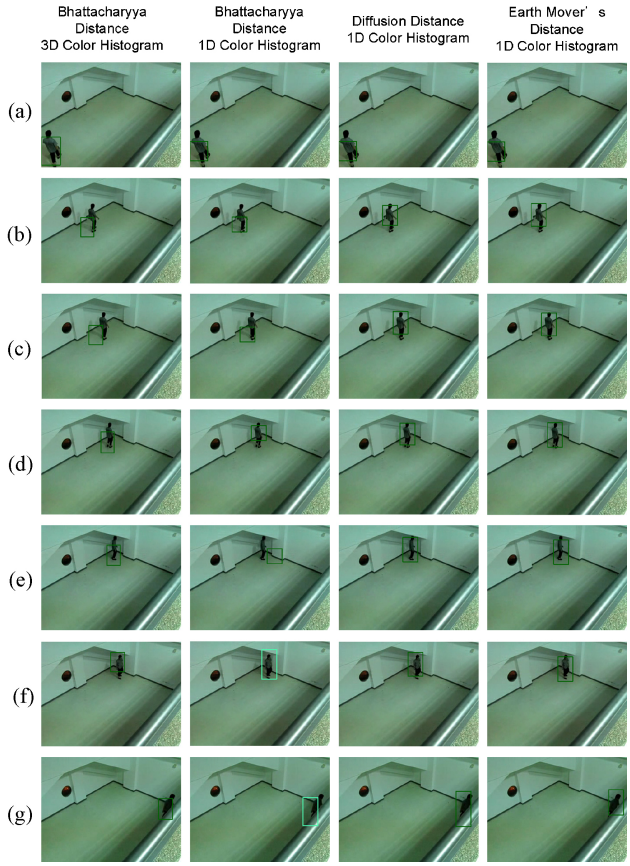


Fig. 13. Comparison of three distance measurements using the sequence *Shadow*. The first column is 3-D histogram and BhD, the second column is 1-D histogram and BhD, the third column is 1-D histogram and DD, and the fourth column is 1-D histogram and EMD. (a) Frame no. 51. (b) Frame no. 167. (c) Frame no. 200. (d) Frame no. 233. (e) Frame no. 266. (f) Frame no. 299. (g) Frame no. 432.

in smart cameras as a preanalysis in a visual surveillance network. The other parts of a complete tracking system can be implemented in the surveillance servers, where information from multiple cameras can also be taken into consideration. Therefore, for the PETS 2010 benchmark, only parts of the sequence were employed and a few objects were tracked to demonstrate the ability of the proposed framework.

Our proposed algorithm was also qualitatively compared with other related tracking algorithms. The methods proposed in [16] and [19] are based on blob matching and Kalman filter. Their performances are highly dependent on the quality of the segmentation results where our proposed object segmentation method can also be applied. The performance of the Kalman filter is not as good as that of particle filter, especially when tracking objects with irregular motions. Gradient descent-based methods [17], [18] employ mean-shift to track objects. They may fail for objects with large motions, such as the sequence in Fig. 14. Moreover, the trackers are easily trapped locally, which may result in loss of tracking for sequences with background clutter conditions, such as the condition in Fig. 15. For other sequences, the performance of the algorithms proposed in [18] should be similar to ours since the distance function employed is differential EMD. The performance of

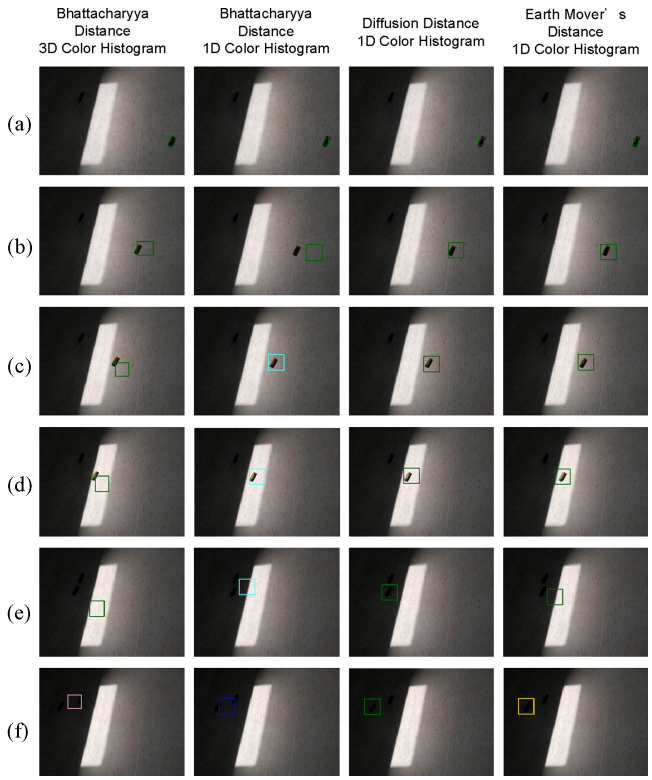


Fig. 14. Comparison of three distance measurements using the sequence *Through White Light*. The first column is 3-D histogram and BhD, the second column is 1-D histogram and BhD, the third column is 1-D histogram and DD, and the fourth column is 1-D histogram and EMD. (a) Frame no. 1. (b) Frame no. 18. (c) Frame no. 35. (d) Frame no. 52. (e) Frame no. 69. (f) Frame no. 86.

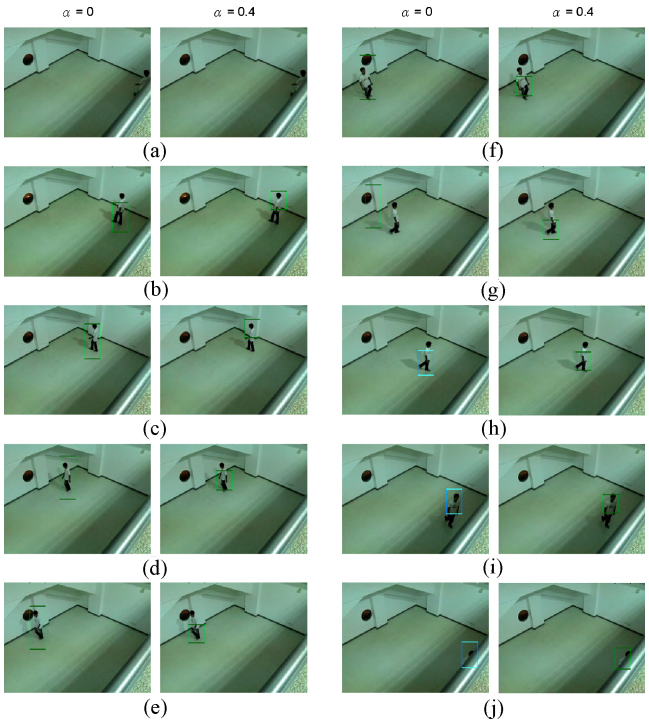


Fig. 15. Tracking results with $\alpha = 0$ (left) and $\alpha = 0.4$ (right). (a) Frame no. 40. (b) Frame no. 100. (c) Frame no. 160. (d) Frame no. 220. (e) Frame no. 280. (f) Frame no. 339. (g) Frame no. 399. (h) Frame no. 460. (i) Frame no. 520. (j) Frame no. 559.

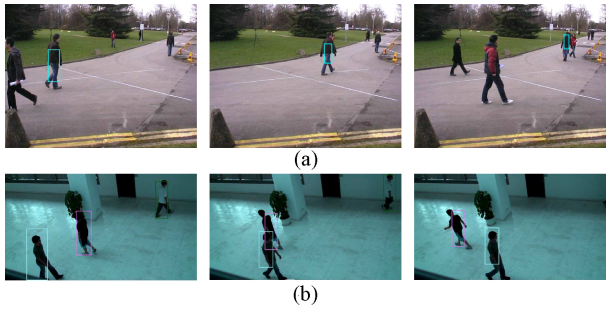


Fig. 16. Tracking results for more complex test cases [34]. (a) Test case with size varying [35]. (b) Test case with object crossover.

the one proposed in [20] should also be similar to ours as it employs the particle filter framework with multiple features. We expect its performance under changes in illumination, such as for the sequence in Fig. 14, to be lower than ours since the color histogram distance function used in [20] is BhD. Please note that the primary focus of our work is to address changes in illumination and background clutter conditions. The framework can accurately track a single object; however, for other issues in object tracking, such as occlusion and foreground clutter, the integration of more mechanisms into the object tracking framework may be required.

V. Conclusion

In this paper, a video object segmentation and video object tracking framework for smart cameras in visual surveillance networks was proposed with two key contributions: threshold decision for multibackground video object segmentation algorithms and DD for measuring color histogram distance. With the proposed threshold decision algorithm, the thresholds for segmentation can be robustly determined for dynamic background conditions without any user input. In addition, it is based on a mechanism that is different from per-pixel background subtraction so as to prevent possible error propagations. Analysis results showed that the segmentation algorithm is memory efficient, and that the memory reduction rate is at least 66% when compared with previous works. For video object tracking, with the information from video object segmentation, the tracker is robust to background clutters. By using DD for color histogram matching, a nonrigid moving object can be robustly tracked even under drastic changes in illumination. From a computational point of view, DD is also ideal as it is most suitable for parallel implementation and hardware realization.

There are several future directions for the proposed framework. First, local thresholds should be effective for some conditions, and the local version of automatic threshold decision can be developed to reduce the required memory size by generating the threshold values on-the-fly. In addition, more powerful postprocessing for the object masks, such as shadow and ghost region removal, can be further integrated into the object segmentation subsystem. In addition, while the primary focus of this paper is to address changes in illumination changing and background clutter conditions, in the future, more mechanisms can be developed to address other issues associated with object tracking, such as occlusion and foreground clutter. More complex particle filters, such as [36] and [37], can also be integrated. Note that when considering the architecture of visual surveillance networks, these mechanisms can be implemented on the server side with our proposed framework on the smart camera side. Furthermore, the proposed illumination-robust object appearance model can also be integrated into some general tracking systems, such as FragTack [38], and online learning-based algorithms, such as MIL [39] and P-N learning [40].

References

[1] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," in *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, Oct. 2008.

[2] C. Regazzoni, V. Ramesh, and G. L. Foresti, "Special issue on video communications, processing, and understanding for third generation surveillance systems," in *Proc. IEEE*, vol. 89, no. 10, pp. 1355–1367, Oct. 2001.

[3] G. Foresti, C. Micheloni, L. Snidaro, P. Remagnino, and T. Ellis, "Active video-based surveillance systems: The low-level image and video processing techniques needed for implementation," *IEEE Signal Process. Mag.*, vol. 22, no. 2, pp. 25–37, Mar. 2005.

[4] W.-K. Chan, J.-Y. Chang, T.-W. Chen, Y.-H. Tseng, and S.-Y. Chien, "Efficient content analysis engine for visual surveillance network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 693–703, May 2009.

[5] T. Horprasert, D. Harwood, and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. IEEE Frame-Rate Appl. Workshop*, Sep. 1999, pp. 1–19.

[6] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Realtime tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[7] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 577–586, Jul. 2002.

[8] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vision*, May 2000, pp. 751–767.

[10] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recogn.*, Jul. 1999, pp. 246–252.

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

[12] W. Wolf, B. Ozer, and T. Lv, "Smart cameras as embedded systems," *IEEE Comput.*, vol. 35, no. 9, pp. 48–53, Sep. 2002.

[13] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2013, pp. 1937–1944.

[14] L. Vosters, C. Shan, and T. Gritti, "Background subtraction under sudden illumination changes," in *Proc. IEEE Int. Conf. Advanced Video Signal Based Surveillance*, Aug. 2010, pp. 384–391.

[15] S.-Y. Chien, Y.-W. Huang, B.-Y. Hsieh, S.-Y. Ma, and L.-G. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," *IEEE Trans. Multimedia*, vol. 6, no. 1, pp. 732–748, Oct. 2004.

[16] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking video objects in cluttered background," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, pp. 575–584, Apr. 2005.

[17] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[18] Q. Zhao, Z. Yang, and H. Tao, "Differential earth mover's distance with its applications to visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 274–287, Feb. 2010.

[19] P. Kumar, S. Ranganath, K. Sengupta, and W. Huang, "Cooperative multi-target tracking with efficient split and merge handling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1477–1490, Dec. 2006.
[20] E. Maggio, F. Smerladi, and A. Cavallaro, "Adaptive multifeature tracking in a particle filtering framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1348–1359, Oct. 2007.
[21] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA: Artech House, 2004.
[22] W.-K. Chan, Y.-H. Tseng, P.-K. Tsung, T.-D. Chuang, Y.-M. Tsai, W.-Y. Chen, L.-G. Chen, and S.-Y. Chien, "ReSSP: A 5.877 TOPS/W reconfigurable smart-camera stream processor," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2011, pp. 1–4.
[23] W.-K. Chan and S.-Y. Chien, "Real-time memory-efficient video object segmentation in dynamic background with multi-background registration technique," in *Proc. IEEE Multimedia Signal Process. Workshop*, Oct. 2007, pp. 219–222.
[24] M. N. Gurcan, Y. Yardimci, and A. E. Cetin, "Influence function based Gaussianity tests for detection of microcalcifications in mammogram images," in *Proc. Int. Conf. Image Process.*, Oct. 1999, pp. 407–411.
[25] M. Muhlich, "Particle filters an overview," in *Proc. Filter-Workshop*, 2003.
[26] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Image Vision Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.
[27] H. Ling and K. Okada, "An efficient earth mover's distance algorithm for robust histogram comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 840–853, May 2007.
[28] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vision*, vol. 40, no. 2, pp. 99–121, 2000.
[29] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit.*, Jun. 2006, pp. 246–253.
[30] IPPR. IPPR: The Chinese image processing and pattern recognition society [Online]. Available: http://www.ippr.org.tw/
[31] IPPR. Dataset of IPPR design contest [Online]. Available: http://archer.ee.nctu.edu.tw/contest/data.htm
[32] S. Brutzer, B. Höferlin, and G. Heidemann, 2011. *Stuttgart Artificial Background Subtraction Dataset* [Online]. Available: http://www.vis.uni-stuttgart.de/index.php?id=sabs
[33] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *Proc. 27th Eur. Conf. Informat. Retrieval*, 2005, pp. 345–359.
[34] S.-Y. Chien, W.-K. Chan, Y.-H. Tseng, and H.-Y. Chen, 2013. *Segmentation and Tracking Results* [Online]. Available: http://media.ee.ntu.edu.tw/research/seg_track/
[35] PETS 2010 benchmark data [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2010/a.html
[36] Z. Khan, T. Balch, and F. Dellaert, "MCMC-based particle filtering for tracking a variable number of interacting targets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1805–1819, Nov. 2005.
[37] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. Int. Conf. Comput. Vision*, Oct. 2009, pp. 1515–1522.
[38] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 1. Jun. 2006, pp. 798–805.
[39] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2009, pp. 983–990.
[40] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Jun. 2010, pp. 49–56.

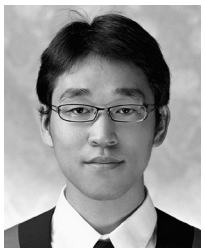
Shao-Yi Chien (S'99–M'04) received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, in 1999 and 2003, respectively.

From 2003 to 2004, he was a member of the Research Staff in Quanta Research Institute, Tao Yuan County, Taiwan. In 2004, he joined the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, as an Assistant Professor, where since 2012, he has been a Professor. His research interests include video segmentation algorithm, intelligent video coding technology, perceptual coding technology, image processing for digital still cameras and display devices, computer graphics, and the associated VLSI and processor architectures. He has published more than 190 papers in these areas.

Dr. Chien serves as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and *Springer Circuits, Systems and Signal Processing*. He also served as a Guest Editor for *Springer Journal of Signal Processing Systems* in 2008. He also serves on the technical program committees of several conferences, such as the IEEE International Symposium on Circuits and Systems, the IEEE International Conference on Multimedia and Expo, the IEEE Workshop on Signal Processing Systems, the Asian Solid-State Circuits Conference, and the International Symposium on VLSI Design, Automation and Test.

Wei-Kai Chan received the B.S. degree in electronics engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 2004, and the Ph.D. degree in the Media IC and System Laboratory, Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, in 2011.

He is currently a Multimedia Engineer in MediaTek Inc., Hsinchu. His research interests include the algorithms and VLSI architectures of computer vision, image signal processing, and multimedia applications.



Yu-Hsiang Tseng was born in Taipei, Taiwan, in 1985. He received the B.S. degree in electrical engineering and the M.S. degree in the Graduate Institute of Electronics Engineering both from National Taiwan University, Taipei, Taiwan, in 2007 and 2009, respectively.

He joined MStar Semiconductor, Inc., Hsinchu, Taiwan, in 2009, where he currently develops integrated circuits related to video coding systems. His major research interests include intelligent video coding technology, computer version, image processing, video object tracking, and associated VLSI architectures.

Hong-Yuh Chen received the B.S. degree in electrical engineering from the National Chiao-Tung University, Hsin-Chu, Taiwan, in 2009. He is currently pursuing the M.S. degree in the Media IC and System Laboratory, Graduate Institute of Electronics Engineering, National Taiwan University.

His current research interests include object segmentation and facial superresolution.