Accurate Foreground Segmentation without Pre-learning

Zhanghui Kuang
Department of Computer Science
University of Hong Kong
Hong Kong, P. R. China
zhkuang@cs.hku.hk

Hao Zhou
Department of Computer Science
University of Hong Kong
Hong Kong, P. R. China
hzhou@cs.hku.hk

Kwan-Yee K. Wong
Department of Computer Science
University of Hong Kong
Hong Kong, P. R. China
kykwong@cs.hku.hk

***Abstract*—Foreground segmentation has been widely used in many computer vision applications. However, most of the existing methods rely on a pre-learned motion or background model, which will increase the burden of users. In this paper, we present an automatic algorithm without pre-learning for segmenting foreground from background based on the fusion of motion, color and contrast information. Motion information is enhanced by a novel method called support edges diffusion (SED) , which is built upon a key observation that edges of the difference image of two adjacent frames only appear in moving regions in most of the cases. Contrasts in background are attenuated while those in foreground are enhanced using gradient of the previous frame and that of the temporal difference. Experiments on many video sequences demonstrate the effectiveness and accuracy of the proposed algorithm. The segmentation results are comparable to those obtained by other state-of-the-art methods that depend on a pre-learned background or a stereo setup.**

***Keywords*-foreground segmentation; contrast attenuation; graph cut;**

I. Introduction

Foreground segmentation plays a key role in a wide variety of computer vision applications, including video surveillance [1], teleconferencing and live background substitution [2]. Although existing methods show that foreground can be extracted successfully from stereo or based on a pre-learned background (i.e., a known background model, or a background model learned from a video without foreground at the beginning) or motion model, they are not so applicable to general situations due to their complex settings or unfriendly initializations. This paper aims at segmenting foreground from monocular videos accurately and efficiently without learning background and/or motion models in advance.

Accurate foreground segmentation without pre-learning is a very challenging problem. It often encounters the following difficulties: (1) textureless or slowly-moving foreground regions may incorrectly be labeled as background (false negatives); (2) occluded background may be misclassified as foreground when it becomes unoccluded (false alarms); (3) changing illuminations, which are common in general application scenarios, often pollute the motion information.

Most of the existing methods employ background subtraction or optical flow to detect motion, and introduce global optimization techniques to obtain a final segmentation [3], [4]. However, they often have difficulties in removing segmentation artifacts.

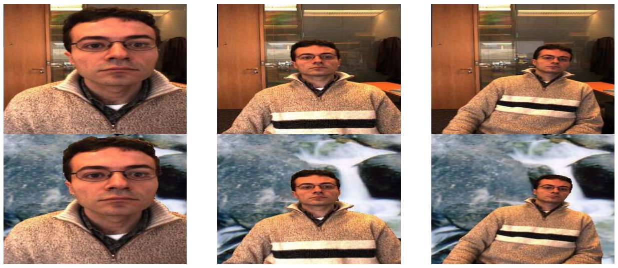

Figure 1. An example of automatic foreground segmentation.

In this paper, we propose a paradigm to segment foreground accurately and efficiently from monocular videos without pre-learning. Figure 1 shows an example of our approach, where the top row illustrates three frames of one input sequence and the bottom row their corresponding foreground and substituted background. Motion and color information are fused to compute a foreground likelihood, which is used with contrast information together to segment the foreground. For each frame of a video, temporal difference between the current frame and the previous frame is evaluated as a motion cue. To enhance the motion cue in textureless or slowly moving regions of foreground, a novel method, named support edges diffusion (SED), is proposed based on a key observation that edges of the temporal difference mostly only appear in moving regions. Histogram of color chromaticity (HCC), which is robust to illumination changes, is used to represent background and foreground models. Motion and color information are combined to obtain an initial foreground likelihood which is refined by a robust foreground rejection scheme based on an incomplete background model learned online. Contrast map is estimated by Canny edge detector and then attenuated based on gradient of the previous frame and that of the temporal difference. Although the proposed foreground segmentation approach simplifies the required setup and does not require learning a background or motion model at the beginning, the segmentation results are comparable to those obtained by other state-of-the-art methods that depend on a pre-learned background or a stereo setup.

II. Related work

Foreground segmentation from videos has long been an active area of research [5]. Conventional approaches for this problem can be roughly classified into two categories based on the criterion whether they need pre-learned models or not.

Approaches with pre-learning. In the compelling work of Criminisi *et al.* [6], an efficient motion *vs* non-motion classifier is trained whose output is then fused with color information. Their algorithm is capable of real-time segmentation of foreground from background in monocular videos. Nevertheless, the classifier needs manually labeled ground truth for training which is not so suitable to general applications. The work of Yin *et al.* [7] requires depth-defined layer labels to train a tree-based classifier. Sun *et al.* [8] proposed "background cut", which achieved a high quality foreground extraction using a single web camera. They combined color and contrast cues with a background model to extract the foreground layer. The task is simplified by learning a background model without foreground at the beginning, which limits the potential application scenarios. For instance, users are often sitting in front of the web camera when they start the video conference application, and it is not possible to learn the background from the video which already contains foreground objects at the beginning.

Approaches without pre-learning. This line of research exploits change detection in video sequences. Chien *et al.* [9] used an accumulated frame difference information to construct a reliable background image and then separated foreground from background region. They elaborated an artifacts removing mechanism which might also degrades segmentation of foreground. Barron *et al.* [10] proposed a motion-based segmentation by estimating optical flow. However, accurate estimation of optical flow is computationally expensive. The most common approach involves "background subtraction". Numerous background subtraction methods, which differ in terms of the background models and rules employed to update the background, were proposed to detect moving foreground [11], [12], [13]. However, background subtraction always generates holes and false alarms, and therefore they are only used as inputs to further high level processes. Postprocessing (such as morphological operations) may attenuate holes or false alarms to a certain extent but tends to lose fidelity near borders of the foreground.

The interesting work of Kolmogorov *et al.* [2] fused color, contrast, and stereo matching information to accurately infer foreground from stereo video sequences. However, as pointed out in [8], this approach has trouble in handling the common situation where only a single web camera is available.

In summary, most of the existing methods with pre-learning can segment foreground accurately and efficiently, while those without learning in advance might not be as accurate or efficient as the former or need complex setups. In this paper, we propose an automatic algorithm without pre-learning to segment foreground accurately and efficiently from monocular videos.

III. Notations and algorithm overview

Consider an input sequence of images with size $m \times n$. An image at time t is represented by $I_t = \{I_t(s) | 1 \leq s \leq mn\}$. The temporal difference image computed by $|I_t(s) - I_{t-1}(s)|$ is denoted by $\Delta I_t = \{\Delta I_t(s) | 1 \leq s \leq mn\}$. For each frame, let V and N be the set of all pixels and all adjacent pixel pairs (4 neighbors), respectively. For the tth frame of a video, M_t denotes the background model, which is learned online. P_t^m and P_t^c are foreground likelihood based on motion information and color information at time t, respectively. $\widehat{P}_t$ is foreground likelihood which is used to segment frame t. $\widehat{C}_t$ denotes contrast map of frame at time t. $\widehat{F}_t$ denotes the segmented foreground of frame at time t.

Our algorithm can be summarized as follows: temporal difference ΔI_t of two adjacent frames is computed and then is mapped to an initial motion likelihood which is enhanced by SED, resulting in foreground likelihood based on motion P_t^m; foreground likelihood based on color P_t^c is computed according to foreground and background color distribution which are represented by HCC; combining P_t^c and P_t^m together with a foreground rejection scheme based on per-pixel background model M_t, we get foreground likelihood $\widehat{P}_t$; contrast $\widehat{C}_t$ is extracted by Canny edge detector and then attenuated based on the previous frame and the temporal difference image; segmentation is then achieved by binary min-cut.

IV. Foreground segmentation

Foreground segmentation can be cast as a binary labeling problem, in which each pixel $I_t(s)$ is assigned a label $X(s) \in \{foreground(=1), background(=0)\}$. The label variables $X = \{X(s) | 1 \leq s \leq mn\}$ can be obtained by minimizing a cost function $E(X)$ [14]:

$$E(X) = \sum_{s \in V} D(X(s)) + \lambda \sum_{(s,r) \in N} B(s,r) \delta(X(s), X(r)) \tag{1}$$

where $\delta(X(s), X(r)) = 1$ if $X(s) \neq X(r)$ otherwise 0.

In (1), $D(X(s))$ is the data term which is the cost when pixel s is labeled as $X(s)$, and $B(s,r)$ is regularization term, which is the cost when the labels of adjacent pixels are different. The coefficient λ (it is set to be 30 in our experiments) specifies the relative importance of the data term and the regularization term. Given foreground likelihood $\widehat{P}_t$ and contrast $\widehat{C}_t$, $D(X(s))$ is defined as follows:

$$\begin{cases} D(X(s)=1) = 1 - \widehat{P}_t(s) \\ D(X(s)=0) = \widehat{P}_t(s) \end{cases} \tag{2}$$

and $B(s,r)$ is given by:

$$B(s,r) = -\frac{\widehat{C}_t(s) + \widehat{C}_t(r)}{2} \tag{3}$$

$B(s,r)$ encourages segmentation along black edges in the negative image of $\widehat{C}_t$.

A. *Motion Cue*

Motion is an important cue in foreground segmentation. Optical flow, which encodes motion information using a dense planar vector field, is commonly employed in motion segmentation [15], [16]. However, it tends to introduce undesirable inaccuracies along boundary of objects and is computationally expensive. In this paper, we use temporal difference [17] to extract motion information.

Consider a pixel s at time t, the probability that s is foreground is given by:

$$P_t^m(s) = T(\log(\max\{\Delta I_t(s), \nu\})/\alpha) \tag{4}$$

where ν is a small constant (we set it to 0.0001) that prevents taking the log of zero and $T(\cdot)$ is a function with its value falling in the range $[0,1]$:

$$T(x) = \begin{cases} 1 & x > 1 \\ 0 & x < 0 \\ x & otherwise \end{cases} \tag{5}$$

The parameter α in (4) is set to be a value in the range $[2,4]$.

Although temporal difference is very adaptive to varying environments, such as illumination changes, it generally does a poor job of extracting the entire relevant feature pixels if the foreground object is textureless or moving slowly [18]. To avoid holes inside moving entities, we propose a new method, called support edges diffusion (SED). It is based on a key observation that support edges (defined as the edges of the temporal difference extracted by Canny edge detector) mostly only appear in moving regions (see figure 2 (d)). Therefore, the neighbor of support edges should be foreground with a high probability.

We begin by detecting edges of the temporal difference ΔI_t, getting support edges $\Gamma = \{s | \Omega(s) = 1, 1 \leq s \leq mn\}$, where $\Omega(s)$ is an indicator such that $\Omega(s) = 1$ if pixel s lies on an edge of the temporal difference or 0 if otherwise. Each pixel s in the set of support edges is associated with a support region $\Phi(s)$, which is usually defined as a circle with center s and radius l (we set $l = 31$ in our experiments). Our goal is to enhance the probability of pixels in the neighbor of support edges being foreground. An additive probability $\Delta P_t^m(s)$ is given by:

$$\Delta P_t^m(s) = \beta \left|\{r | r \in \Gamma \wedge s \in \Phi(r)\}\right| \tag{6}$$

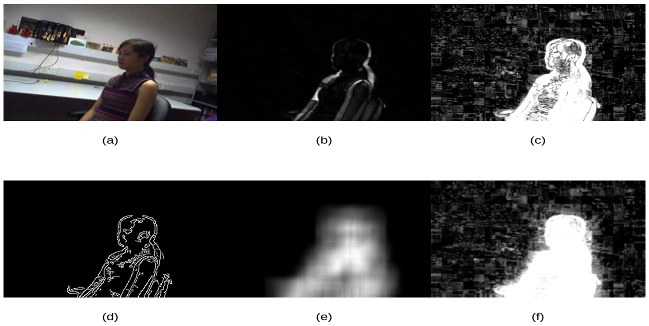

Figure 2. Temporal difference and support edges diffusion. (a) Current frame. (b) Temporal difference computed for the frame in (a). (c) P_t^m computed by (4). (d) Support edges of (b) extracted by Canny edge detector. (e) Additive probability $\Delta P_t^m(s)$. (f) Motion likelihood computed by (7). We can see that SED can fill holes in moving regions.

where $|\cdot|$ denotes the cardinality of a given set. $\beta \in [0,1]$ is a ratio parameter. Obviously, the more pixels from Γ with their support regions covering pixel s, the higher the value of $\Delta P_t^m(s)$ is. Additive probability can be computed efficiently using convolution with an average blur mask. The probability $P_t^m(s)$ is modified to:

$$P_t^m(s) = T(\log(\max\{\Delta I_t(s), \nu\})/\alpha + \Delta P_t^m(s)) \tag{7}$$

Figure 2 shows that support edges have a good ability to distinguish between moving regions and static regions, and SED can augment the foreground probability of pixels in their support regions while keeping others unchanged. In this way, it can greatly reduce the difficulty caused by textureless or slowly moving foreground. Figure 3 compares results using temporal difference alone and temporal difference with SED. Obviously, using temporal difference with SED outperforms using temporal difference alone.

Motion cue of background pixels near the boundary of foreground might also be strong in two ways: (1) large temporal variance will be produced when the occluded background pixels appear; (2) the motion of background pixels near foreground boundary can also be enhanced by SED. Nonetheless, a contrast term based on Canny edges and a foreground rejection scheme based on historic reliable color of background pixels, which we will discuss later, can overcome these problems in most of the cases. Experiments show that foreground can be accurately extracted even with a cluttered background.

B. *Color Cue*

Color was modeled by Gaussian Mixture Models (GMM) in [14], [2], [8]. In [8], the authors employed a GMM-based global color model to describe the foreground, and the background was then described by a linear combination of a GMM-based global color model and a per-pixel color model. Instead of GMM, Criminisi *et al.* [6] introduced two

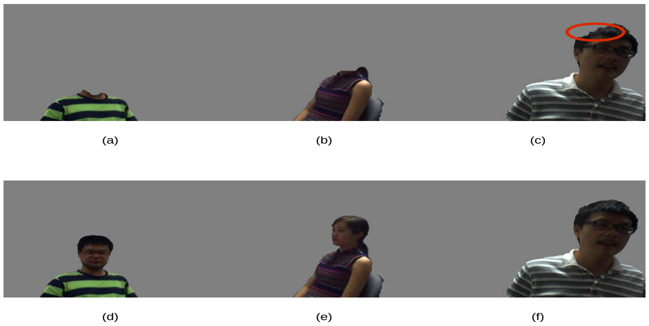

Figure 3. Comparison using temporal difference alone and using temporal difference with SED. (a), (b) and (c) are results using temporal difference alone. (d), (e) and (f) are corresponding results using temporal difference with SED.

3D look-up tables to represent the histograms of foreground and background color, respectively, to avoid issues in the initialization of expectation maximization used to learn GMM.

To reduce the effect of illumination changes, we use two 2D look-up tables to represent the histograms of foreground and background color chromaticity. For each pixel s with three color variables, R, G, and B, color chromaticity coordinates $\widehat{r} = R/(R+G+B)$ and $\widehat{g} = G/(R+G+B)$ are computed [19]. Using histogram of color chromaticity has the advantages of being more insensitive to illumination changes and reducing the number of dimensions of histogram from 3 to 2, which is good for statistics of small samples.

At time $t \geq 1$ (t starts from 0), we have a foreground color distribution $F_t(\widehat{r},\widehat{g})$ and a background color distribution $B_t(\widehat{r},\widehat{g})$. They are learned online according to the previous segmentation. For a particular pixel s with color chromaticity $(\widehat{r},\widehat{g})$, its probability being foreground based on its color is given by:

$$P^c_t(s) = \frac{F_t(\widehat{r},\widehat{g})}{F_t(\widehat{r},\widehat{g}) + B_t(\widehat{r},\widehat{g}) + \varepsilon} \tag{8}$$

where ε is a small constant (we set it to 0.0001) that prevents division by zero. If both $F_t(\widehat{r},\widehat{g})$ and $B_t(\widehat{r},\widehat{g})$ are smaller than 0.001, then we set $P^c_t(s) = 0.5$. After segmentation of each frame, we compute foreground and background color chromaticity distribution $D^f_t(\widehat{r},\widehat{g})$ and $D^b_t(\widehat{r},\widehat{g})$ of this frame, respectively. Foreground and background color chromaticity distributions of the video are then updated as follows:

$$\begin{cases} F_{t+1}(\widehat{r},\widehat{g}) = (1-\rho_f)F_t(\widehat{r},\widehat{g}) + \rho_f D^f_t(\widehat{r},\widehat{g}) \\ B_{t+1}(\widehat{r},\widehat{g}) = (1-\rho_b)B_t(\widehat{r},\widehat{g}) + \rho_b D^b_t(\widehat{r},\widehat{g}) \end{cases} \tag{9}$$

where ρ_f and ρ_b (we set them to 0.1) are learning rates for foreground and background respectively.

After computing $P^c_t(s)$, the probability of pixel s being foreground is given by:

$$P_t(s) = \gamma P^m_t(s) + (1-\gamma)P^c_t(s) \tag{10}$$

where γ ($\gamma \in (0,1)$) is introduced to balance the weights of $P^m_t(s)$ and $P^c_t(s)$.

C. Foreground Rejection Scheme

Different from [8] where each background pixel is represented by a single isotopic color model. We do not fuse per-pixel color model into the probability in (10) because some regions of background can never be learned if they are always covered by the foreground.

A robust foreground rejection scheme is proposed to remove false alarms based on incomplete per-pixel color models learned online, which form an incomplete background image M_t. To avoid accumulated error, we propose a novel scheme to evaluate the reliability of learned per-pixel background models.

Background image M_t at time t is updated after segmenting I_t, as following:

$$M_{t+1}(s) = \begin{cases} (1-\varphi)M_t + \varphi I_t(s) & s \notin \widehat{F}_t \\ M_t(s) & s \in \widehat{F}_t \end{cases} \tag{11}$$

where φ (we set it to 0.5) is a learning rate.

The estimated background image can be used directly to compute the probability of each pixel of the current frame being a foreground. However, this would lead to accumulated error since the background image learned is not reliable. To address this problem, each pixel s is associated with a counter $\Psi(s)$ initialized to zero. The counter increases by one if s is classified as background in each frame and resets to zero otherwise. For the current frame, we compute a difference image $G(s) = |M_t - I_t(s)|$. Based on this difference image and the counters, we can refine foreground likelihood computed by (10) as follows:

$$\widehat{P}_t(s) = \begin{cases} \eta P_t(s) & \Psi(s) \geq \tau \wedge G(s) < \omega \\ P_t(s) & otherwise \end{cases} \tag{12}$$

where τ is a threshold (τ is set to 3). $\Psi(s) \geq \tau$ indicates the background model of s is reliable. $G(s) < \omega$ ($\omega = 10$ in our experiments) denotes s fits its corresponding background model well. η is a decay parameter (it is set to be 0.5).

The intuition behind the foreground rejection scheme is that if a particular pixel s is labeled as background in last τ frames, the background model of s is reliable. Pixel s has a high probability to be background if it fits its own reliable background model well. Figure 4 illustrates the foreground rejection scheme.

D. Contrast Generation and Attenuation

$B(s,r)$ in the energy function is employed to encourage adjacent pixels being assigned with the same labels. [2], [8] defined $B(s,r) \propto exp(-|I_t(r) - I_t(s)|)$ where $(s,r) \in N$. Such a penalty term does not consider the consistence of

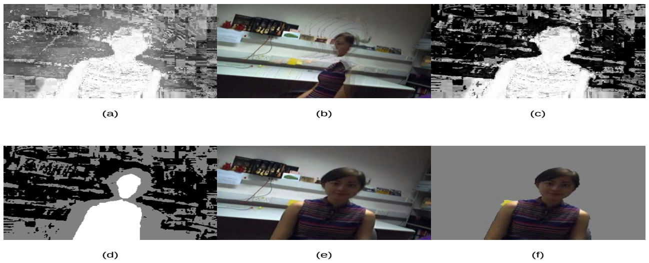

Figure 4. Foreground rejection scheme. (a) Likelihood map based on probability computed by (10). (b) Background model M_t learned online. (c) $\widehat{P}_t$ computed by (12). (d) The region with unreliable background model ($\Psi(s) < \tau$) is marked in white; the foreground rejection region ($\Psi(s) \geq \tau \wedge G(s) < \omega$) is marked in black; the region with reliable background model but do not reject foreground hypothesis ($\Psi(s) \geq \tau \wedge G(s) \geq \omega$) is marked by gray. (e) Segmentation result using (a). All the pixels in the current image is segmented as foreground. (f) Segmentation result using (c). We can see that foreground rejection schema can avoid false alarms.

different adjacent pixel pairs and is sensitive to noise in the image. Instead, we use Canny edge detector to extract image edges and then construct a contrast map. Hence, the energy function can encourage segmentation along the edges of images. To attenuate the edges of the background, we propose a novel attenuate algorithm based on the previous image and the temporal difference.

Let $E_t(s)$ be the edge image. $E_t(s) = 1$ if s is an edge pixel or 0 if otherwise. The penalty term $B(s, r)$ is initially defined as

$$B(s, r) = -\frac{E_t(s) + E_t(r)}{2} \tag{13}$$

Considering an edge image as a contrast map is sufficient to segment foreground accurately in most of the time. However, when a background is cluttered, errors may happen. Since there is no complete background image as given in [8], we cannot use their method to attenuate edges in the background. Instead, edges in background are attenuated based on the previous image and the temporal difference. The previous image I_{t-1} and the temporal difference ΔI_t are convoluted with a Sobel mask [20] to extract their differentiations (gradient), $I^{'}_{t-1}$ and $\Delta I^{'}_t$, respectively. A large value of $I^{'}_{t-1}(s)$ denotes a large variance of I_{t-1} at pixel s, and so does $\Delta I^{'}_t$. The attenuated contrast is given by:

$$\widehat{C}_t(s) = \kappa E_t(s) \frac{\Delta I^{'}_t(s)}{\max\{I^{'}_{t-1}(s), \sigma\}} \tag{14}$$

where κ balances the effect of $I^{'}_{t-1}$ and $\Delta I^{'}_t$ and σ is a small value (we set it to be 0.001) that prevents division by zero.

Intuitively, most of the boundary between background and foreground is changing from frame $t-1$ to t. Therefore, the large variance at pixel s in the previous frame is a good cue to imply s in the current frame does not lie on the boundary of the foreground with a high probability.

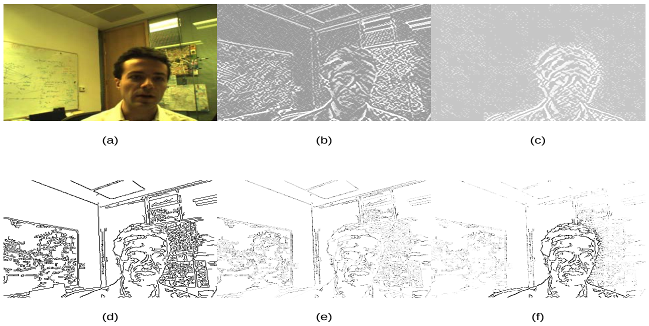

Figure 5. Contrast attenuation. (a) Current frame I_t. (b) Gradient of the previous frame I_{t-1}. (c) Gradient of the temporal difference. (d) Negative image of the contrast of I_t extracted by Canny edge detector. (e) Negative image of the attenuated contrast obtained by $E_t(s)/\max\{I^{'}_{t-1}(s), \sigma\}$. (f) Negative image of the attenuated contrast obtained by (14).

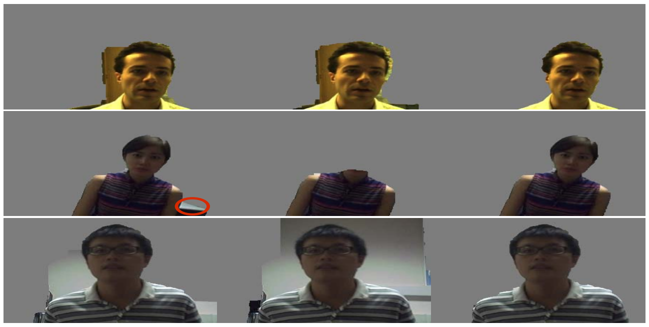

Figure 6. Comparison of segmentation results. The first column is obtained without contrast attenuation; the second column is the results using contrast attenuation based on the previous frame; the third column is obtained with contrast attenuation based on the previous frame and the temporal difference. We can see that contrast attenuation based only on the previous frame can produce false negatives (see the image at center).

$E_t(s)/\max\{I^{'}_{t-1}(s), \sigma\}$ can greatly attenuate the edges in background while preserving the boundary of the foreground. However, when foreground is moving slowly, some boundary of the foreground may remain unchanged, and the simple attenuation may also attenuate the boundary of foreground which is not desirable. Fortunately, the boundary of the foreground is usually accompanied by a large variance of $\Delta I^{'}_t$. We can further augment the boundary of the foreground by the multiplication of $\Delta I^{'}_t$. After getting $\widehat{C}_t$, we can compute $B(s, r)$ according to (3). Figure 5 shows the attenuation procedure. Figure 6 compares segmentation results obtained without contrast attenuation, with simple contrast attenuation and with contrast attenuation.

V. Experimental results

We validated our proposed algorithm on a number of videos which were captured by a Logitech QuickCam Pro 4000 with default settings (auto gain control and auto white balance) and from the benchmark data set [2]. Each

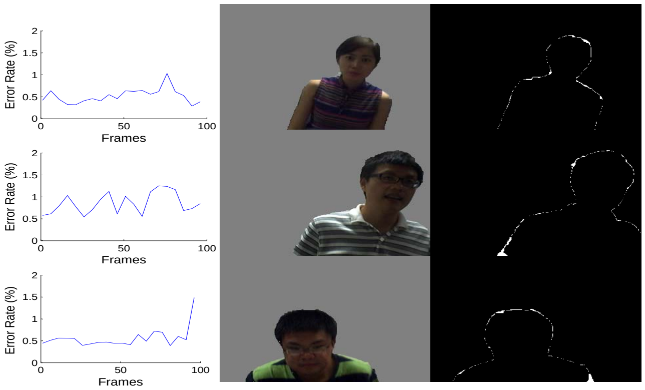

Figure 7. Accuracy of segmentation.

segmentation was initialized by motion cue only or one manually labeling frame at the beginning.

Evaluation of efficiency. The proposed algorithm was coded in C++ and implemented on a desktop PC with 3.00 GHz CPU and 2G RAM. It can segment 5-8 frames per second for a 320×240 video. If multi-scale implementation is employed as [8], the efficiency can be improved further.

Comparison with ground truth. Results of the proposed approach were first compared with hand-labeled ground truth to illustrate its accuracy. Three testing videos ("AY ", "KZ", "ZH") were labeled manually using photoshop's magnetic lasso tool to get an initial foreground and lasso tool to refine the foreground. Performance was evaluated by error rate which was defined as the percentage of misclassified pixels with respect to the image area. Figure 7 provides both objective and subjective measures of segmentation accuracy. The top, middle and bottom rows correspond to testing videos "AY ", "KZ", and "ZH" respectively. The first column is error rate curves. It can be observed that all the three segmentation errors for all three test videos are smaller than 2%. The mean of the error rates for "AY ", "KZ", and "ZH" are 0.517%, 0.858% and 0.563%, respectively. The second column shows the segmentation results of our method at frame 50 for each of the test videos. The third column is the difference between our results with their corresponding ground truth at frame 50 for each of the test videos.

Comparison with other methods. We compared our proposed method with two other state-of-the-art algorithms—"Background cut" [8] and "Bi-layer segmentation" [2] on "AC" video from the benchmark data set [2]. Only the left view of the video was estimated. In figure 8, the red solid curve and the green dot lines are the error rate curve and error bar of our proposed method; the blue dashed lines are the error bar of "Background cut"; the black dashed dot lines are the error bar of "Bi-layer segmentation"; the middle is an example segmentation for "AC"; the right shows the corresponding difference image between our result and ground truth. It can be seen that the accuracy of our method is comparable with "Background cut" and "Bi-layer segmentation" without using any pre-learned background models or stereo setups.

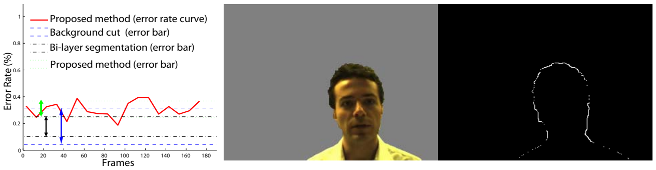

Figure 8. Comparison with other methods.

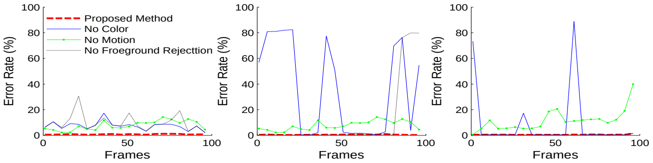

Figure 9. Analysis of the effect of components.

Analysis of the effect of components. Figure 9 analyzes the effect of each component of our proposed method. The left, the middle and the right show the error rates for "AY ", "KZ", and "ZH", respectively. Four cases are compared with respect to error rates: (1) the proposed method; (2) the proposed method without color information; (3) the proposed method without motion information; (4) the proposed method without foreground rejection scheme. It has been shown that if we remove one of the three components, namely color, motion, and foreground rejection scheme, error rates increases considerably. For "AY", the color, motion and foreground rejection scheme almost have the same contribution. For "KZ", without color information or foreground rejection scheme, large error would be resulted. For "ZH", color and motion information dominate the contribution.

Robustness to illumination changes and cluttered background. The test videos were captured by a web camera with automatic gain control and automatic white balance, resulting in large variation of illumination even in static background. Our proposed method can segment them accurately as figure 7 suggests. The underlying reasons are that the motion cue (temporal difference) we use is very adaptive to illumination changes and the foreground rejection scheme can avoid false alarms.

Our method encourages foreground to be segmented along the edges of images. Cluttered background with many edges should be a challenge. However, the attenuation of contrast will suppress this kind of effect in most of the cases. Experiments show that our method can obtained accurate foreground with cluttered background (see figure 10).

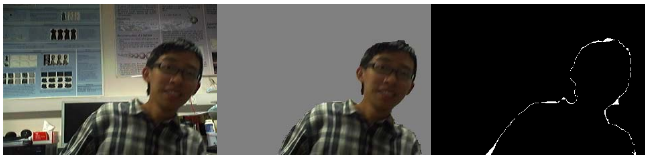

Figure 10. Segmentation with cluttered background. Left: current frame. Middle: segmentation result of the proposed method. Right: the difference between our result with ground truth.

VI. Discussion and Conclusion

In this paper, a robust segmentation approach is proposed to extract foreground from videos accurately and efficiently. This approach is developed based on the fusion of motion, color and contrast information. We employ several mechanisms to avoid segmentation artifacts. Support edges diffusion and foreground rejection scheme are proposed to enhance the foreground likelihood. Novel attenuated contrast cue is used to encourage segmentation along boundary of the foreground. Our approach does not need pre-learned models or complex setups. Experimental results show that our method is comparable to other representative methods and robust to illumination changes.

Our system still has some limitations. First, casual camera shaking may disturb our motion cue, which may be alleviated by detecting camera motion and aligning frames. Second, stationary foreground produces little temporal difference. Magnitude of motion should be monitored and temporal difference should be redefined as difference of two frames with a long interval of time. Last, edges in the constantly occluded background cannot be attenuated when they become unoccluded. More boundary knowledge such as boundary pattern may be used.

Acknowledgment

We would like to thank Dongxia Zhao and Xiaolong Zhu for their assistance with the collection of experimental data.

References

[1] 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.

[2] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, “Bi-layer segmentation of binocular stereo video,” in *Proceedings of CVPR*, 2005, pp. 407–414.

[3] N. R. Howe and A. Deschamps, “Better foreground segmentation through graph cuts,” Tech. Rep., 2004.

[4] T. Schoenemann and D. Cremers, “Near real-time motion segmentation using graph cuts,” in *LNCS*, 2006, pp. 455–464.

[5] J. Bergen, P. Burt, R. Hingorani, and S. Peleg, “A three-frame algorithm for estimating two-component image motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 886–896, 1992.

[6] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, “Bi-layer segmentation of live video,” in *Proceedings of CVPR*, 2006, pp. 53 – 60.

[7] P. Yin, A. Criminisi, J. Winn, and M. Essa, “Tree-based classifiers for bilayer video segmentation,” in *Proceedings of CVPR*, 2007, pp. 1–8.

[8] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum, “Background cut,” in *Proceedings of ECCV*, 2006, pp. 628–641.

[9] C. Shao, M. Shyh, and C. Liang, “Efficient moving object segmentation algorithm using background registration technique,” *IEEE Trans. Circuits Syst. Video Technol*, vol. 12, pp. 577–586, 2002.

[10] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, “Performance of optical flow techniques,” in *Proceedings of CVPR*, 1992, pp. 236 –242.

[11] M. Heikkila and M. Pietikainen, “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.

[12] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Background modeling and subtraction by codebook construction,” in *Proceedings of ICIP*, 2004, pp. 3061–3064.

[13] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings of CVPR*, 1999, pp. 246 –252.

[14] Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images,” in *Proceedings of ICCV*, 2001, pp. 105 –112.

[15] A. Meygret and M. Thonnat, “Segmentation of optical flow and 3d data for the interpretation of mobile objects,” in *Proceedings of ICCV*, 1990, pp. 238 –245.

[16] A. Verri, S. Uras, and E. DeMicheli, “Motion segmentation from optical flow,” in *Proceedings of the Fifth Alvey Vision Conference*, 1989, pp. 209–214.

[17] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa, “A system for video surveillance and monitoring,” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-12, 2000.

[18] L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis,” *Pattern Recognition*, vol. 36, no. 3, pp. 585 –601, 2003.

[19] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151 – 1163, 2002.

[20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed., 2001.