Background Cut

Jian Sun, Weiwei Zhang, Xiaoou Tang, and Heung-Yeung Shum

Microsoft Research Asia, Beijing, P.R. China
{jiansun, weiweiz, xitang, hshum}@microsoft.com

Abstract. In this paper, we introduce *background cut*, a high quality and real-time foreground layer extraction algorithm. From a single video sequence with a moving foreground object and stationary background, our algorithm combines background subtraction, color and contrast cues to extract a foreground layer accurately and efficiently. The key idea in background cut is *background contrast attenuation*, which adaptively attenuates the contrasts in the background while preserving the contrasts across foreground/background boundaries. Our algorithm builds upon a key observation that the contrast (or more precisely, color image gradient) in the background is dissimilar to the contrast across foreground/background boundaries in most cases. Using background cut, the layer extraction errors caused by background clutter can be substantially reduced. Moreover, we present an adaptive mixture model of global and per-pixel background colors to improve the robustness of our system under various background changes. Experimental results of high quality composite video demonstrate the effectiveness of our background cut algorithm.

1 Introduction

Layer extraction [2, 20] has long been a topic of research in computer vision. In recent work [8], Kolmogorov et al. showed that the foreground layer can be very accurately and efficiently (near real time) extracted from a binocular stereo video in a teleconferencing scenario. One application of foreground layer extraction is high quality live background substitution. The success of their approach arises from a probabilistic fusion of multiple cues, i.e, stereo, color, and contrast.

In real visual communication scenario, e.g., teleconferencing or instant messaging, however, most users have only a single web camera. So, can we achieve a similar quality foreground layer extraction using a single web camera? For an arbitrary scene (e.g. non-static background), automatically foreground layer extraction is still a monumental challenge to the current state of the art [21, 23]. To facilitate progress in this area, we address a somewhat constrained but widely useful real world problem in this paper — high quality, real-time foreground extraction (or background removal) from a single camera with a known, stationary background.

To address this problem, the most efficient approach is background subtraction. Background subtraction detects foreground objects as the difference between the current image and the background image. Nevertheless, there are two issues in background subtraction: 1) the threshold in background subtraction is very sensitive to noise and background illuminance changes. A larger threshold detects fewer foreground pixels and vice versa. 2) foreground color and background color might be

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 628–641, 2006.

very similar, resulting in holes in detected foreground object. More sophisticated techniques [7, 22, 1, 6, 17, 16, 14, 11, 12, 18] have been proposed to overcome these problems. But the results are still error-prone and not accurate enough for high quality foreground extraction required in our application because most of these methods make local decisions. Figure 2 (b) shows a background subtraction result. Postprocessing (e.g, morphological operations) may help but cannot produce an accurate and coherent foreground.

Recent interactive image and video segmentation techniques [15, 10, 19, 9] have shown the powerful effectiveness of the color/contrast based model proposed by Boykov et al. [3]. The color/contrast based model considers both color similarity to manually obtained foreground/background color models and contrast (or edge) strength along the segmentation boundary. The final foreground layer is globally determined using the min-cut algorithm. But, as demonstrated in [8], using only color and contrast cues is insufficient.

Therefore, a straightforward solution is to combine the above two techniques - building foreground and background color models from background subtraction and then applying the color/contrast based model. Because the background image is known and stationary, the background color model can be modeled as a mixture of a global color model and a more accurate per-pixel color model, as done in [8] and [19]. This combination can produce a more accurate segmentation result. We refer to this as the basic model.

However, there are still problems in the basic model. Since the basic model considers both color and contrast simultaneously, the final segmentation boundary will inevitably be snapped or attracted to high contrast edges in a cluttered background, more or less as shown in Figure 2 (c). Though this kind of error may be small around the boundary

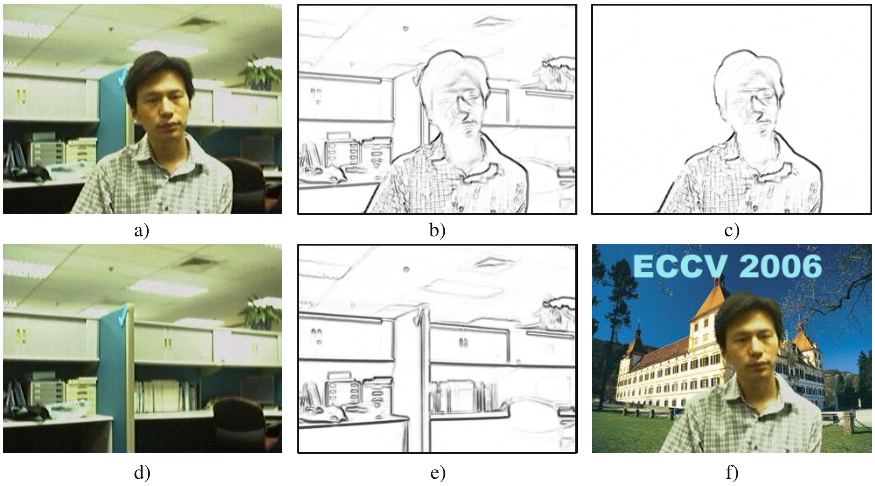


Fig. 1. Background Cut. (a) an image I in a video sequence. (b) contrast map of I. (c) attenuated contrast map by our approach. (d) the background image I^B. (e) contrast map of I^B. (f) our final foreground extraction result using attenuated contrast map.

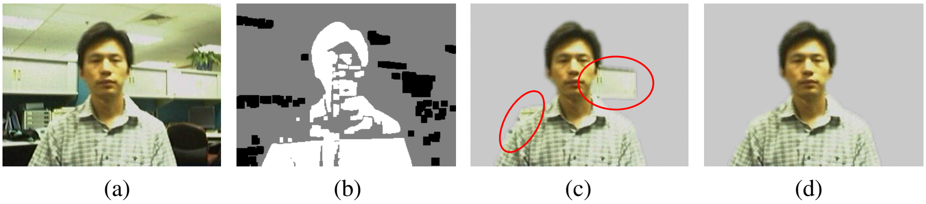


Fig. 2. Foreground layer extraction by different approaches. (a) an image in a video sequence. (b) background subtraction result. Threshold is set to a conservative value to avoid classifying more pixels to foreground. (c) color/contrast-based segmentation result. Red circles indicate notable segmentation errors. (d) our result.

or only occur in partial frames, the flickering artifact in the video due to this error can be very distractive and unpleasant in the final composite video.

In this paper, we propose an new approach, "background cut", to address the above issue in the basic model. The novel component in background cut is "background contrast attenuation" which can substantially reduce the segmentation errors caused by high contrast edges in the clutter background. Background contrast attenuation is based on an key observation that the contrast from background is dissimilar to the contrast caused by foreground/background boundaries in most cases. Figure 1 (b) and (e) show contrast maps of current image and background image respectively. Notice that most contrasts caused by foreground/background boundaries in (b) is not consistent with the contrasts in (e). Based on this observation, background contrast attenuation adaptively modified the contrast map in (b) to produce an attenuated contrast map in (c). Most contrasts from background are removed while contrasts caused by foreground/background boundaries are well preserved. Using this attenuated contrast map, background cut can extract high quality foreground layer from clutter background as shown in (f). Figure 2 (d) also shows that segmentation errors can be significantly reduced in comparison to the basic model.

Another challenge in real scenarios is background maintenance. Many techniques [7, 22, 1, 6, 17, 16, 14, 11, 12, 18] have been proposed to handle various changes in the background, e.g, gradual and sudden illuminance change (light switch in office), small moving objects in the background (e.g, moving curtain), casual camera shaking (e.g, webcame on laptop), sleeping object (an object moves into the background and then becomes motionless), waking object (an object that moves away from the background and reveals new parts of the background), and cast shadows by foreground. To make our system more practical and robust to background changes, we propose a background maintenance scheme based on modeling an adaptive mixture of global and per-pixel background color model.

The paper is organized as follows. In Section 2, we give notations and introduce the basic model. In Section 3, we present our approach - background cut. Background maintenance is described in Section 4 and experimental results are shown in Section 5. Finally, we discuss the limitations of our current approach and give conclusions in Section 6.

2 Notation and Basic Model

Let I^B be the known background image and I be the image at the current timestep that is to be processed. I^B_r and I_r are color values of pixel r in I^B and I respectively. Let $\mathcal{V}$ be the set of all pixels in I and $\mathcal{E}$ be the set of all adjacent pixel pairs (4 neighbors or 8 neighbors) in I. Foreground/background segmentation can be posed as a binary *labeling* problem — to assign a unique label x_r to each pixel $r \in \mathcal{V}$, i.e. $x_r \in \{\text{foreground}(=1), \text{background}(=0)\}$. The labeling variables $X = \{x_r\}$ can be obtained by minimizing a Gibbs energy $E(X)$ [3]:

$$E(X) = \sum_{r \in \mathcal{V}} E_1(x_r) + \lambda \sum_{(r,s) \in \mathcal{E}} E_2(x_r, x_s), \tag{1}$$

where $E_1(x_i)$ is the color term, encoding the cost when the label of pixel r is x_r, and $E_2(x_r, x_s)$ is the contrast term, denoting the cost when the labels of adjacent nodes r and s are x_r and x_s respectively. The parameter λ balances the influences of the two terms.

2.1 Basic Model

Color term. To model the likelihood of each pixel r belonging to foreground or background, a foreground color model $p(I_r|x = 1)$ and a background color model $p(I_r|x = 0)$ are learned from samples. Both models are represented by spatially global Gaussian mixture models (GMMs).

The global background color model $p(I_r|x = 0)$ can be directly learned from the known background image I^B:

$$p(I_r|x = 0) = \sum_{k=1}^{K_b} w_k^b N(I_r|\mu_k^b, \Sigma_k^b), \tag{2}$$

where $N(\cdot)$ is a Gaussian distribution and $(w_k^b, \mu_k^b, \Sigma_k^b)$ represents the weight, the mean color, and the covariance matrix of the kth component of the background GMMs. The typical value of K_b is 10-15 for the background. For stationary background, a per-pixel single isotopic Gaussian distribution $p_B(I_r)$ is also used to model the background color more precisely:

$$p_B(I_r) = N(I_r|\mu_r^B, \Sigma_r^B), \tag{3}$$

where $\mu_r^B = I_r^B$ and $\Sigma_r^B = \sigma_r^2 I$. The per-pixel variance σ_r^2 is learned from a background initialization phase. The per-pixel color model is more precise than the global color model but is sensitive to noise, illuminance change, and small movement of background. The global background color model is less precise but more robust. Therefore, an improved approach is to mix the two models:

$$p_{mix}(I_r) = \alpha \cdot p(I_r|x = 0) + (1 - \alpha) \cdot p_B(x_r) \tag{4}$$

where α is a mixing factor for the global and per-pixel background color models.

The global foreground color model is learned from background subtraction. With a per-pixel background color model, we can mark the pixel that has a very low background probability as “definitely foreground”. Let B, F, U represent “definitely background”, “definitely foreground” and “uncertainty region” respectively, we have:

$$I_r = \begin{cases} B & p_B(I_r) > t_b \\ F & p_B(I_r) < t_f \\ U & otherwises \end{cases}, \tag{5}$$

where t_b and t_f are two thresholds. Then, the global foreground color model $p(I_r|x_r = 1)$ is learned from the pixels in F. In order to enforce temporal coherence, we also sample the pixels from the intersection of F and the labeled foreground region (after segmentation) in the frame at the previous timestep. The component number in the global foreground color model is set to 5 in our experiments because foreground colors are usually simpler than background colors.

Finally, the color term is defined as:

$$E_1(x_r) = \begin{cases} -\log p_{mix}(I_r) & x_r = 0 \\ -\log p(I_r|x_r = 1) & x_r = 1 \end{cases}. \tag{6}$$

Contrast term. For two adjacent pixels r and s, the contrast term $E_2(x_r, x_s)$ between them is defined as:

$$E_2(x_r, x_s) = |x_r - x_s| \cdot \exp(-\beta d_{rs}), \tag{7}$$

where $d_{rs} = ||I_r - I_s||^2$ is the L_2 norm of the color difference, which we call *contrast* in this paper. β is a robust parameter that weights the color contrast, and can be set to $\beta = \left(2\langle\|I_r - I_s\|^2\rangle\right)^{-1}$ [15], where $\langle\cdot\rangle$ is the expectation operator. Note that the factor $|x_r - x_s|$ allows this term to capture the contrast information only along the segmentation boundary. In other words, the contrast term E_2 is the penalty term when adjacent pixels are assigned with different labels. The more similar the colors of the two adjacent pixels are, the larger contrast term E_2 is, and thus the less likely the edge is on the object boundary.

To minimize the energy $E(X)$ in Equation (1), we use the implementation of the min-cut algorithm in [4].

3 Background Cut

The basic model usually produces good results in most frames. However, when the scene contains background clutter, notable segmentation errors around the boundary often occur. This generates flickering artifacts in video. The top row of Figure 3 shows several frames in a video and the third row shows segmentation results by the basic model. Notable segmentation errors are marked by red circles. Why does this happen? The reason is that the basic model contains two terms for both color and contrast. Inevitably, high contrasts (strong edges) from the background will bias the final segmentation result. The second row in Figure 3 shows the corresponding contrast maps[1] of input frames. Notice that most incorrect segmentation boundaries pass strong edges in background. These errors are mainly caused by the contrast term in the basic model:

$$E_2(x_r, x_s) = |x_r - x_s| \cdot \exp(-\beta \cdot d_{rs}). \tag{8}$$

How to fix this bias? More specifically, can we remove or attenuate the contrasts in the background to obtain more accurate segmentation results?

[1] For display, the contrast for each pixel r is computed as $\sqrt{d_{r,r_x} + d_{r,r_y}}$, where r_x and r_y are two adjacent pixels on the left and above pixel r.

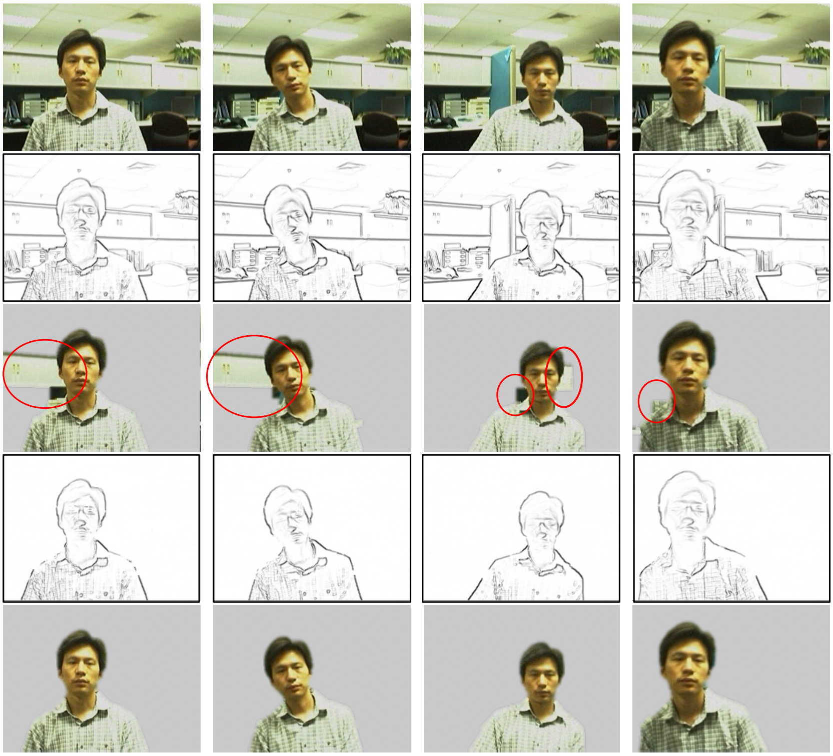


Fig. 3. Background contrast attenuation. Top row: several frames from a video. Second row: contrast maps. Third row: segmentation results by the basic model. Red circles indicate notable segmentation errors. Fourth row: attenuated contrast maps. Last row: segmentation result using attenuated contrast map.

3.1 Background Contrast Attenuation

Because the background is known, a straightforward idea is to subtract the contrast of the background image I^B from the contrast of the current image I. To avoid hard thresholding and motivated by anisotropic diffusion [13], we attenuate the contrast between two adjacent pixels (r, s) in image I from $d_{rs} = ||I_r - I_s||^2$ to d'_{rs} by the contrast $||I^B_r - I^B_s||^2$ in the background image:

$$d'_{rs} = ||I_r - I_s||^2 \cdot \frac{1}{1 + \left(\frac{||I^B_r - I^B_s||}{K} \right)^2}, \tag{9}$$

where K is a constant to control the strength of attenuation. The larger the contrast $||I^B_r - I^B_s||^2$ is in the background, the more attenuation is applied on the contrast $||I_r - I_s||^2$ in image I. Figure 4 (a) and (c) show the contrast maps before and after this soft contrast subtraction. Unfortunately, the contrast caused by the foreground/background

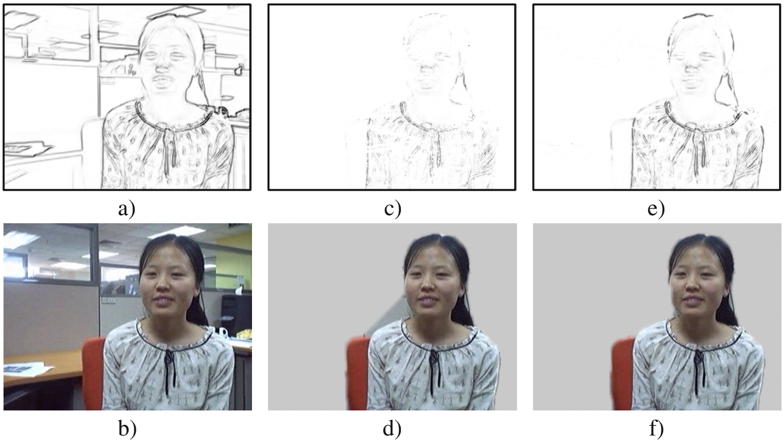


Fig. 4. Adaptive contrast attenuation. (a) contrast map of image I. (b) an image I in a video sequence. (c) and (d) attenuated contrast map and segmentation result using Equation (9). (e) and (f) adaptively attenuated contrast map and segmentation result using Equation (10).

boundary is also attenuated. Figure 4 (d) shows the unsatisfactory segmentation result using this simple subtraction.

In this paper, we propose an adaptive background contrast attenuation method. An ideal attenuation method should attenuate most contrasts in the background and preserve contrasts along the foreground/background boundary simultaneously. To achieve this goal, we define the following method to adaptively preform background contrast attenuation:

$$d''_{rs} = ||I_r - I_s||^2 \cdot \frac{1}{1 + \left(\frac{||I^B_r - I^B_s||}{K}\right)^2 \exp(-\frac{z^2_{rs}}{\sigma_z})}, \tag{10}$$

where z_{rs} measures the dissimilarity between pixel pair (I_r, I_s) in image I and (I^B_r, I^B_s) in background image I^B. A Hausdorff distance-like definition for z_{rs} is:

$$z_{rs} = \max\{||I_r - I^B_r||, ||I_s - I^B_s||\}. \tag{11}$$

If z_{rs} is small, the pixel pair (I_r, I_s) has a high probability of belonging to the background, and the attenuation strength should be large ($\exp(-z^2_{rs}/\sigma_z) \rightarrow 1$). Otherwise, it probably belongs to the contrast caused by the foreground/background boundary, and the attenuation strength should be small ($\exp(-z^2_{rs}/\sigma_z) \rightarrow 0$). Figure 4 (e) shows the contrast map after adaptive background contrast attenuation by Equation (10). Clearly, most contrasts in the background are greatly attenuated and most contrasts along the foreground object boundary are well preserved. Figure 4 (f) shows the corresponding segmentation result. The last two rows of Figure 3 also show the attenuated contrast maps and good segmentation results.

Figure 5 shows attenuation results using different values for parameters K and z_{rs}. Figure 5 (b) shows that a large K will decrease the attenuation strength. A small

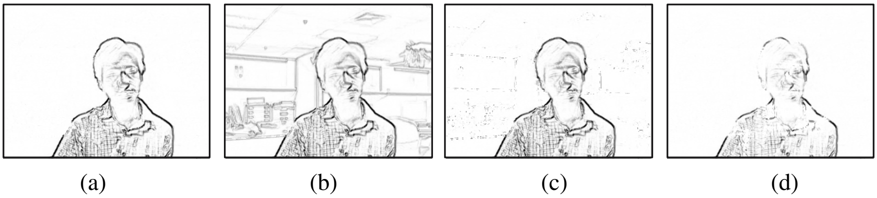

(a) (b) (c) (d)

Fig. 5. Parameter settings. (a) $K = 5, \sigma_z = 10$. (b) $K = 500, \sigma_z = 10$. (c) $K = 5, \sigma_z = 1$. (d) $K = 5, \sigma_z = 50$.

z_{rs} will leave more contrasts in the image (Figure 5 (c)) and vise versa (Figure 5 (d)). In all our experiments, we set the default values of K and z_{rs} to 5 and 10 respectively to obtain good segmentation results on average, as shown in Figure 5 (a). These values are quite stable — there is no notable change in segmentation results when we change K and z_{rs} within the ranges $(2.5, 10)$ and $(5, 20)$ respectively.

This adaptive attenuation method works very well in most cases if there is no large illuminance change in the background image. In order to make our background contrast attenuation more robust, we also propose a measure z_{rs} which is not sensitive to large illuminance change:

$$z_{rs} = \left| \left| \vec{v}(I_r, I_s) - \vec{v}(I_r^B, I_s^B) \right| \right|, \tag{12}$$

where $\vec{v}(a, b)$ is a vector from point a to point b in RGB color space. z_{rs} is illuminance-invariant if we assume the color changes of two adjacent pixels to be the same.

4 Background Maintenance

4.1 Adaptive Mixture of Global and Per-pixel Background Color Model

As mentioned in section 2.1, for the color term, there is a tradeoff between the global background color model (more robust to background change) and the per-pixel background color model (more accurate). In previous works [8] and [19], the mixing factor α in Equation (4) is a fixed value. To maximize robustness, an ideal system should adaptively adjust the mixing factor: if the foreground colors and background colors can be well separated, it should rely more on the global color model such that the whole system is robust to various changes of background; otherwise, it should rely on both the global and per-pixel color models. To achieve this goal, we adaptively mix two models based on the discriminative capabilities of the global foreground and background color models. In this paper, we adopt an approximation of the Kullback-Liebler (KL) divergence between two GMMs models [5]:

$$KL_{fb} = \sum_{k=0}^{K} w_k^f \min_i (KL(N_k^f || N_i^b) + \log \frac{w_k^f}{w_i^b}), \tag{13}$$

where N_k^f and N_i^b are the kth component of foreground GMMs and the ith component of background GMMs respectively. The KL-divergence between N_k^f and N_i^b can be computed analytically. Our adaptive mixture for the background color model is:

$$p'_{mix}(I_r) = \alpha' p(I_r|x=0) + (1-\alpha')p_B(I_r) \tag{14}$$

$$\alpha' = 1 - \frac{1}{2}\exp(-KL_{fb}/\sigma_{KL}), \tag{15}$$

where σ_{KL} is a parameter to control the influence of KL_{fb}. If the foreground and background color can be well separated, i.e., KL_{fb} is large, the mixing factor α' is set to be large to rely more on the global background color model. Otherwise, α' is small (minimum value is 0.5) to use both the global and per-pixel background color models.

4.2 Background Maintenance Scheme

Because visual communication (e.g., video chat) usually last a short period, sudden illuminance change is the main issue to be considered due to auto gain/white-balance control of the camera, illumination by fluorescent lamps (asynchronous with frame capture in the camera), and light switching. In addition, we also consider several background change events, i.e., small movement in background, casual camera shaking, sleeping and waking object. The following is our background maintenance scheme based on the above adaptive mixture of global and per-pixel background color model.

Sudden illuminance change. Illuminance change caused by auto gain/white-balance control of a camera or illumination by a fluorescent lamp is usually a small global change. We adopted histogram specification to adjust the background image globally. After segmentation at each timestep, we compute a histogram transformation function between two histograms from the labeled background regions in I and I^B . Then we apply this transformation to update the whole background image I^B. This simple method works well for small global illuminance or color changes. The large sudden illuminance change is detected by using frame differences. If the difference is above a predefined threshold, we trigger the following process:

Before segmentation: the background image I^B is updated by histogram specification and the global background color model is rebuilt. The foreground threshold t_f is increased to $3t_f$ to avoid introducing incorrect samples. A background uncertainty map $U^B = \{u_r^B = 1\}$ is initialized. The mixture for the background color model is modified as:

$$p'_{mix}(I_r|x=0) = \alpha' p(I_r|x=0) + (1-u_r^B)\cdot(1-\alpha')p_B(I_r). \tag{16}$$

After segmentation: the color, variance, and uncertainty of each pixel in the labeled background region is updated as follows:

$$I^B_{r,t} = (1-\rho)I^B_{r,t} + \rho I_{r,t} \tag{17}$$

$$\sigma^2_{r,t} = (1-\rho)\sigma^2_{r,t} + \rho(I_{r,t} - I^B_{r,t})^T(I_{r,t} - I^B_{r,t}) \tag{18}$$

$$u^B_r = (1-\rho)u^B_r + \rho(1 - \exp(-||I_{r,t} - I^B_{r,t}||/2\sigma^{-2}_{r,t})), \tag{19}$$

where $\rho = \beta N(I_{r,t}|I^B_{r,t}, \sigma^2_{r,t})$ and β (typically 0.2) is the learning rate. Note that the uncertainty of the hidden pixel behind the foreground will never be decreased because we have no information about it.

Movement in background. We handle moving backgrounds using two mechanisms: 1) if the foreground colors and background colors can be well separated, our model will automatically self adjust to rely on the global background color model which is robust to small movements or dynamic motions (e.g., moving curtain) in background. 2) if there is no intersection between a moving object and the foreground, we can keep the biggest connected component in the segmentation result as foreground object. Otherwise, our system will treat the moving object as foreground if there is no higher-level sematic information available.

Sleeping and waking object. Both cases are essentially the same - a sleeping object is a new static object in the background and a waking object reveals new background areas. We should absorb these new pixels into background when they do not intersect with the foreground. After segmentation, the small connected components far from the foreground (largest connected component) are identified as new pixels. If these pixels and their neighboring pixels are labeled as background for a sufficient time period, we trigger background maintenance processing (Equation (17-19)) to absorb these pixels into the background.

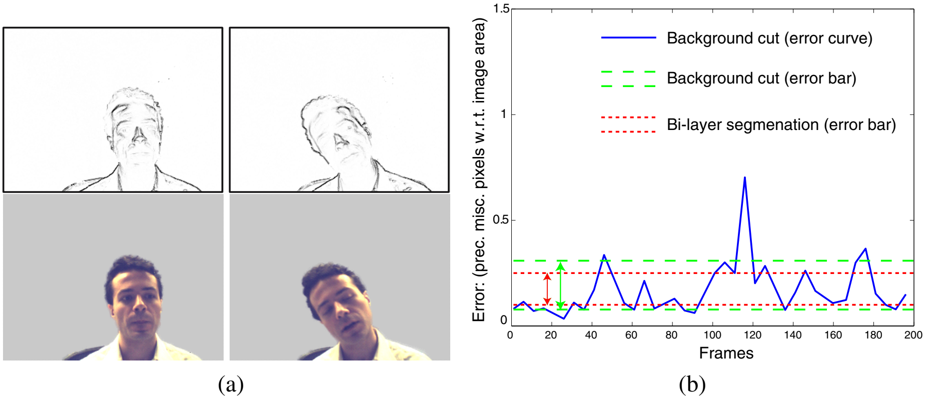
Casual camera shaking. Camera shaking often occurs for a laptop user. We detect camera translation between the current and previous frames. If the translation is small (<4 pixels), a Gaussian blurred (standard variance 2.0) background image is applied and the weight of the per-pixel color model is decreased because global background color model is insensitive to camera shaking. If the translation is large, we disable the per-pixel color model. We will investigate motion compensation in the next step.

We show our background maintenance and segmentation results on the above mentioned background changing cases in the next section.

5 Experimental Results

All videos in our experiments are captured by consumer level web cameras (Logitech QuickCam@ Pro 5000 and Logitech QuickCam@ for Notebooks Deluxe) and we leave all parameters in the web cameras at the default settings (auto gain control and auto white balance). The frame rate is about 12-15 frames/seconds for a 320x240 video on a 3.2GHz desktop PC, with our 2-level multi-scale implementation (the result at the fine level is computed in a narrow band (20 pixels width) around the result at the coarse level). The opacity around the object boundary is obtained by a feathering operation.

Comparison with "Bi-layer segmentation". We quantitatively evaluate the accuracy of our approach on "AC" video which is a stereo video sequence for the evaluation of "Bi-layer segmentation" [8]. The ground truth foreground/background segmentation is provided every 5 frames. The segmentation error is measured as the percentage of bad pixels with respect to the whole image. We only use the video of the left view to test our approach (static background image is obtained by image mosaicing). Figure 6 (a) shows



(a) (b)

Fig. 6. Comparison with “Bi-layer segmentation” on “AC” video. (a) Background cut results (attenuated contrast map and final segmentations). (b) Error statistics. The solid blue line and two green dash lines are error curve and 1 standard error bar of background cut. Two red dotted lines is 1 standard variance error bar of “Bi-layer segmentation”. The original video and ground truth segmentation are obtained from (http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm).

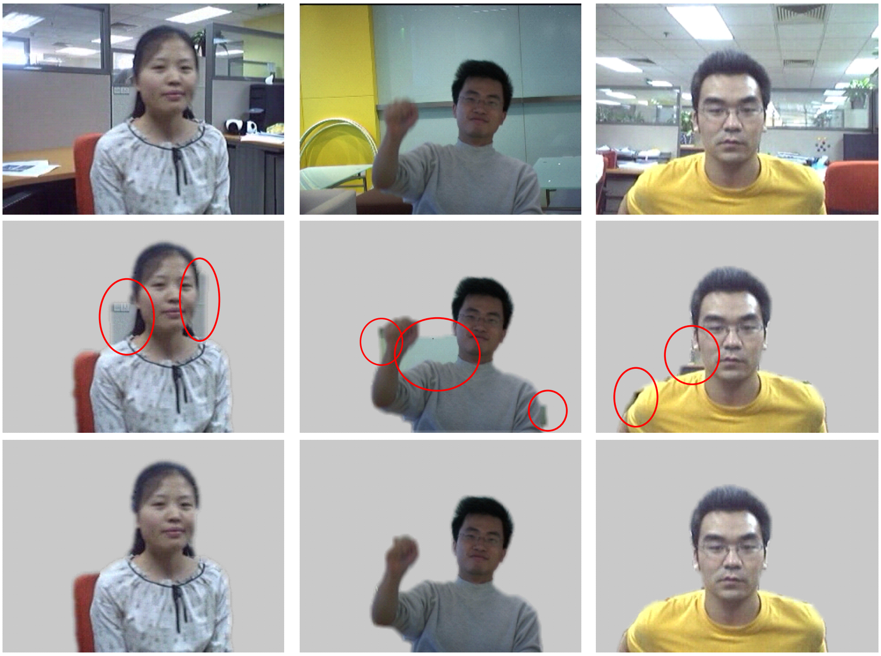


Fig. 7. Comparison with the basic model. Top row: a frame in a video sequence. Second row: result by the basic model. Red circles indicate notable segmentation errors. Last row: result by background cut.

two attenuated contrast maps and segmented foreground layers in the video. Figure 6 (b) plots an error curve (blue solid line) and 1 std error bar (two green dash lines) for our approach, and 1 std error bar (two red dotted lines) for "Bi-layer segmentation". Without using stereo information, the accuracy of our approach is still comparable.

Comparison with "basic model". We compare our approach with the basic model. Figure 7 shows the results produced by the basic model (2nd row) and background cut (last row), respectively. Using the attenuated contrast map, our approach can

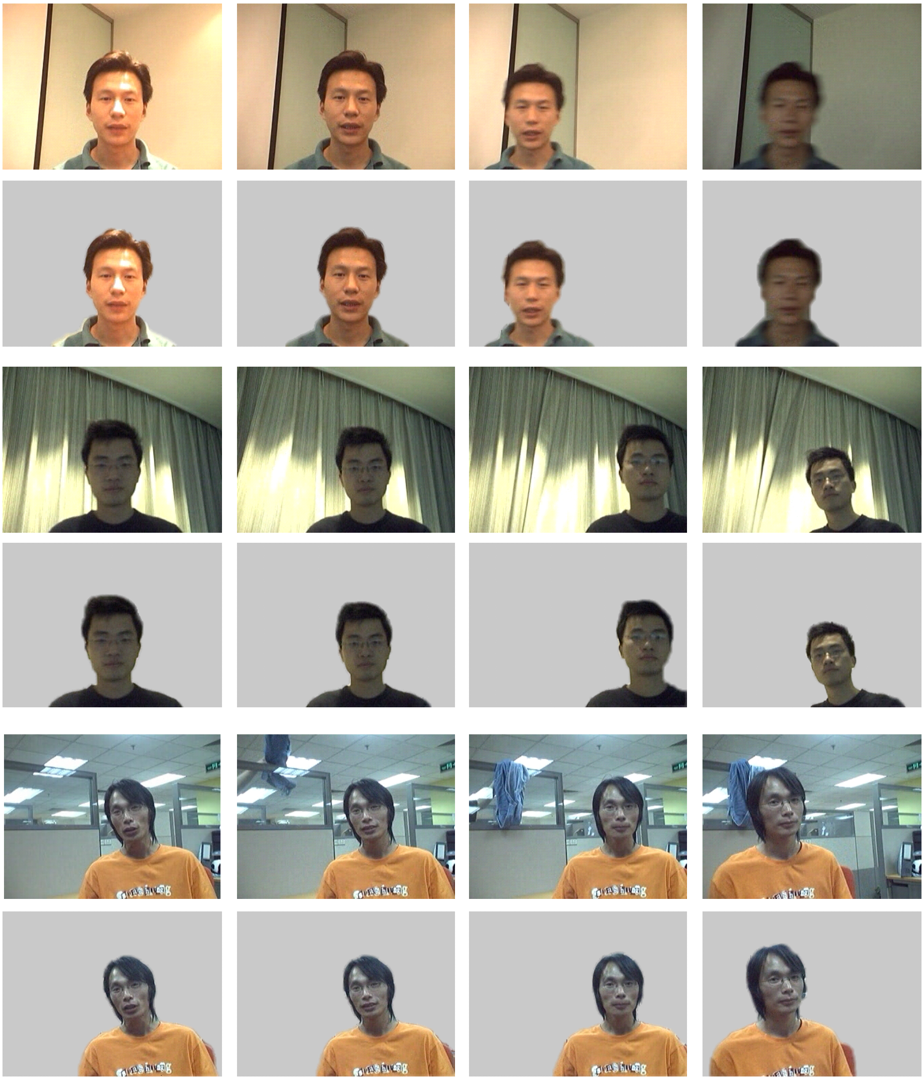


Fig. 8. "Light1", "Curtain", and "Sleeping" examples (from top to bottom). In each example, the upper row shows input images and the lower row shows our segmentation results.

substantially reduce the errors caused by background contrast. Notice that the error of the basic model often results in temporal flickering artifacts around the boundary. For side-by-side comparisons, we highly recommend the reader to view our videos online (http://research.microsoft.com/$\sim$jiansun/).

Background maintenance. Figure 8 shows partial examples to demonstrate our background maintenance scheme. In the "Light1" example, there are two sudden illuminance changes in the 20th frame (first light off) and 181th frame (second light off). The system detected these changes and triggered the background maintenance process. The segmentation results in the 2nd row of Figure 8 shows that good segmentation results can still be obtained during maintenance process. The updated background image sequence is shown in the accompanying video. The "Curtain" example shows a moving curtain in the background. The system adaptively adjusted the mixture of global and per-pixel background color models to handle movements in the background. In the "Sleeping" example, a cloth is put into the background in the 50th frame. Then, it becomes motionless from the 100th frame. The system identified this event and gradually absorbed the cloth into the background. The right most image in the last row of Figure 8 shows correct segmentation when the foreground is interacting with this "sleeping" object. More examples containing sudden illuminance change, casual camera shaking and waking object are shown in our accompanying videos.

6 Discussion and Conclusion

In this paper, we have proposed a high quality, real-time foreground/background layer extraction approach called background cut, which combines background subtraction, color and contrast cues. In background cut, background subtraction is not only done on image color but also on image contrast — *background contrast attenuation* which reduces segmentation errors significantly. Our system is also robust to various background changes in real applications.

The current system still has some limitations. First, when the foreground and background colors are very similar or the foreground object contains very thin structures with respect to image size, high quality segmentation usually is hard to be obtain with our current algorithm. Enforcing more temporal coherence of the foreground boundary may improve the result to a certain extent. Second, in the current system, we assume a static background is obtained in an initialization phase. Automatically initialization of the background image is also important in real applications. Last, we misclassified the moving object which is interacting with the foreground. To solve this ambiguity, high level priors should be integrated into the system.

References

1. D. Harwood A. Elgammal and L. Davis. Non-parametric model for background subtraction. In *Proceedings of ECCV*, pages 751–767, 2000.
2. J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. In *IEEE Trans. on PAMI*, volume 14, pages 886–896, 1992.

3. Y. Boykov and M. Pi. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of ICCV*, pages 105–112, 2001.
4. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Energy Minimization Methods in CVPR*, 2001.
5. J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proceedings of CVPR*, pages 487–494, 2004.
6. W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings of CVPR*, pages 22–29, 1998.
7. D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *Proceedings of ECCV*, pages 189–196, 1993.
8. V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *Proceedings of CVPR*, pages 1186–1193, 2005.
9. Y. Li, J. Sun, and H. Y. Shum. Video object cut and paste. In *Proceedings of ACM SIGGRAPH*, pages 595–600, 2005.
10. Y. Li, J. Sun, C. K. Tang, and H. Y. Shum. Lazy snapping. In *Proceedings of ACM SIGGRAPH*, 2004.
11. A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of CVPR*, pages 302–309, 2004.
12. A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *Proceedings of ICCV*, pages 1305–1312, 2005.
13. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. In *IEEE Tran. on PAMI*, volume 12, pages 629–63, 1990.
14. Y. Ren, C. S. Chua, and Yeong-Khing HO. Motion detection with non-stationary background. In *Machine Vision and Applications.*, pages 332–343, 2003.
15. C. Rother, A. Blake, and V. Kolmogorov. Grabcut - interactive foreground extraction using iterated graph cuts. In *Proceedings of ACM SIGGRAPH*, pages 309–314, 2004.
16. Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *Proceedings of CVPR*, pages 1778–1792, 2005.
17. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Proceedings of ICCV*, pages 255–261, 1999.
18. O. Tuzel, F. Porikli, and Peter Meer. A bayesian approach to background modeling. In *IEEE Workshop on Machine Vision for Intelligent Vehicles*, 2005.
19. J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. In *Proceedings of ACM SIGGRAPH*, pages 585–594, 2005.
20. J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proceedings of CVPR*, pages 361–366, 1993.
21. J. Wills, S. Agarwal, and S. Belongie. What went where. In *Proceedings of CVPR*, pages 37–44, 2003.
22. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Tran. on PAMI*, volume 19, pages 780–785, 1997.
23. J. J. Xiao and M. Shah. Motion layer extraction and alpha matting. In *Proceedings of CVPR*, pages 698–703, 2005.