A ROBUST VIDEO OBJECT SEGMENTATION SCHEME WITH PRESTORED BACKGROUND INFORMATION

Jinhui Pan*[1], Chia-Wen Lin†, Chuang Gu‡, and Ming-Ting Sun*

*Department of Electrical Engineering, University of Washington, USA
†Department of Computer Science & Information Engineering, National Chung Cheng University, Taiwan
‡Microsoft Corporation, USA

ABSTRACT

In this paper, we propose a robust video object extraction algorithm using background subtraction. The algorithm combines two statistical features of the background subtracted images to extract the objects. This method can produce satisfying results with pixel-wise precision. Furthermore, it is robust to the effects of camera noise and the changing of lighting condition. The proposed method is useful in applications with a still background which can be captured and analyzed beforehand, such as virtual conferencing and video surveillance.

1. INTRODUCTION

MPEG-4 video coding standard provides object-based functionalities by introducing a concept of Video Object Plane (VOP). With the extraction of video objects and allocating different number of bits or different frame-rates for different objects, the standard can support object-based scalability that is useful in many practical applications. For example, In remote collaboration, it is desirable to put the collaborators in an immersive environment so that multiple collaborators at remote locations appear as if they are sitting in front of the same table and having a face-to-face meeting [1]. In video surveillance applications, we may like to distinguish and track intruding objects for security purposes [2]. These applications require a video object extraction algorithm in order to segment out the objects of interest for further processing. The video object extraction algorithm can also be used in a videophone system, in which the users may replace the background with an artificial image for privacy concern [3]. It is also useful to support other MPEG-4 applications that use object-based coding features.

There have been many research works on video object segmentation. Those proposed segmentation algorithms could be divided into two categories: interactive and automatic. The interactive algorithms [4-6] require human interaction at least for the segmentation of the initial frame. They are flexible and relatively accurate. However, they are not suitable for real-time applications due to the human interaction required. The automatic algorithms [7-10] attempt to extract video objects without human interactivity. In [8], a moving object extraction algorithm based on moving object edge detection in difference frames and video object tracking using the Hausdorff distance is proposed. In [9], the moving objects are extracted using multiple features, such as motion, color and/or texture to achieve good results. In [10], the foreground is separated from the background based on the motion information. All these algorithms use motion as the main feature to distinguish the foreground from the background.

It is desirable that a video object extraction scheme can be used in general and practical situations. Among all the algorithms mentioned above, interactive algorithms cannot be used in a real-time system without appropriate user interactions, which makes the applications quite limited. Current automatic video object extraction schemes are not promising in general and practical situations. Also, most of the automatic algorithms described above use motion as the main feature to distinguish the foreground from the background. Combined with other features, some of the algorithms can get very precise results. However, the algorithms are not suitable for video conferencing and surveillance applications because in these applications the objects may keep still for a long time. This may lead to a failure of those video object segmentation algorithms. Also, the edges of the moving objects can be rather rough and noisy, making these methods not suitable for applications that require precise edges of the video objects.

Based on the observation that the background of the video conferencing, in most situations, is still and can be captured beforehand, we may use this information to get the precise segmentation of the persons in the conference. However, there are still many annoying problems that can affect the performance of the segmentation algorithm. For example, some parts of the foreground objects may have similar colors as the background. The change of the lighting condition and the noise from the camera may also affect the result of the segmentation. In this paper, in order to get a precise and robust segmentation algorithm, we make use of the two statistical features together to extract the objects. This method can achieve better and more robust results comparing to those algorithms that use only either single statistical feature.

2. VIDEO OBJECT SEGMENTATION USING BACKGROUND SUBTRACTION

Background subtraction is an efficient method to discriminate moving objects from the still background [1-3]. The idea of background subtraction is to subtract the current image from the still background, which is acquired before the objects move in. After subtraction, only non-stationary or new objects

[1] Jinhui Pan is now with the Department of Computer Science, Stanford University, CA, USA (email: jinhuipan@yahoo.com).

0-7803-7448-7/02/$17.00 ©2002 IEEE

are left. This method is especially suitable for video conferencing [1] and surveillance [2] applications, where the backgrounds remain still during the conference or the monitoring time. Nevertheless, there are still many annoying factors such as similar color appearing in both foreground and background areas, changing of lighting condition, and noise which have prevented us from using a simple difference and threshold method to automatically segment the video objects. To overcome these problems and to obtain a robust segmentation algorithm, we propose to utilize two statistics of the background-subtracted images to obtain a more reliable segmentation result as described below.

2.1. Background analysis

The goal of the background analysis is to get a better reference background frame, which has less noise and random lighting change. To achieve this goal, we capture the statistics from the first N background frames (N was set as 10 in our experiments). Two statistical parameters, namely, mean and standard deviation, for each pixel of the background frames can be obtained:

$$\mathbf{bm}_i = \frac{1}{N}\sum_{k=1}^{N}\mathbf{b}_{k,i} \tag{1}$$

$$\mathbf{bv}_i = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{b}_{k,i} - \mathbf{bm}_i)^2 \tag{2}$$

where $\mathbf{b}_{k,i}$ is the ith pixel in the kth background frame. Here, $\mathbf{bm}_i=(bm_{Y,i},bm_{Cb,i},bm_{Cr,i})$ and $\mathbf{bv}_i=(bv_{Y,i},bv_{Cb,i},bv_{Cr,i})$, each contains three components corresponding to the Y, Cb, and Cr components.

The difference frame D can be obtained by the subtraction of the averaged background B from the current frame C by:

$$\mathbf{d}_i = |\,\mathbf{c}_i - \mathbf{bm}_i\,| \tag{3}$$

where $\mathbf{bm}_i$ is taken as in (1); $\mathbf{d}_i \in D$ and $\mathbf{c}_i \in C$, correspond to pixel i in the difference frame and the current frame respectively.

2.2. Segmentation with normalized statistics

It is intuitive that the change caused by a foreground object can be large while the change caused by noise should be small and varies only around the mean value of the corresponding pixel in the background frames. Here, we introduce:

$$\xi_i^n = \frac{|cy_i' - bm_{Y,i}| + k_u \times |cu_i - bm_{Cb,i}| + k_v \times |cv_i - bm_{Cr,i}|}{bv_{Y,i} + bv_{Cb,i} + bv_{Cr,i}} \tag{4}$$

where cy_i' is the luminance value of the ith pixel with background illumination normalization to take into account the effect of automatic gain control used in commercial cameras, ξ_i^n is the normalized statistical feature of pixel i which will be used to classify the pixel i as a foreground pixel or a background pixel (the superscript n is used to indicate it is a normalized feature), k_u and k_v are both empirically set as 1.2. The denominator is the background variance due to camera noises, illumination change, etc. and is used as a normalization factor. Simulation results show that we can model the conditional probability distribution of the normalized statistical feature defined in (4) with a Gaussian distribution:

$$p(\xi_i^n \mid H_l) = \frac{1}{\sqrt{2\pi}\sigma_l} e^{-\frac{(\xi_i^n - m_l)^2}{2\sigma_l^2}} \quad l = 0,1 \tag{5}$$

where m_0 and σ_0 represent the mean value and the standard deviation of ξ^n for the background while m_1 and σ_1 represent the mean value and the standard deviation of ξ^n of the foreground object. Fig. 1 shows the actual distribution of a test video sequence. From Fig. 1, we can see that the simulation result fits the Gaussian model reasonably well. Thus we can make a decision based on the following hypothesis test:

$$p(\xi^n \mid H_0) \times p(H_0) \underset{H_1}{\overset{H_0}{\gtrless}} p(\xi^n \mid H_1) \times p(H_1) \tag{6}$$

Assuming the conditional distribution of the normalized statistical feature is a Gaussian distribution as shown in (5), using the threshold to classify the foreground/background will cause some foreground pixels to be mis-classified as background pixels. If we denote this error probability as P_{err}, then P_{err} can be calculated by:

$$P_{err} = \frac{p(H_1)}{p(H_0)} \int_{-\infty}^{T^n} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-m_1)^2}{2\sigma_1^2}} dx \tag{7}$$

where T^n is the threshold for the classification. Define

$$Q(\alpha) = \int_{\alpha}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \tag{8}$$

$$\text{where} \quad Q(\alpha) \approx e^{-\frac{\alpha^2}{2}} \tag{9}$$

From (7)-(9), it can be shown that:

$$P_{err} \times \frac{p(H_0)}{p(H_1)} = Q\left(\frac{m_1 - T^n}{\sigma_1}\right) \tag{10}$$

So, the threshold is related to P_{err}:

$$T^n = m_1 - \sqrt{-2 \times \sigma_1^2 \times \ln\left(P_{err} \times \frac{p(H_0)}{p(H_1)}\right)} \tag{11}$$

From our simulations, we found that this threshold is slightly different from the optimal value for practical video sequences. The reason is that although the distribution is close to a Gaussian distribution, it is not a strict Gaussian distribution. The distribution of the background temporal statistical feature actually is slightly narrower than the Gaussian distribution and it drops off rather quickly after the peak. Therefore, we control P_{err}, to result in a threshold $T^{n'}$ which is smaller than T^n in (11) and thus closer to the optimal value.

Because the threshold should always be larger than the mean value of the distribution of ξ^n for the background in order to avoid a major part of background pixels being assigned as foreground, we use an adaptive threshold:

$$T^{n*} = \begin{cases} T^{n'} & T^{n'} > m_0 \\ m_0 & otherwise \end{cases} \tag{12}$$

Now, for the final decision, we use:

$$f_i^n = \begin{cases} 1 & \xi_i^n > T^{n*} \\ 0 & otherwise \end{cases} \tag{13}$$

where $f_i^n=1$ means that pixel i belongs to the foreground object, otherwise, it belongs to the background object.

2.3. Segmentation with high-order statistics

It is shown in our previous study [1] that a pixel-wise statistical feature collected within a spatial window centered at each pixel of the difference frame can also be used for object extraction. For this spatially-windowed statistical feature, however, we found that using the difference pixel-value as the feature for classification, the conditional probability of a pixel belongs to the object cannot be modeled with a Gaussian distribution and overlaps a lot with the distribution of the difference caused by noise and lighting change, which make the segmentation difficult. However, we found that if we use the 4-th order variance of the spatially windowed statistical feature for classification, it is much easier to distinguish the background from the foreground object.

We set a 3×3 window centered at each pixel in the difference frame and calculate the 4th-order variance as described in (14) and (15) [8,10],

$$m_i = \frac{1}{M}\sum_{k\in S_i} dy_k \tag{14}$$

$$\sigma_i^4 = \frac{1}{M}\sum_{k\in S_i}(dy_k - m_i)^4 \tag{15}$$

where dy_k is the luminance component of d_k, s_i represents the window centered at pixel i, and M is the number of pixels in the window.

The 4-th order variance σ_i^4 is compared with a threshold T^s:

$$f_i^s = \begin{cases} 1 & \xi_i^s > T^s \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Where, f_i^s = 1 means that pixel i belongs to the foreground object, otherwise, it belongs to the background.

3. SGMENTATION USING COMBINED FEATURES

Fig. 2 shows the proposed segmentation scheme with combined statistical features. After analyzing the background, we then investigate both the normalized and the high-order statistical features of the difference frame D as described above. A decision is made for each feature using the methods mentioned in Section 2. The intermediate results obtained using the two statistical features are then combined to get the final extracted foreground objects.

In general, both the two statistical features can distinguish the foreground object from the background. The advantage of using the normalized statistical feature is that it can reduce the effect of noises and lighting condition with normalization. However, it is difficult to get the exact object boundary without the high-order statistical feature. On the other hand, the advantage of using the high-order statistical feature is that it can extract a better object boundary compared to that using the normalized statistics. Unfortunately, when there are large flat regions or regions with little texture in the current frame, the performance of the high-order statistics-based method can be degraded. The main reason is that, when the regions in the foreground object and background are both flat, the high-order statistics of difference behaviors like noise that is concentrated in a very small zone around the mean value.

Fig. 3 shows two situations in which neither the normalized statistics nor the high-order statistics can provide good results. In Fig. 3(a), the high-order statistics fails when the background is flat, which leads to the failure of collecting the high-order statistics inside the detecting window. In Fig. 3(b), the similarity of the foreground object and the background results in the failure of the normalized statistics. Fortunately, these two methods normally do not fail at the same time. Therefore, we integrate these two intermediate results by an OR operator, which is:

$$f_i = \begin{cases} 1 & f_i^n = 1 \text{ or } f_i^s = 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

After the initial segmentation, post-processing is applied to refine the final segmentation result. First, we extract the largest object region from the segmentation and remove extra regions in the background. Second, we fill in the small holes in the objects to remove noise. Finally, morphological operators are used to refine the boundry.

4. EXPERIMENTAL RESULTS

In our experiments, instead of using standard video sequences, we used video sequences captured by ourselves, which are all in a QCIF (176×144) format. There are two reasons why we used the self-captured sequences for test. First, we need background frames in our algorithm that are not available in most of the standard video sequences. Second, the self-captured sequences can present more complex environment and are noisier than the standard sequences, thereby making them suitable for testing the robustness of the segmentation algorithms in practical situations.

Objective evaluations are performed to compare the performance of the temporal, spatial, and combined segmentation schemes, as well as to choose the optimal thresholds. To obtain the ground truth, we first extracted objects with chromakeying. The extracted objects were then combined with other backgrounds to obtain the composed video sequences. We segmented out the video objects from these composed sequences using the proposed segmentation schemes and then compared the segmentation results with the ground truth. We adopt the objective criterion proposed in [11] to evaluate our segmentation algorithm as follows:

$$d(M_t^{ref}, M_t^{seg}) = 1 - \frac{\sum_{(x,y)} M_t^{ref}(x,y) \oplus M_t^{seg}(x,y)}{\sum_{(x,y)} M_t^{ref}(x,y)} \tag{18}$$

where M_t^{ref} is the reference mask (the "ground truth") and M_t^{seg} is the segmented mask; $\oplus$ denotes the Exclusive-OR (XOR) operation. The segmentation performance measure $d(M_t^{ref}, M_t^{est})$ is equal to or less than 1. The closer the measure to 1, the better the segmentation result. The test results are shown in Figs. 9-12, in which the Y-axis represents $d(M_t^{ref}, M_t^{est})$ of each frame and the X-axis indicates the corresponding frame number of the composed video sequences.

Fig. 4 shows the objective evaluation results on a test video. It shows that the performance of the proposed segmentation schemes is very promising with stable objective evaluation very close to 1 for all the frames simulated. From Fig. 4, the combined scheme, which adopts both two statistical features, significantly outperforms the other two schemes using only a single feature. This is because the high-order statistics is better for accurately extracting the object boundary, while the normalized statistics is better for discriminating the inner part. Therefore combining two statistical features can lead to a more

robust segmentation result. To show the performance of the algorithm in practical scenarios, we also tested it on three natural video sequences, which are with complex backgrounds, with textures in the foreground similar to that in the background, and with loops inside the objects, respectively. The results show our method produces satisfactory segmentation results for various practical situations.

To evaluate the processing speed of the proposed algorithm, we tested our algorithm on a Pentium-III 600MHz PC. The average processing speed is about 9.5 frames/s. Our implementation has not yet been optimized with speed consideration. After optimization, it should be able to achieve higher speed, thereby being suitable for real-time applications.

5. CONCLUSION

In this paper, we proposed a robust object segmentation scheme based on pre-stored background information. By combining two statistical features, our segmentation algorithm can produce very promising segmentation results with low computational complexity. This segmentation algorithm can be performed in real-time for practical applications. Specifically, it's suitable for videophone/conferencing and video surveillance applications where the background information can be easily obtained beforehand.

6. REFERENCES

[1] J. Pan, C. Gu and M.-T. Sun, "An MPEG-4 virtual video conferencing system with robust video object segmentation," in *2nd MPEG-4 Workshop and Exhibition*, San Jose, Jun. 2001.

[2] I. Haritaoglu, D. Harwood, and L.S. Davis. "W4: Who? When? Where? What? A real-time system for detecting and tracking people," *Proc. The third IEEE Int. Conf. Automatic Face and Gesture Recognition*, pp. 222-227, Los Alamitos, CA, 1998.

[3] B. Li and M. I. Sezan, "Adaptive video background replacement," in *Proc. IEEE Int. Conf. Multimedia*, Tokyo, Japan, Aug. 2001.

[4] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuit Circuits Syst. Video Technol.*, vol. 8, no. 5, Sept. 1998.

[5] R. Castango, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia application," *IEEE Trans. Circuit Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 562-571, Sep. 1998.

[6] F. Marques and C. Molina, "Object tracking for content-based functionalities," in *Proc. SPIE Visual Commun. Image Processing*, vol.3024, pp190-199, San Jose, CA, Feb.1997.

[7] M. Kim et al., "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuit Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1216-1226, Dec. 1999.

[8] T. Meier and K. N. Ngan, "Video segmentation for content-based coding," *IEEE Trans. Circuit Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1190-1203, Dec. 1999.

[9] A. A. Alatan, L. Onural, M. Wollborn et al, "Image sequence analysis for emerging interactive multimedia services—the European COST 211 framework," *IEEE Trans. Circuit Circuits Syst. Video Technol*, vol. 8, no. 7, pp. 802-813, Nov. 1998.

[10] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing* (66), pp. 219-232,1998.

[11] M.Wollborn and R. Mech, "Refined procedure for objective evaluation of video object generation algorithms," Doc. ISO/IEC JTC1/SC29/WG11 M3448, Mar. 1998.

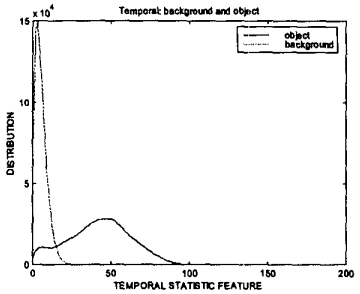

Fig. 1. Normalized statistics distribution.

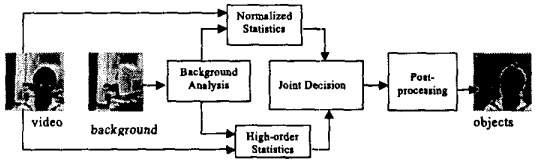

Fig. 2. Proposed object segmentation algorithm.

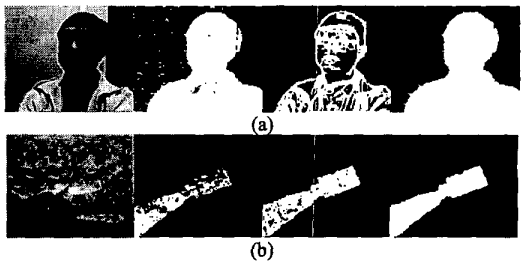

Fig. 3. Examples of segmentation results: Form left to right, each column in (a) and (b) represents: source video, extraction by the nomalized statistics, extraction by the high-order statistics and combination of two statistics features.

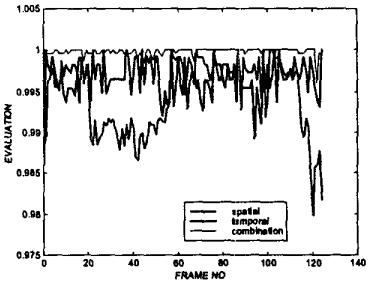

Fig. 4. Objective evaluation.