REALTIME OBJECT EXTRACTION AND TRACKING WITH AN ACTIVE CAMERA USING IMAGE MOSAICS

Chia-Wen Lin
Dept. Computer Science & Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
cwlin@cs.ccu.edu.tw

Chih-Ming Wang, Yao-Jen Chang, and Yung-Chang Chen
Dept. Electrical Engineering
National Tsing Hua University
Hsinchu 300, Taiwan
ycchen@ee.nthu.edu.tw

***Abstract*—Moving object extraction plays a key role in applications such as object-based videoconference, surveillance, and so on. The difficulties of moving object segmentation lie in the fact that physical objects are normally not homogeneous with respect to low-level features and it's usually tough to segment them accurately and efficiently. Object segmentation based on prestored background information has proved to be effective and efficient in several applications such as videophone, video conferencing, and surveillance, etc. The previous works, however, were mainly concentrated on object segmentation with a static camera and in a stationary background. In this paper, we propose a robust and fast segmentation algorithm and a reliable tracking strategy without knowing the shape of the object in advance. The proposed system can real-time extract the foreground from the background and track the moving object with an active (pan-tilt) camera such that the moving object always stays around the center of images.**

Keywords—object segmentation; object tracking; image mosaics; virtual meeting

I. INTRODUCTION

Recently, there has been growing interest in segmentation for object-based video coding. This is mainly due to the development of MPEG-4 standard, which has become a new video coding standard for multimedia communication. MPEG-4 provides many new features to cater for future multimedia applications and to enable object interactivity in video sequences. It also supports an object-based representation of audio-visual objects that allows the access of objects in the compressed domain, selective decoding of such objects and their manipulation. Moving object extraction plays a key role in such kind of applications.

The moving object extraction can be applied to the popular video conferencing environments. In general multipoint video conferencing environments, each participant joins the conference together from separate places. Each participant has its own background, and the conferencing environment looks not concordant at all. This kind of video conferencing environment is quite different from traditional conference. To overcome this disadvantage, we can create a virtual environment and put the segmented objects in it, so the object-based videoconference will look more realistic. The technique can be also applied to home/office surveillance to detect and segment out the intruding objects in the house/office.

In general, video object segmentation is computationally very expensive. As an example, in virtual meeting applications [1], a fully automatic real-time segmentation is needed, which must be able to extract a foreground object even if it is still since the users may keep still for a long time. The segmentation should also have pixel-wise shape accuracy, and be robust against noise and lighting changes. Theses requirements prevent us from using most of the prior segmentation methods [2,3], which may require human interaction, may fail to extract video objects which keep still for a long time, cannot be operated in real-time, or cannot provide pixel-wise shape accuracy.

Automatic video object extraction is a difficult problem in general situations. However, in some application such as video conferencing, surveillance, and studio video, it is relatively easy to pre-capture the background which can be useful for the automatic extraction of the foreground objects. Background subtraction is an efficient method to discriminating moving objects from the still background [1,4]. 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 are left. This method is especially suitable for video conferencing [1] and surveillance [4] applications, where the backgrounds remain still during the conference or 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 camera noise which have prevented us from using a simple difference and threshold method to automatically segment the video objects.

Furthermore, using a static camera may restrict the foreground object to be in a very limited view which is not satisfying in many applications. Conventional background analysis schemes could not be applied easily to images taken from an active camera. Although a few references [5,6] have addressed the problem of object segmentation and tracking using an active camera, they cannot segment the moving object in an arbitrary pan and tilt angle. Therefore, they must store all the views in fixed pan and tilt angles, which is very inefficient and non-flexible. Moreover, in applications such as video conferencing and surveillance, the above operations need to be done in real-time, making computational complexity also a major concern.

In the paper, our aim is to real-time extract and track the foreground human object from the background in a video conference session captured with an active camera such that the moving object always stays around the center of images. Hence, the system requires a robust segmentation algorithm and a reliable tracking strategy.

II. VIDEO OBJECT EXTRACTION AND TRACKING WITH ACTIVE CAMERA

Fig. 1 depicts the conceptual diagram of the proposed object extraction and tracking scheme. Prior to performing segmentation and tracking, a number of background images with equally spaced pan and tile angles are captured and analyzed. A panoramic mosaic image of the background is then automatically constructed from those background images as the reference background model in the segmentation process. After the mosaic image is constructed, the live

video captured from the camera is fed into the detection module. The detection module monitors scene changes and activates the segmentation module when an intruding object is detected. As the segmentation mechanism is activated, the foreground object is extracted from the background and the extracted foreground is utilized as the basis to control the active camera to track the moving object. In addition, the separated background is utilized to update the corresponding background model to improve the segmentation result. In our system, an active camera is used which enables the moving object to be extracted from the background with arbitrary pan and tilt angles, and the object can move in a wide range of background.

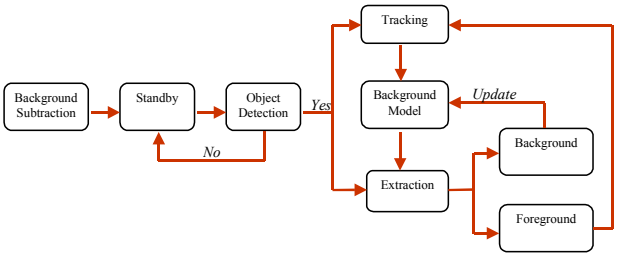


Fig. 1. Conceptual diagram of the proposed object segmentation and tracking scheme.

A. *Reference Bacground Construction Using Imag Mosaics*

The first step to constructing the panoramic mosaic is the alignment of images, i.e., estimate the global motion between the consecutive images. The 2-parameter translation motion model stated in (1) is used for images alignment.

$$\begin{cases} x' = x + a \\ y' = y + b \end{cases} \tag{1}$$

The global motion vector (a^*,b^*) is determined by minimizing a specified error function in the overlapping region as shown in (2).

$$(a^*,b^*) = \arg\min_{(a,b)} \frac{\sum_{(x,y)\in S} [I_1(x+a, y+b) - I_0(x,y)]^2}{\sum_{(x,y)\in S} 1} \tag{2}$$

where I_0 and I_1 are two consecutive background images; S stands for the overlapping region of I_0 and I_1.

Once the global motion of the consecutive frames is obtained, these frames can be integrated into a panoramic mosaic. Pixel blending is used to reduce the discontinuities in color and in luminance. In addition to constructing the panoramic mosaic, our goal is to construct an accurate background model for object extraction. We propose to use an exponential weighting function to blend the overlapping regions, as shown in (3).

$$w_{x,y} = e^{-\left(\frac{(x-x_c)^2}{C_x} + \frac{(y-y_c)^2}{C_y}\right)} \tag{3}$$

where $w_{x,y}$ is the weight at the (x,y) position in the image to be blended and (x_c,y_c) represents the central position of the image. The mosaic image blended with the exponential weighting function is more seamless than that with linear blending functions. Another reason is that the center region in the image is more suitable for background subtraction and the weights around the central regions should be larger than the boundary regions.

The blended pixel value of the mosaic image is computed as follows:

$$M'_{x_m,y_m} = (1-\alpha)\cdot M_{x_m,y_m} + \alpha \cdot f_{x,y} \tag{4}$$

$$\alpha = \frac{w_f(x,y)}{w_m(x_m,y_m) + w_f(x,y)} \tag{5}$$

where M_{x_m,y_m} is the original pixel value in the mosaic image, M'_{x_m,y_m} is the updated pixel value $f_{x,y}$ is the pixel value of the incoming image to be integrated into the mosaic, $w_m(x_m,y_m)$ is the weight at (x_m,y_m) in the mosaic, and $w_f(x,y)$ is the weight at (x,y) in the incoming image.

Fig. 2. Sample mosaic image used as the reference background.

Fig. 2 shows an example of a sub-view in the mosaic image. The panoramic mosaic image is constructed from 15 views taken from equally spaced pan and tilt angle positions. In our method, the mosaic image is to provide an initial rough reference background model for the background subtraction method, and the background model is then refined gradually according to the segmentation result.

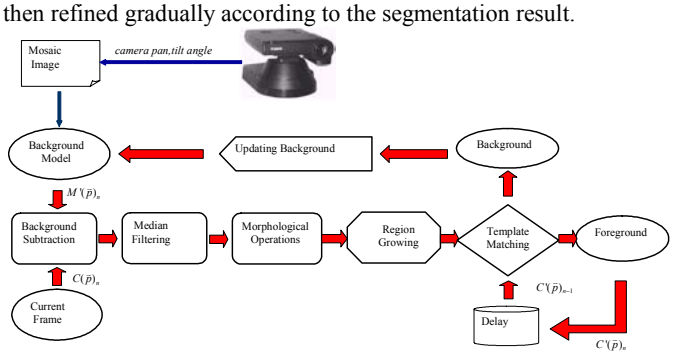


Fig. 3. Procedure of active video object tracking.

B. *Object Classification Using Background Subtraction*

Fig. 3 depicts the detailed procedures of the proposed object extraction and tracking scheme. In constructing the mosaic image, each pixel value in each view may change over a period of time due to camera noises and illumination fluctuations by lighting sources. Therefore, each view is analyzed over several seconds of video, and is then modeled by representing each pixel value with two parameters, mean and standard deviation, during the analyzing period, as follows:

$$\mu(\mathbf{p}) = \frac{1}{N}\sum_{i=1}^{N} R_i(\mathbf{p}) \tag{6}$$

$$\text{std}(\mathbf{p}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} R_i^2(\mathbf{p}) - \mu^2(\mathbf{p})} \tag{7}$$

where $\mathbf{p}$ presents the index of pixels in the pre-captured background frames; $R_i(\mathbf{p})$ is a vector with the luminance and chrominance values of the pixel $\mathbf{p}$ in the i-th background frame; $\mu(\mathbf{p})$ and std($\mathbf{p}$) represent the mean and the standard deviation of the luminance and chrominance values of the pixel $\mathbf{p}$ during the N analyzed background frames in the view. After calculating the background model parameters for each pixel, those different views are then fused into a mosaic image in which the model parameters are blended with (4).

The criterion to classify pixels is described as follows:

$$
\begin{aligned}
&\textit{if } (|C(\mathbf{p}) - \mu_{\mathrm{M}}(\mathbf{p})| > k \times \mathrm{std}_{\mathrm{M}}(\mathbf{p})) \\
&\quad \mathbf{p} \in \textit{foreground} \\
&\textit{else} \\
&\quad \mathbf{p} \in \textit{background}
\end{aligned}
$$

where C(**p**) is the pixel value of position **p** in the current frame to be segmented; $\mu_{\mathrm{M}}(\mathbf{p})$, $\mathrm{std}_{\mathrm{M}}(\mathbf{p})$ are the corresponding mean and the standard deviation of the sub-view in the mosaic background image, respectively; k is a constant used to control the threshold for segmentation. A more sophisticated threshold decision scheme and detailed performance evaluation can be found in [8].

Five steps are used to locate the sub-view in the mosaic image as the corresponding background model:

(1) get the camera pan and tilt angle position from the active camera and use these parameters to roughly locate the sub-view in the mosaic image,
(2) segment and remove the foreground in the current frame by the background subtraction method,
(3) use the remaining background in the current frame to find the more accurate sub-view in the mosaic image,
(4) update the corresponding background model,
(5) iteratively repeat Steps 2 ~ 4 until the corresponding background model is stable.

C. Post-processing

Background subtraction can roughly classify pixels of background and foreground, but the resultant segmentation result may still be quite noisy due to camera noises, illumination variations, and inappropriate threshold selections. Some post-filtering operations are subsequently performed to refine the segmentation result. To mitigate the distortion of the corresponding background model, the binary segmentation result is median filtered with a 3x3 mask, then is further refined with a morphological filter.

At the final step of object discrimination, a region growing method is used to reduce coarse granular noise in the alpha plane. This preserves the region of interest in the alpha plane and erases the other regions considered as a background. We propose two ways to selecting the seed point in the region growing procedure. One is to use "integral projection" proposed in [7] with the alpha plane to obtain the seed point. The other is to calculate the centroid of the skin-color region in the frame as the seed point because the human face is usually the region of interest in our system. In our method, the integral projection method is adopted to obtain the seed point when the area of the skin-color region is less than a threshold, otherwise, the centroid of the skin-color is used.

The separated background in the incoming frame is utilized to update the intensity mean of corresponding background model, $\mu_{\mathrm{M}}(\mathbf{p})$, obtained from the mosaic image. The update mechanism is as follows.

$$
\begin{aligned}
&\textit{If } C(\mathbf{p}) \in \textit{foreground} \\
&\quad \mu'_{\mathrm{M}}(\mathbf{p}) = \mu_{\mathrm{M}}(\mathbf{p}) \\
&\textit{else} \\
&\quad \mu'_{\mathrm{M}}(\mathbf{p}) = (1-\eta)\mu_{\mathrm{M}}(\mathbf{p}) + \eta C(\mathbf{p})
\end{aligned}
$$

Since the standard deviation of each pixel in the mosaic background model usually does not have significant variations during the time, it is not updated. The update mechanism is very effective in improving the segmentation result and it can also resist the slow changes in lighting conditions in the images.

D. Object Tracking with an Active Camera

The proposed segmentation method mentioned above works well when the active camera keeps stationary. However, it may fail to obtain an accurate and robust segmentation when the camera moves while tracking the object, since the corresponding background model becomes inaccurate. The background update mechanism may also fail when the camera moves because the background in the incoming image is changing. In addition, the aforementioned iterative procedure for finding the corresponding background model in the mosaic image will cause delay in the system when the active camera moves.

In our method, as shown in Fig. 3, when camera moves, we get the camera pan and tilt angle position through an RS-232 port and then only use these parameters to roughly locate the sub-view in the mosaic image without iteratively finding the corresponding background model. This can save much computation. To refine the rough segmentation result, a template matching and object tracking method is adopted. Each pixel value of the current extracted object $C_n(\mathbf{p})$ is matched to the corresponding one of the previous extracted object $C_{n-1}(\mathbf{p})$ for reducing the segmentation noises in $C_n(\mathbf{p})$. To reduce the complexity and computation of the tracking, only the centroid of skin-color region in the extracted object is selected. In this way, the proposed strategy achieves robust tracking without any prior knowledge on the object shape.

The correspondence problem can be formulated as a backward matching motion estimation problem, similar to that employed in predictive video compression. The template matching criterion is described as follows:

$$
\begin{aligned}
&\textit{if } \; C_n(\mathbf{p}_1) \in \textit{foreground} \\
&\quad \textit{find } \; C_{n-1}(\mathbf{p}_2), \; \textit{which corresponds to } C_n(\mathbf{p}_1) \\
&\quad \textit{if } \; C_{n-1}(\mathbf{p}_2) \in \textit{foreground} \\
&\qquad C_n(\mathbf{p}_1) \in \textit{foreground} \\
&\quad \textit{else} \\
&\qquad C_n(\mathbf{p}_1) \in \textit{background}
\end{aligned}
$$

In the tracking mode, the active camera is controlled to put the moving object in the central region of the captured image according to the feature point (e.g. the centroid of the skin-color region) selected. A linear trajectory model is used to predict the feature point's future position. Suppose the feature point position is at $\mathbf{p}(t)$ at time t. We can predict the feature point position at time t+1, $\hat{\mathbf{p}}(t+1)$, by assuming a constant velocity $\boldsymbol{v}$ which is obtained from the two previous frames.

$$
\begin{cases}
\boldsymbol{v}(t) = \mathbf{p}(t) - \mathbf{p}(t-1) \\
\hat{\mathbf{p}}(t+1) = \mathbf{p}(t) + \boldsymbol{v}(t)
\end{cases} \tag{8}
$$

Once the feature point leaves the central region, the active camera is re-adjusted. Furthermore the pan-and-tilt speed of the camera is determined by the distance between the predicted feature point, $\hat{\mathbf{p}}(t+1)$, and the center of the central region, $\mathbf{P_c}$.

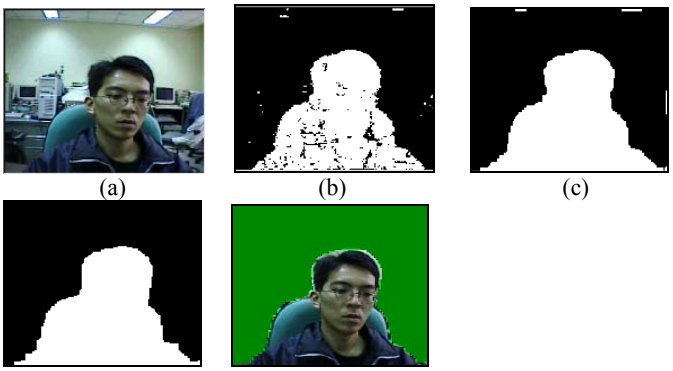
$$
\begin{aligned}
&\textit{if } \; \hat{\mathbf{p}}(t+1) \notin CR \\
&\quad \textit{set } \; \mathbf{CameraSpeed} = \frac{\boldsymbol{S}\left(\mathbf{p}(t+1) - \mathbf{p}_c\right)}{\Delta t}
\end{aligned}
$$

where CR stands for the center region of the captured image with a predefined area, $\boldsymbol{S}$ is a diagonal matrix containing scaling factors for camera tilt and pan motions, and Δt is the temporal interval between the consecutive frames.

III. Experimental Results

Fig. 4 shows the experimental result of the proposed object segmentation scheme. Fig. 4(a) shows a captured image containing the foreground object. Fig. 4(b) depicts the rough segmentation results after performing the background subtraction scheme. The rough segmentation is still quite noisy. The result after applying the

post-filtering is illustrated in Fig. 4(d). The small granular noises can be effectively eliminated using the post-filtering process as shown. Figs. 4(d)-(e) show the final result after region growing.



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

Fig. 4. Simulation result of the proposed object discrimination: (a) the incoming image; (b) alpha plane obtained by background subtraction; (c) alpha plane after post-filtering; (d) alpha plane after region growing; (e) final segmentation result.

Discriminating the skin-color in the foreground region can reduce the computation and avoid the disturbance of skin color in the background. The skin color information is also useful for the tracking and the region growing in our system. The centroid of the skin-color region is utilized as the tracking feature since it varies smoothly and is less complex. Fig. 5 illustrates that the predicted position is very close to the real position, which justifies that the assumptions used in our method are fair approximations.

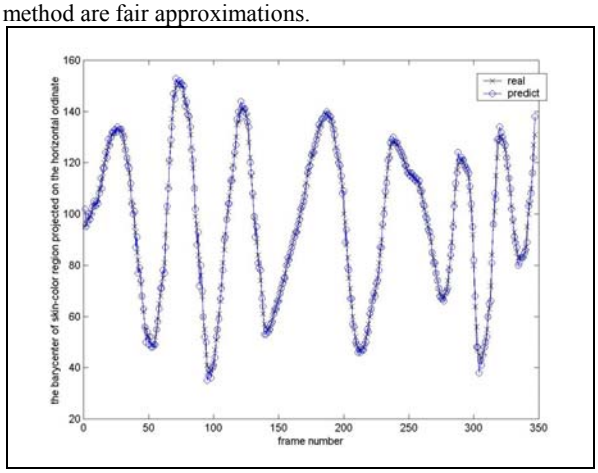


Fig. 5. Predicted centroid of skin-color region in the foreground.

(a) (b) (c)

Fig. 6. (a) segmentation without template matching, (b) previous segmented result used as the template, (c) refined segmentation result after template matching.

Fig. 6 illustrates that template matching can effectively reduce the segmentation noise. However, if the camera continuously moves for a long time, the boundary of the segmentation result may become inaccurate since we use the previous segmented result for tracking the current object which may result in error propagations.

We have implemented the object extraction and tracking algorithm on a Pentium-III 733 MHz PC with a Winnov capture card and a Canon VCC-3 CCD active camera. The processing speed is about 20 QCIF (176x144) frames per second.

IV. Conclusions

We proposed a robust and fast segmentation algorithm and a reliable tracking strategy without the prior knowledge of object shape. The proposed method can extract the object from the background and track the moving object with an active (pan-tilt zoom) camera such that the moving object always stays around the center of images.

Firstly, we introduced how to construct a mosaic image and utilize the mosaic image as the reference background image database in the background subtraction step. Although the sub-view in the mosaic image does not exactly match the background in the current frame captured from the camera, we can segment a rough foreground and reduce the memory cost by using the mosaic image.

The rough segmentation is further refined by performing post-filtering, region growing, adaptive background updating, template matching, and object tracking operations.

In the tracking mode, the centroid of the skin-color region in the foreground is utilized as the feature for detection and tracking with an active camera. The feature is reliable and varies smoothly so that the tracking does not need a complex temporal filter (e.g., a Kalman filter). The proposed system can process about 20 QCIF frames per second on a Pentium-III 733 MHz PC without the need of special-purpose hardware.

The proposed scheme can be used in real-time object-based applications such as MPEG-4 video coding, home surveillance, virtual videophone and video conferencing. We have implemented an H.263 compliant virtual meeting system integrating this work with that in [1] to demonstrate the effectiveness of the proposed schemes as reported in [9].

References

[1] C.-W. Lin, Y.-J. Chang, Y.-C. Chen, and M.-T. Sun, "Implementation of a realtime object-based virtual meeting system," in *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 565-568, August 2001, Tokyo, Japan.

[2] 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, pp. 572-584, September 1998.

[3] 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, September 1998.

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

[5] Y. Ye, J. K. Tsotsos, K. Bennet, and E. Harley, "Tracking a person with pre-recorded image database and a pan, tilt, and zoom camera," in *Proc. IEEE Workshop on Visual Surveillance*, pp.10-17, 1998.

[6] S. Hat, M. Saptharishi, and P. K. Khosla, "Motion detection and segmentation using image mosaics," in *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 1577-1580, July 2000, NY, USA.

[7] C.-W. Lin, Y. -J. Chang, and Y.-C. Chen, "Low-complexity face-assisted video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 207-210, September 2000, Vancouver, BC, Canada.

[8] J. Pan, C.-W. Lin, C. Gu, and M.-T. Sun, "A robust spatio-temporal video object segmentation scheme with prestored background information," in *Proc. IEEE Int. Symp. Circuits and System*, May 2002, Arizona, USA, in press.

[9] C.-W. Lin, Y.-J. Chang, C.-M. Wang, Y.-C. Chen, and M.-T. Sun, "A standard compliant virtual meeting system with cctive video object tracking," *EURASIP Journal on Applied Signal Processing*, June 2002, in press.