EURASIP Journal on Applied Signal Processing 2002:6, 622–634
© 2002 Hindawi Publishing Corporation

A Standard-Compliant Virtual Meeting System with Active Video Object Tracking

Chia-Wen Lin
Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan
Email: cwlin@cs.ccu.edu.tw

Yao-Jen Chang
Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan
Email: kc@benz.ee.nthu.edu.tw

Chih-Ming Wang
Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan
Email: neil@benz.ee.nthu.edu.tw

Yung-Chang Chen
Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan
Email: ycchen@ee.nthu.edu.tw

Ming-Ting Sun
Information Processing Laboratory, University of Washington, Seattle, WA 98195, USA
Email: sun@ee.washington.edu

Received 2 October 2001 and in revised form 24 March 2002

This paper presents an H.323 standard compliant virtual video conferencing system. The proposed system not only serves as a multipoint control unit (MCU) for multipoint connection but also provides a gateway function between the H.323 LAN (local-area network) and the H.324 WAN (wide-area network) users. The proposed virtual video conferencing system provides user-friendly object compositing and manipulation features including 2D video object scaling, repositioning, rotation, and dynamic bit-allocation in a 3D virtual environment. A reliable, and accurate scheme based on background image mosaics is proposed for real-time extracting and tracking foreground video objects from the video captured with an active camera. Chroma-key insertion is used to facilitate video objects extraction and manipulation. We have implemented a prototype of the virtual conference system with an integrated graphical user interface to demonstrate the feasibility of the proposed methods.

Keywords and phrases: video conference, virtual meeting, immersive video conference, networked multimedia, multipoint control unit, MCU, multimedia over IP, object segmentation, object tracking.

1. INTRODUCTION

With the rapid growth of multimedia signal processing and communication, virtual meeting technologies are becoming possible. A virtual meeting environment provides the remote collaborators advanced human-to-computer or even human-to-human interfaces so that scientists, engineers, and businessmen can work and conduct business with each other as if they were working face-to-face in the same environment. The key technologies of virtual meeting can also be used in many applications such as telepresence, remote collaboration, distance learning, electronic commerce, entertainment, Internet gaming, and so forth.

A virtual conferencing prototype, personal presence system (PPS), which provides user presentation control (e.g., scaling, repositioning, etc.) was firstly proposed in [1, 2]. Due to the large computational demand, a powerful dedicated hardware is required for supporting the virtual meeting functionality, thereby leading to a high implementation cost. Recently, several works have been done with a focus on the development of avatar-based virtual conferencing systems. For example, a virtual chat room application, V-Chat [3], has been developed to provide a 3D environment with 2D cartoon-like characters, which is capable of sending text messages and performing some predefined actions.

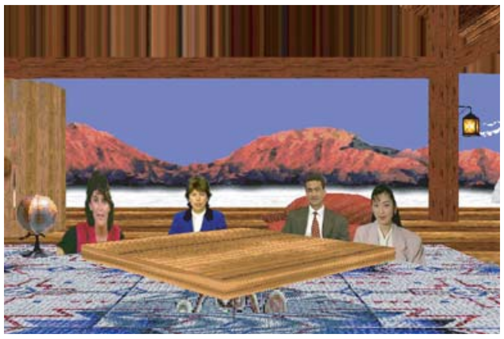


Figure 1: A virtual meeting with 2D video objects manipulated against a 3D virtual environment.

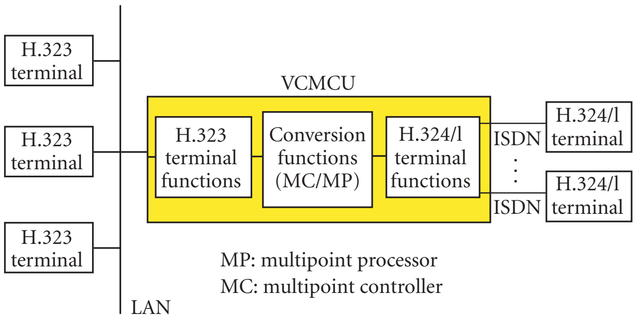


Figure 2: The conceptual model of the proposed VCMCU.

While the 2D avatar of V-Chat provides acceptable representation, several research works have been conducted on seeking for 3D avatar solutions such as the virtual space teleconferencing system proposed by Ohya et al. [4], the virtual life network (VLNet) with life-like virtual humans proposed by Çapin et al. [5], and the networked intelligent collaborative environment (NetICE) with an immersive visual and aural environment and speech-driven avatars proposed by Leung et al. [6]. Although the synthetic avatar-based solutions usually consume a small bandwidth, they may not provide satisfactory viewing quality due to the immature technologies to date.

Hence, an efficient alternative is to adopt natural video instead of synthesizing facial expressions on a 3D avatar. In the FreeWalk system developed by Nakanishi et al. [7], the avatar is represented by a pyramid of 3D polygons with user's video attached on one flat rectangular face. A 3D environment is also provided, where users can freely navigate to perform casual meetings. The InterSpace system proposed in [8, 9] also provides similar functionalities with avatars containing a 3D body with a synthesized computer monitor showing user's video as the head of the 3D body. The use of natural video indeed increases the viewing quality and is more suitable for real-time communications. However, the direct use of the natural video containing the background from real world of different users would degrade the sense of immersion. It is more preferable to have the background removed. Figure 1 illustrates a scenario of virtual video conferencing which involves 2D natural video objects (from the remote conferees) manipulated in a synthetic 3D virtual environment. In order to make this possible, several key technologies need to be incorporated. For example, we need to be able to segment the conferees involved and compose them together into a single scene in real-time so that these video objects appear interacting in the same environment.

To facilitate audio and video communications with existing coding standards, we adopt ITU-T H.323 multimedia communication standard [10] as the basic system architecture. ITU-T H.323 is the most widely adopted standard to provide audiovisual communications over LANs. H.323 can be used in any packet-switched network, regardless of the ultimate physical layer. H.323 includes many mandatory or optional component standards and protocols such as audio codec (G.711/G.722/G.723.1/G.728/G.729), video codec (H.261/263), data conferencing protocol (T.120), call signaling, media packet formatting and synchronization protocol (H.225.0), and system control protocol (H.245) which defines a message syntax and a set of protocols to exchange multimedia messages.

In this paper, we present a low-cost virtual meeting system which fully conforms to the H.323 standard. We propose a virtual conference multipoint control unit (VCMCU) which adopts the H.263 [11] video coding standard. The proposed virtual meeting system involves 2D natural video objects and 3D synthetic environment. We propose a realtime, reliable, and accurate scheme for extracting and tracking foreground video objects from the video captured with an active camera (i.e., a camera with a certain range of pan, tilt, and zoom-in/out control). After object segmentation, we use chroma-key-based schemes so that the developed techniques can be adopted in H.263 compatible systems without sending the video object shape information. The concept can also be easily extended to MPEG-4 based systems where the shape information is already included in the coded video data.

The rest of this paper is organized as follows. In Section 2, we discuss the architecture of the proposed virtual video conferencing system. Section 3 describes an active video object extraction and tracking scheme based on background mosaicking for real-time segmentation. Section 4 presents the video object compositing and manipulation scheme in the proposed system. Finally, conclusions are given in Section 5.

2. PROPOSED SYSTEM ARCHITECTURE

The proposed VCMCU is a PC-based prototype which performs the multimedia communications and protocol translations among multiple LAN and WAN terminals. The conceptual model of the proposed VCMCU is shown in Figure 2. The LAN users can establish a multipoint video conference session with the WAN (ISDN) users via the VCMCU. Typically a gateway is used for point-to-point communication. When there are more than two endpoints to hold a conference, an MCU is required to control and manage the

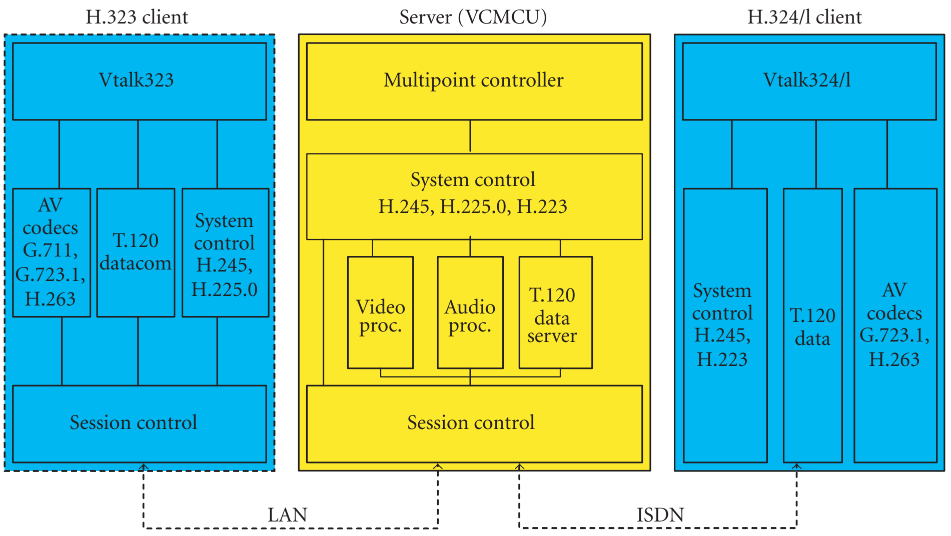
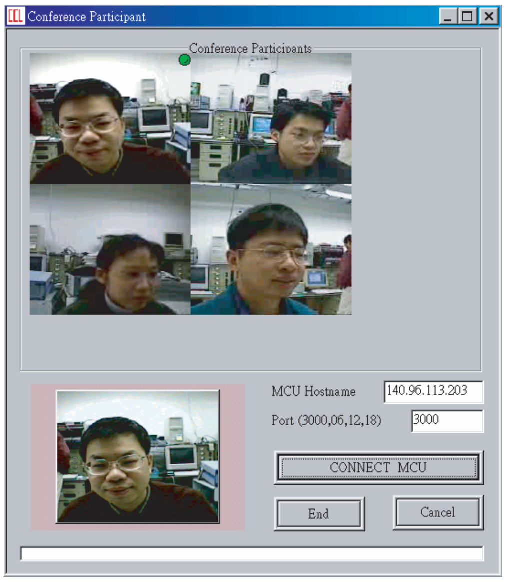


Figure 3: The proposed VCMCU system architecture.

multipoint session. However, in practical applications, using two separate devices to perform the gateway and the conference control functions is expensive and undesirable. Figure 3 depicts the system architecture of the VCMCU. The VCMCU not only serves as an MCU but also plays the role of a gateway to interwork between the H.323 (for LAN) and H.324 [12] (for WAN) client terminals. The client side is basically an H.323 or H.324 multimedia terminal, which generates an H.323- or H.324-compliant bit-stream with integrated audio, video, and data content. The VCMCU server receives and terminates bit-streams from the H.323 and H.324 client terminals as well as performs the multipoint controller (MC) and multipoint processor (MP) functions [10]. The MC and MP are used for protocol conversion and bandwidth adaptation, respectively. In fact, the VCMCU server also includes the complete functions of the H.323/H.324 client terminals.

The VCMCU supports two operation modes. One is the base-line mode, and the other is the advanced virtual meeting mode. Figure 4 shows a snapshot of the baseline-mode user-interface of the VCMCU. It supports the split-screen display format so that four subwindows can be seen simultaneously in a continuous presence fashion. To balance the asymmetrical bandwidth requirement of one incoming and four outgoing video streams, a dynamic bit-allocation video transcoding scheme, described in [13, 14], is used in the VCMCU to allocate appropriate bit-rates to the subwindows by taking into account the joint spatio-temporal complexity of each subwindow. The client terminals may also send their preferences on the conferee bit-allocations to the VCMCU to favor the quality of the video objects of interest.

Figure 5 shows the proposed server-client architecture for video processing used in the advanced virtual meeting mode. At the client side, the video object of each conferee is first extracted. Then a chroma-key [15] is filled in the background. After the chroma-key insertion, the client video is encoded using an H.263 video coder, then the resultant bit-stream is sent to the VCMCU. The server extracts all



MCU Hostname 140.96.113.203
Port (3000,06,12,18) 3000

Figure 4: The baseline-mode user-interface for client terminals.

the client video objects according to the chroma-key information, transcodes the video objects to adapt to the available network bandwidth and the user demands, and inserts the chroma-key filled background again. The server then sends back to each client the transcoded video objects with chroma-keyed background. The clients decode the received bit-stream, extract the chroma-keyed objects, then compose and render the 2D video objects in a pre-stored 3D synthetic environment.

In the process described above, object extraction, tracking, compositing, and manipulation are in general the most computationally demanding tasks. To meet the real-time requirement, we propose a fast object segmentation and tracking scheme, as will be described in the following, which

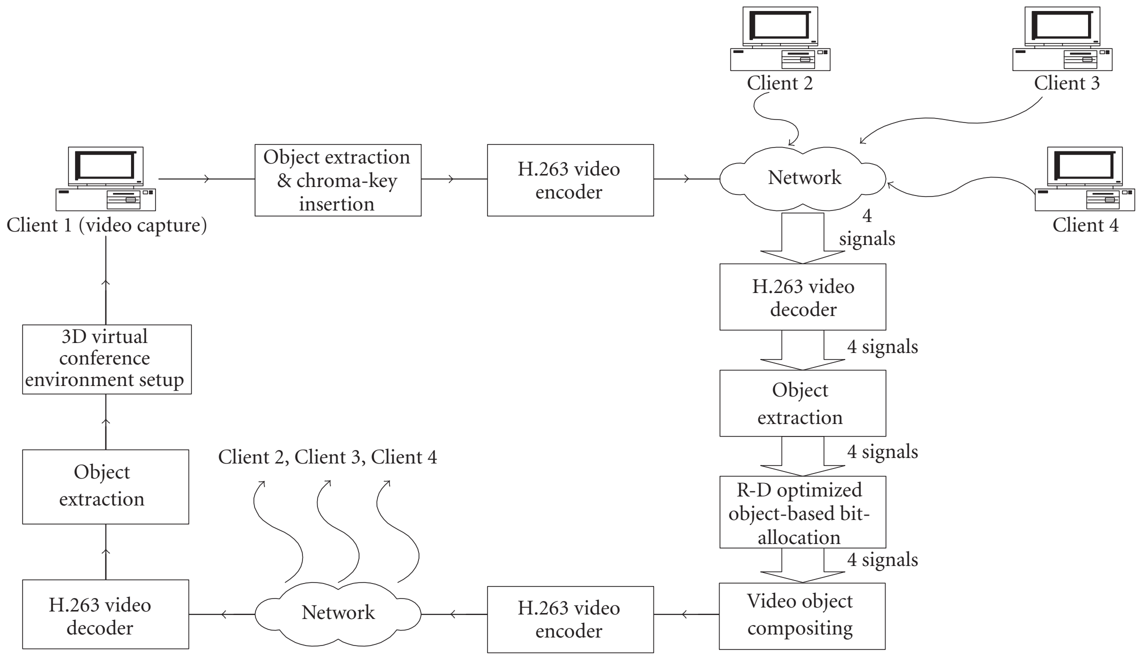


Figure 5: Video processing architecture used in the advanced virtual meeting mode.

uses pre-stored background information, and a chroma-key-based object extraction scheme. We also propose a table look-up based geometrical transformation method for real-time manipulating and rendering the video objects in the pre-stored 3D synthetic background. The computation required for the proposed object compositing, manipulating, and rendering can also be done in the video display cards if they support OpenGL technologies, so as to reduce the computation burden at the client terminals drastically.

3. VIDEO OBJECT EXTRACTION AND TRACKING WITH AN ACTIVE CAMERA

In general, video object segmentation is computationally very expensive. For virtual video conferencing applications, a fully automatic real-time segmentation is needed, which must be able to detect 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 [16, 17, 18], 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 a virtual video conferencing system, 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 discriminate moving objects from the still background [19, 20, 21, 22]. 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 nonstationary or new objects are left. This method is especially suitable for video conferencing [21, 22] and surveillance applications [19], 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 conditions, 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, [23, 24], 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 unflexible. Moreover, in video conferencing applications, the above operations need to be done in real-time, making computational complexity also a major concern.

To overcome the inconvenience and limitation caused by a static camera, we propose a real-time object extraction and tracking scheme using a mosaic technique with an active camera. Figure 6 depicts the conceptual diagram of our proposed scheme, which is aimed at the segmentation of the

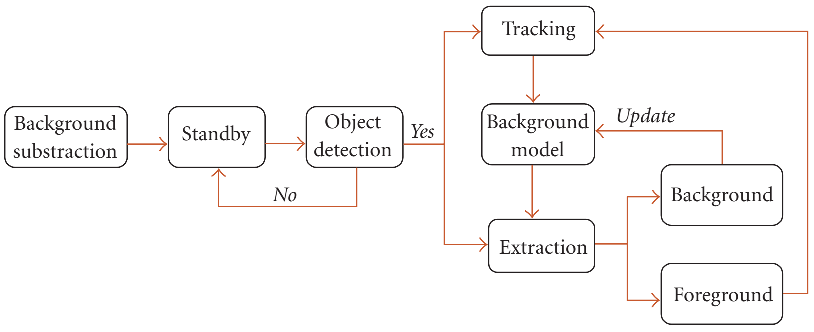


FIGURE 6: The conceptual diagram of the proposed object segmentation and tracking scheme.

foreground object from the background scene and tracking the moving object with an active camera such that the moving object always stays around the center of images. In video conferencing applications, the system requires a robust segmentation algorithm with a pixel-wise accuracy, a reliable tracking strategy, and a low computational complexity.

Prior to performing segmentation and tracking, a number of background images with equally spaced pan and tilt 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 has been constructed, the live video captured from the camera is fed into the detection module. The detection module monitors any change in the scene 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 (foreground) to be extracted from the background with arbitrary pan and tilt angles, and the object can move in a wide range of background.

3.1. Image mosaics construction

The first step to constructing the panoramic mosaic is the alignment of images, that is, to estimate the global motion between the consecutive images, as shown in Figure 7. The following 2-parameter translation motion model is used for images alignment:

$$x' = x + a, \qquad y' = y + b. \tag{1}$$

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

$$\begin{aligned}(a^*, b^*) &= \underset{(a,b)}{\operatorname{argmin}} E(a, b)\\ &= \underset{(a,b)}{\operatorname{argmin}} \frac{\sum_{(x,y)\in S}\left[I_1(x+a, y+b) - I_0(x, y)\right]^2}{\sum_{(x,y)\in S} 1},\end{aligned} \tag{2}$$

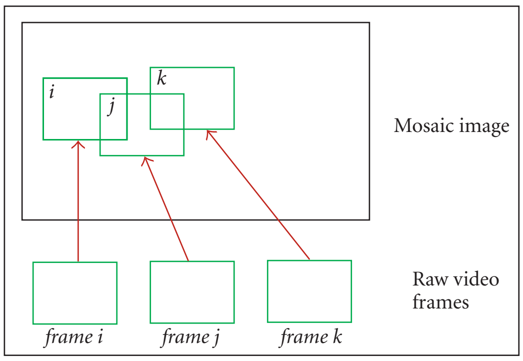


FIGURE 7: Estimation of the global motion and construction of panoramic mosaic image from consecutive frames.

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

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 the following equation and in Figure 8,

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

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:

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

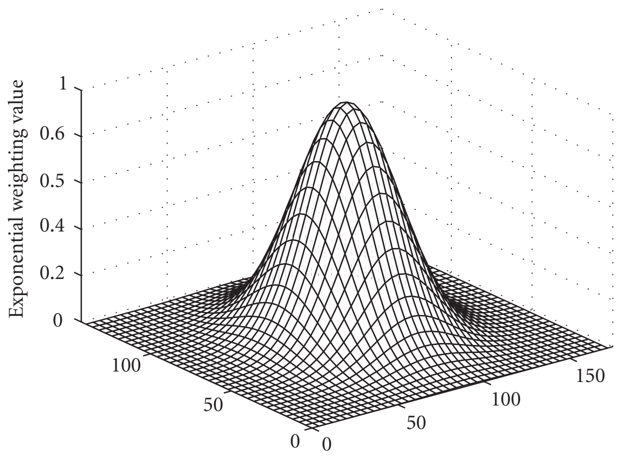


Figure 8: Exponential blending weighting function.

with

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

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.

Figure 9 shows an example of a subview 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.

3.2. Object segmentation based on background subtraction

3.2.1 Background subtraction

Figure 10 depicts the proposed object segmentation and tracking procedure with an active camera using background subtraction. 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:

$$\mu(\mathbf{p}) = \frac{1}{N}\sum_{i=1}^{N} R_i(\mathbf{p}), \qquad \operatorname{std}(\mathbf{p}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} R_i^2(\mathbf{p}) - \mu^2(\mathbf{p})}, \tag{6}$$

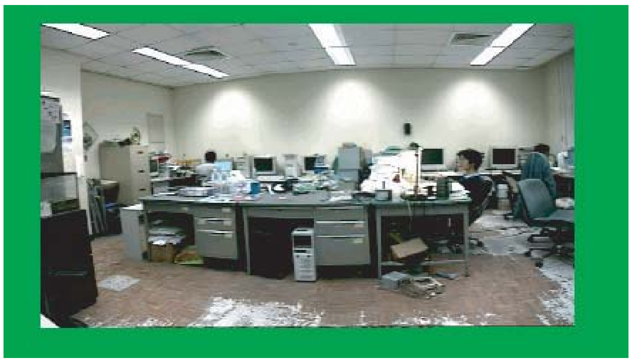


Figure 9: The mosaic image constructed as a reference background.

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 ith background frame; $\mu(\mathbf{p})$ and $\operatorname{std}(\mathbf{p})$ 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). Therefore, each pixel in the mosaic image has two parameters: the mean $\mu_M(\mathbf{p})$ and the standard deviation $\operatorname{std}_M(\mathbf{p})$.

The criterion to classify pixels is described as follows:

$$\begin{aligned}&\text{if } \left(\left|C(\mathbf{p}) - \mu_M(\mathbf{p})\right| > k \times \operatorname{std}_M(\mathbf{p})\right)\\ &\qquad \mathbf{p} \in \text{foreground}\\ &\text{else}\\ &\qquad \mathbf{p} \in \text{background}\end{aligned}$$

where $C(\mathbf{p})$ is the pixel value of position $\mathbf{p}$ in the current frame to be segmented; $\mu_M(\mathbf{p})$, $\operatorname{std}_M(\mathbf{p})$ are the corresponding mean and the standard deviation of the subview in the mosaic background image, respectively; k is a constant used to control the threshold for segmentation.

To locate the subview in the mosaic image as the corresponding background model, there are five steps:

(1) get the camera pan and tilt angle position from the active camera and use these parameters to roughly locate the subview 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 subview in the mosaic image,
(4) update the corresponding background model,
(5) iteratively repeat steps (2), (3), and (4) until the corresponding background model is stable.

3.2.2 Post-filtering

Background subtraction can roughly classify pixels of background and foreground, but the resultant segmentation result may still be quite noisy due to camera noises,

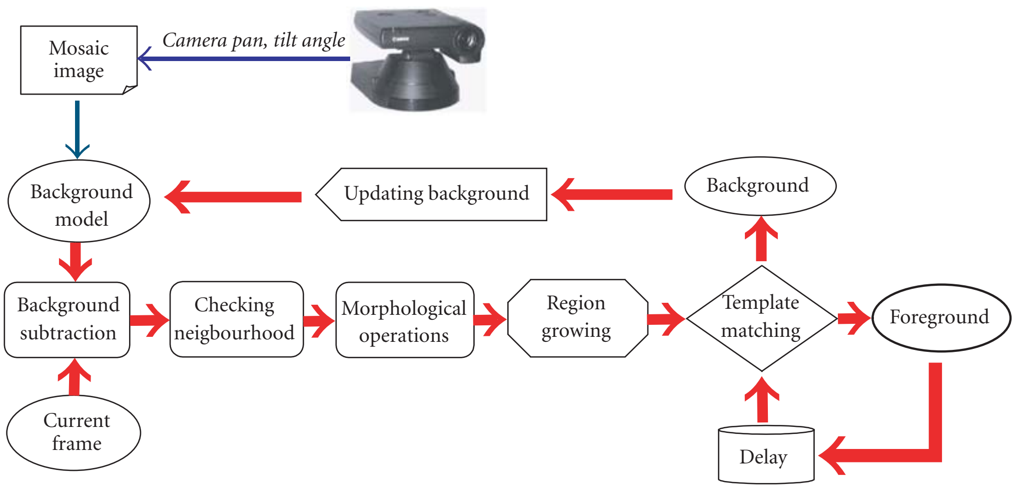


Figure 10: Procedure of the proposed object extraction and tracking with an active camera.

p_1	p_2	p_3
p_4	p	p_5
p_6	p_7	p_8

Figure 11: Neighborhood majority voting.

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, we assume that there may be one-pixel displacement error between the current frame and the background model. Then, each pixel $\mathbf{p}$ in the current frame, as shown in Figure 11, is classified as either a background or a foreground class by considering not only the corresponding pixel in the background model but also the eight neighbors of the corresponding pixel. The neighborhood majority-voting criterion is described as follows:

```
count = 0;
for ($\mathbf{p}_i \in$ Neighborhood($\mathbf{p}$))
    if ($|C(\mathbf{p}) - \mu_M(\mathbf{p}_i)| > k \times \text{std}_M(\mathbf{p}_i)$)
        count = count + 1;

if (count $\geq 3$)
    $\mathbf{p} \in$ foreground
else
    $\mathbf{p} \in$ background.
```

After the neighborhood majority voting, a refined alpha mask [25] is obtained. Then, median filtering is performed to reduce noises. Next, morphological opening and closing operations are used for further improving the segmentation result.

3.2.3 Region growing

At the final step of object discrimination, a region growing method is used to reduce coarse granular noises in the alpha plane. It preserves the region of interest in the alpha plane and erases the other regions, which are considered as background noises. In the region growing procedure, the seed point in the interested region must be chosen first. We propose two ways to select the seed point in the system. One is to use “integral projection” [26] with the alpha plane to obtain the seed point. And the other is to calculate the centroid of the skin-color region in the frame as the seed point for region growing 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 procedure of the region growing method which obtains the seed point using integral projection is summarized as follows:

(1) project the alpha plane $F(x, y)$ using horizontal integral projection and vertical integral projection as shown in Figures 12b and 12c:

$$\begin{aligned} \text{Proj}\, H(y) &= \sum_x F(x, y), \\ \text{Proj}\, V(x) &= \sum_y F(x, y), \end{aligned} \tag{7}$$

where $\text{Proj}\, H(y)$ and $\text{Proj}\, V(x)$ are the horizontal integral projection and vertical integral projection, respectively,

(2) find the index of each maximum value in $\text{Proj}\, H(y)$ and $\text{Proj}\, V(x)$:

$$\begin{aligned} \text{SeedPoint}_x &= \underset{x}{\operatorname{argmax}}\{\text{Proj}\, V(x)\}, \\ \text{SeedPoint}_y &= \underset{y}{\operatorname{argmax}}\{\text{Proj}\, H(y)\}, \end{aligned} \tag{8}$$

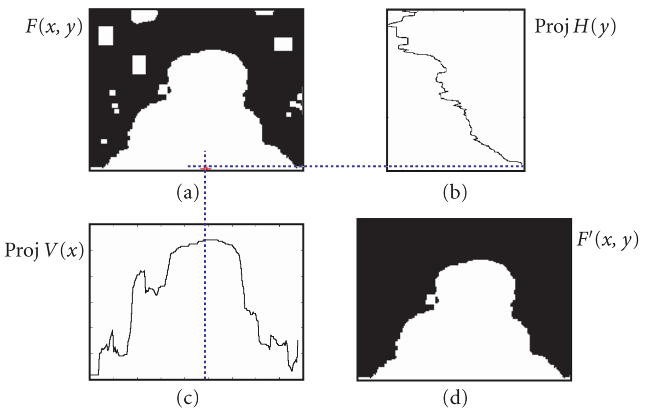


FIGURE 12: Integral projection process: (a) alpha plane with coarse granular noises; (b) horizontal integration result from the alpha plane and dotted lines indicate the maximum value; (c) vertical integration result from the alpha plane and dotted lines indicate the maximum value; (d) segmentation result after removing the coarse granular noises.

(3) retain the region containing the seed point, and erase the other regions.

To find the skin-color region of a captured image, we adopted the method described in [26]. In the skin-color detection method, a pre-trained Gaussian model with the C_b and C_r chrominance components is utilized for characterizing the probability distribution of skin-color. By adopting this probability model, a pixel is classified as the skin-color class when the probability of its color tone being skin-color is higher than a predetermined threshold. After skin-color segmentation, the centroid of the skin-color region is then computed.

3.2.4 Background update

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

```
If $C(\mathbf{p}) \in$ foreground
    $\mu'_M(\mathbf{p}) = \mu_M(\mathbf{p})$
else
    $\mu'_M(\mathbf{p}) = (1-\eta)\mu_M(\mathbf{p}) + \eta C(\mathbf{p})$.
```

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 to the slow changes in lighting conditions in the images.

3.3. Object segmentation and 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, for 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 Figure 10, when camera moves, we get the camera pan and tilt angle position through an RS-232 port and then we only use these parameters to roughly locate the subview in the mosaic image without iteratively finding the corresponding background model. This can save a lot of computation. To refine the roughly segmented 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 $C_n(\mathbf{p})$.

Tracking the moving object is a challenging task because searching for features and obtaining the correspondence between consecutive frames is not easy. 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 of the object shape. The tracking is performed as follows.

3.3.1 Template matching

The correspondence problem can be formulated as a backward matching motion estimation problem, similar to that employed in predictive video compression. For fast and robust template matching, we adopted the diamond search algorithm proposed in [27]. The template matching criterion is described as follows:

```
if $C_n(\mathbf{p}_1) \in$ foreground
    find $C_{n-1}(\mathbf{p}_2)$, which corresponds to $C_n(\mathbf{p}_1)$
    if $C_{n-1}(\mathbf{p}_2) \in$ foreground
        $C_n(\mathbf{p}_1) \in$ foreground
    else
        $C_n(\mathbf{p}_1) \in$ background
```

3.3.2 Moving object tracking

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 selected feature point (e.g., the centroid of the skin-color region). A linear trajectory model is used to predict the feature point's future position. Suppose that 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 $\mathbf{v}$ which is obtained from the two previous frames,

$$\mathbf{v}(t) = \mathbf{p}(t) - \mathbf{p}(t-1), \qquad \hat{\mathbf{p}}(t+1) = \mathbf{p}(t) + \mathbf{v}(t). \tag{9}$$

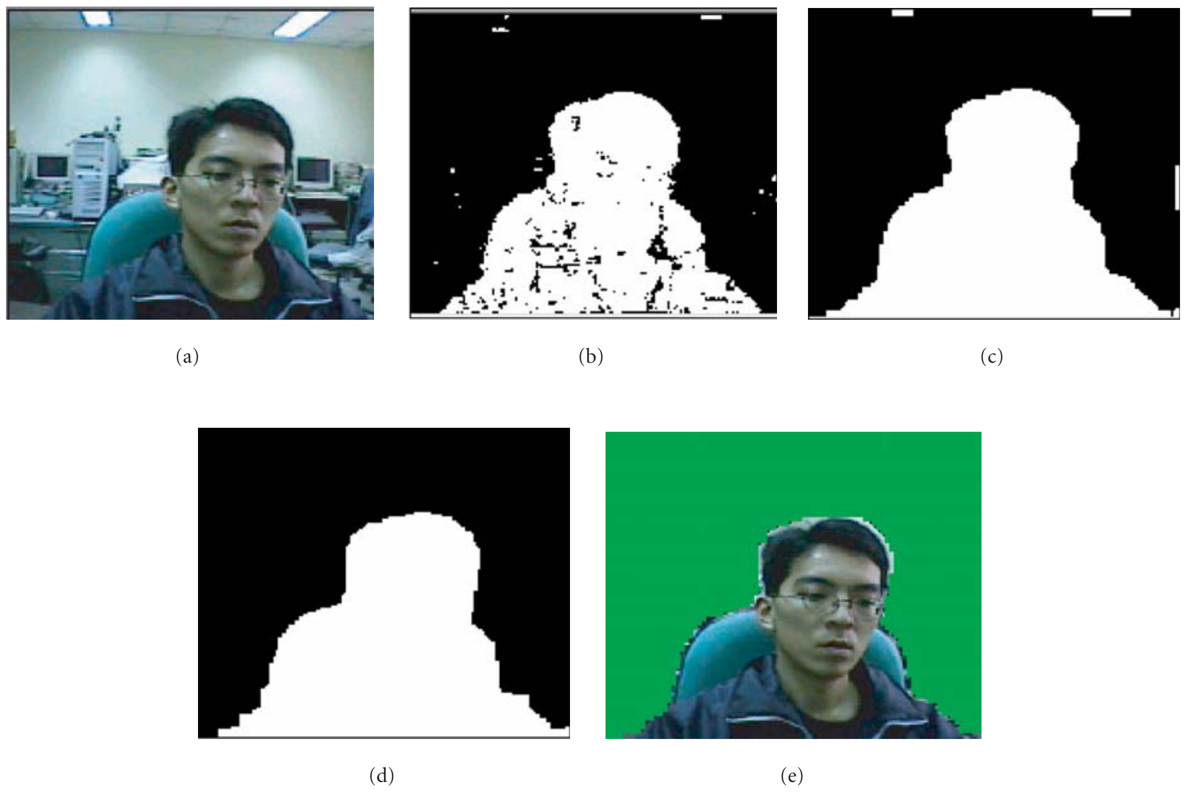


FIGURE 13: The simulation result of the proposed object discrimination: (a) the incoming image; (b) alpha plane obtained by background subtraction; (c) alpha plane after post-filtering; (d) alpha plane after region growing; (e) final segmentation result.

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

$$\text{if } \widehat{\mathbf{p}}(t+1) \notin \text{CR}$$
$$\text{set Camera Velocity} = \frac{\mathbf{S}\left(\mathbf{p}(t+1)-\mathbf{p}_c\right)}{\Delta t},$$

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

3.4. Experiment results of object extraction

Figure 13 shows the simulation result of the proposed object segmentation scheme. Figure 13a shows a captured image containing the foreground object. Figure 13b depicts the rough segmentation results after performing the background subtraction scheme. The rough segmentation can still be quite noisy. The result after applying the neighborhood majority voting and morphological filtering is illustrated in Figure 13c . The small granular noises can be effectively eliminated using the post-filtering process as shown. Figures 13d and 13e show the final segmentation mask and the segmented video after region glowing, respectively.

Figure 14 illustrates the result of skin-color classification. Discriminating the skin-color in the foreground region can reduce computation and avoid the disturbance of skin-color in the background. The centroid of the skin-color region is utilized as the tracking feature since it varies smoothly and is easy to obtain. As a typical example shown in Figure 15, the predicted position is very close to the real position, which justifies that the assumptions used in our method are fair approximations.

Figure 16 illustrates that template matching can effectively reduce the noise in the segmentation result. However, if the camera continuously moves for a long time, the boundary of the segmentation result may become inaccurate for possible error propagations caused by iterative use of previous segmented result for tracking the current object.

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 more than 15 QCIF (176×144) frames per second.

4. VIDEO OBJECT COMPOSITING USING GEOMETRICAL TRANSFORMATION

As shown in Figure 5, after extracting the video objects, the server transcodes the video objects using the dynamic bit-allocation method described in [13, 14] and inserts

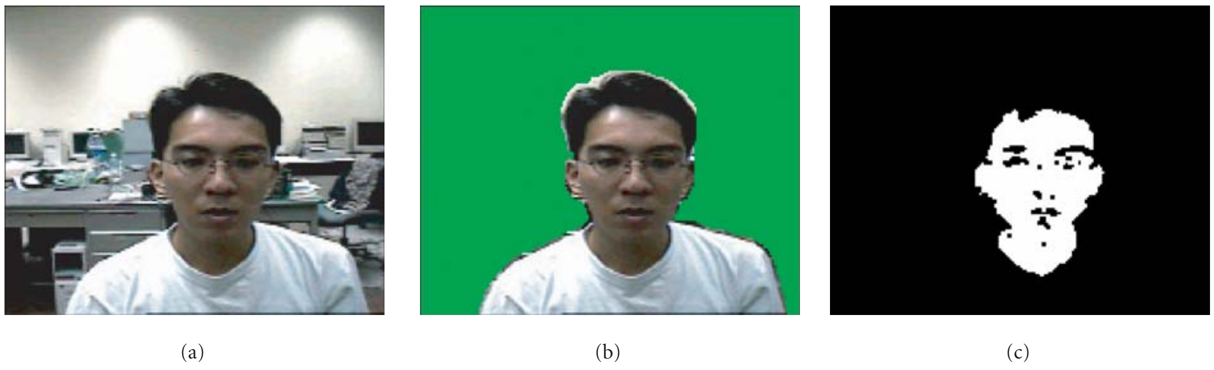


Figure 14: (a) Image captured from the camera, (b) the extracted foreground, (c) the skin-color region in the foreground after median filtering.

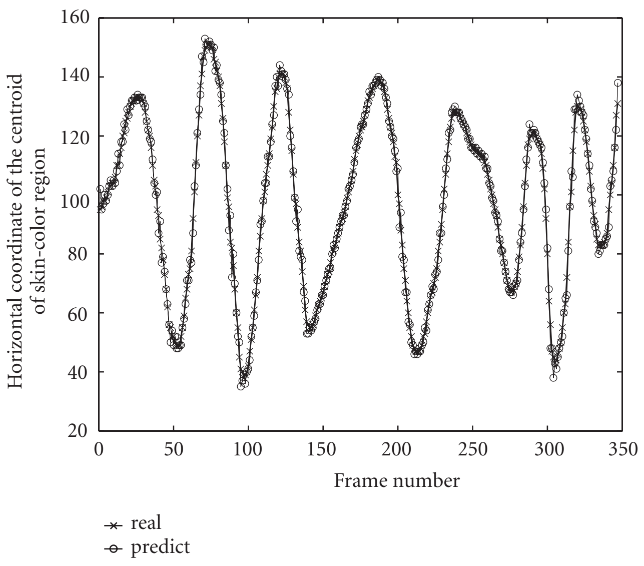


Figure 15: Horizontal coordinate of predicted and real centroids of skin-color region in the foreground.

a chroma-key background [15], then sends the chroma-keyed video back to the client terminals. The resulting video bit-stream still conforms to the H.263/H.263+ standard. The client side subsequently extracts the video objects from the received bit-stream and manipulates and renders the 2D video objects against a virtual scene.

The simplest form of a virtual scene is a 2D synthetic background image as shown in Figure 17. Video object manipulation can be achieved with geometrical transform consisting of translation, scaling, and orientation-change. For each video stream, the user specifies the position of the object in the composite frame, along with the scale and orientation for each object. If object overlap is permitted, we need to specify priorities for the different streams. This priority dictates the order to be followed in the object compositing.

Translation is achieved by simply changing the pixel indices of the segmented objects. Scaling is achieved by a simple interpolation or decimation process. For example, if the size of an image is to be enlarged from $[N_1 \times N_2]$ to $[M_1 \times M_2]$, a pixel at the location (m_1, m_2) of the enlarged image is obtained from the pixel location $(\lceil m_1 \times N_1/M_1 \rceil, \lceil m_2 \times N_2/M_2 \rceil)$ of the original frame, where $\lceil\,\rceil$ represents the rounding operation. In the implementation, we use a lookup table to decide which index corresponds to which index in the transformed image. To achieve better visual effects, we may need to rotate the image plane in 3D. This can be achieved by "pasting" the frame to an imaginary cube in 3D and then rotate this cube in 3D by the two rotation angle parameters, and then take either an orthographic or perspective projection of the 3D object to 2D again. However, we can approximate this procedure by the simple "rotation" procedure in 2D itself as described below.

Assume that we are given an image and we have to rotate it as shown in Figure 18. We can simply specify the rotation parameter and obtain a mapping between the "rotated" object and its original. For a point (x, y) on the "rotated figure," the corresponding (x_0, y_0) on the original figure will be given by the following simple equations:

$$y_0 = \begin{cases} y + x\tan\theta, & \text{if } y \geq 0.5H, \\ y - x\tan\theta, & \text{otherwise,} \end{cases}$$
$$x_0 = \frac{\text{Stretched length}}{\text{Original length}} \cdot \sqrt{x^2 + y^2}, \tag{10}$$

where H is the height of the video object.

In our implementation, the above simple model adequately achieves the effect of an actual 3D rotation. This simple model also makes it possible to incorporate all the parameters for translation, rotation, and scaling into a single transformation neatly implemented by a single lookup table. This approach is very computationally efficient since the lookup table is established only when the users change their configuration of virtual meeting control.

The compositing of the final video can be done in different ways. One approach is to decide for each pixel of the composite frame, which incoming stream's pixel should be allocated, based on their priorities. However this scheme

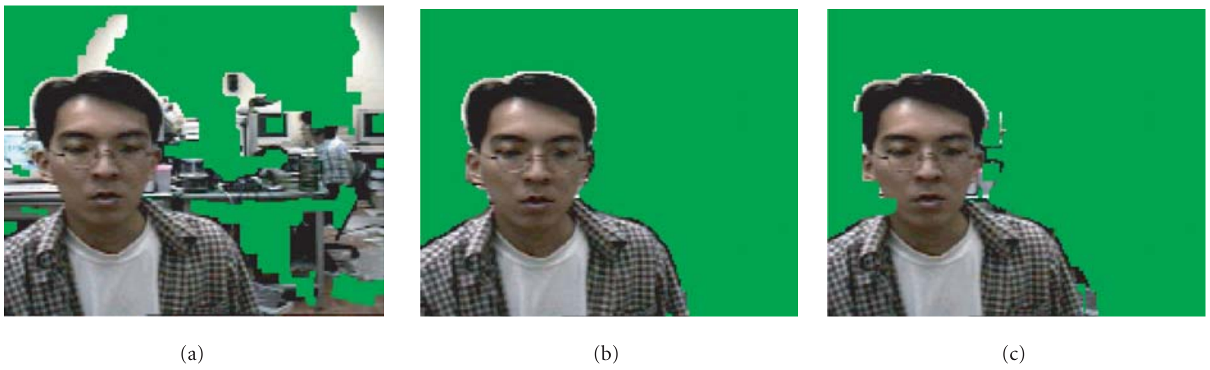


FIGURE 16: (a) segmentation without template matching, (b) previous segmented result used as template, (c) refined segmentation result after template matching.

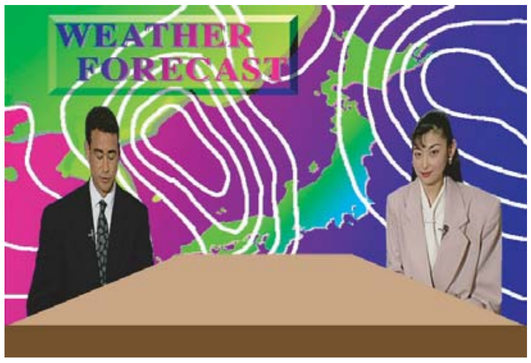


FIGURE 17: Object compositing with user control of scaling, repositioning, and rotation.

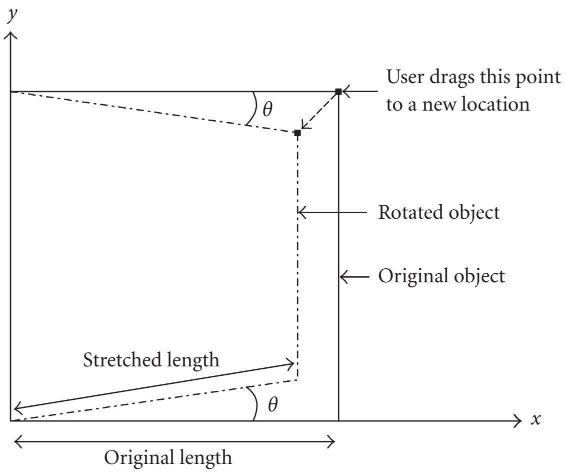


FIGURE 18: Illustrating the simple plane image rotation scheme.

involves several comparison operations, which can be time-consuming. Another approach is to write all the segmented video objects to the new background, according to their priority. We first write the object with the lowest priority, followed by the next lower priority object and so on. Of course these objects are "geometrically transformed" before we actually write them to the composite frame.

To provide better visual perception and interactivity, a shared 3D virtual environment as in [7, 8, 9] can be created for navigation and interaction. Additional 3D location information is required to be sent and delivered between server and clients, so the server-client architecture should be modified as in Figure 19.

By using 3D computer graphics libraries such as OpenGL and DirectX, a 3D virtual environment can be created with 3D object modeling parameters and rendered according to the specified viewing models. Integration with mouse operation functions, a 3D immersive environment can be generated in which users can freely navigate and interact with other participants in the 3D space. As shown in Figure 20, we create a 3D scene composed of a meeting room and wooden floor as virtual environment. Using OpenGL technologies, which have been supported by most of the commercial 3D display cards, at the client side, the required computation for object manipulations (e.g., scaling, repositioning, and rotation) and rendering can be handled by OpenGL APIs with hardware support. Thus, most of the computing power of the client terminals can be dedicated to the object segmentation and video encoding and decoding. Furthermore, speech signals are mixed according to the virtual distance between the sender and the receiver in the 3D space for providing an aural immersive virtual environment.

The overall system can be operated with a processing speed of about 10 frames per second on Pentium-III 733 MHz PCs (clients: QCIF frame; server: CIF frame) without particular code optimization, where we used one PC for each client and the server.

5. CONCLUSIONS

We have described the algorithms implemented in a virtual video conference system which plays the role of both MCU and gateway. The LAN users through Ethernet and the WAN users through ISDN can be brought together via the system. The proposed architecture provides an

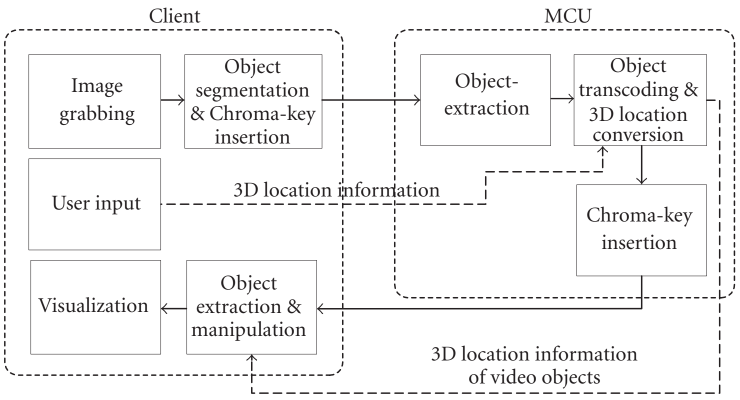
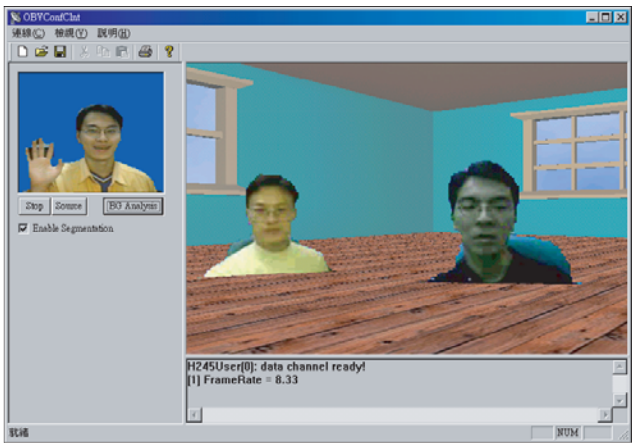

Client
MCU
Image grabbing
Object segmentation & Chroma-key insertion
Object-extraction
Object transcoding & 3D location conversion
User input
3D location information
Chroma-key insertion
Visualization
Object extraction & manipulation
3D location information of video objects

FIGURE 19: The proposed server-client architecture for virtual meeting presentation.



```
H245User[0]: data channel ready!
[1] FrameRate = 8.33
```

FIGURE 20: One snapshot of the proposed virtual meeting with natural 2D objects in a synthetic 3D environment using OpenGL technologies.

integrated platform for video, voice, and data communications, and is fully compatible to the H.323/H.324 standards. In addition to conventional split-screen display, the proposed VCMCU also provides advanced user controllable object processing features such as scaling, repositioning, rotation, and dynamic bit-allocation in real-time. We have presented efficient methods for implementing video object extraction, tracking, compositing, and manipulation, and other user-friendly object processing features. We have implemented a real-time virtual conference system prototype to demonstrate the feasibility of the proposed methods.

REFERENCES

[1] M. E. Lukacs, D. G. Boyer, and M. Mills, "The personal presence system experimental research prototype," in *Proc. IEEE International Conference on Communications*, vol. 2, pp. 1112–1116, Dallas, Tex, USA, June 1996.

[2] M. E. Lukac and D. G. Boyer, "A universal broadband multipoint teleconferencing service for the 21st century," *IEEE Communications Magazine*, vol. 33, no. 11, pp. 36–43, 1995.

[3] *Microsoft V-Chat 2.0, developed by the social computing group of Microsoft Research, September 1999, available at: http://research.microsoft.com/scg/*.

[4] J. Ohya, K. Kitamura, F. Kishino, N. Terashima, H. Takemura, and H. Ishii, "Virtual space teleconferencing: real time reproduction of 3D human images," *Journal of Visual Communication and Image Representation*, vol. 6, no. 1, pp. 1–25, 1995.

[5] T. K. Çapin, I. S. Pandzic, H. Noser, D. Thalmann, and N. M. Thalmann, "Virtual human representation and communication in VLNet," *IEEE Computer Graphics and Applications*, vol. 17, no. 2, pp. 42–53, 1997.

[6] W. H. Leung, K. Goudeaux, S. Panichpapiboon, S.-B. Wang, and T. Chen, "Networked intelligent collaborative environment (NetICE)," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, pp. 1645–1648, New York, NY, USA, July 2000.

[7] H. Nakanishi, C. Yoshida, T. Nishimura, and T. Ishida, "Freewalk: a 3D virtual space for casual meetings," *IEEE Multimedia*, vol. 6, no. 2, pp. 20–28, 1999.

[8] S. Sugawara, G. Suzuki, Y. Nagashima, M. Matsuura, H. Tanigawa, and M. Moriuchi, "Interspace: networked virtual world for visual communication," *IEICE Transactions on Information and Systems*, vol. E77-D, no. 12, pp. 1344–1349, 1994.

[9] S. Sugawara, N. Matsura, Y. Kato, et al., "Interspace project cyber campus," in *ACM 1996 Conference on Computer Supported Cooperative Work, Video Program, no. 3*, ACM, November 1996.

[10] ITU-T Recommendation H.323, "Visual telephone systems and terminal equipment for local area networks which provide a non-guaranteed quality of service," 1998.

[11] ITU-T Recommendation H.263, "Video codec for low bit-rate communication," 1996.

[12] ITU-T Recommendation H.324, "Terminal for low bit rate multimedia communication," .

[13] C.-W. Lin, T.-J. Liou, and Y.-C. Chen, "Dynamic rate control in multipoint video transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 17–20, Geneva, Switzerland, May 2000.

[14] C.-W. Lin, W.-H. Wang, Y.-C. Chen, M.-T. Sun, and J.-N. Hwang, "Implementation of H.323 video conference systems with personal presence control," in *Proc. IEEE Int. Conf. Consumer Electronics*, Los Angeles, Calif, USA, June 2000.

[15] T. Chen, C. T. Swain, and B. G. Haskell, "Coding of subregions for contentbased scalable video," *IEEE Trans. Circuits*

and Systems for Video Technology, vol. 7, no. 1, pp. 256–260, 1997.

[16] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 572–584, 1998.

[17] R. Castango, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features for interactive multimedia application," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 562–571, 1998.

[18] M. Kim, J. G. Choi, D. Kim, et al., "A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1216–1226, 1999.

[19] 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, Nara, Japan, April 1998.

[20] B. Li and M. I. Sezan, "Adaptive video background replacement," in *Proc. IEEE Int. Conf. Multimedia and Expo*, pp. 385–388, Tokyo, Japan, August 2001.

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

[22] J. Pan, C. Gu, and M.-T. Sun, "An MPEG-4 virtual video conferencing system with robust video object segmentation," in *Proc. MPEG-4 Workshop and Exhibition*, San Jose, Calif, USA, June 2001.

[23] Y. Ye, J. K. Tsotsos, K. Bennet, and E. Harley, "Tracking a person with prerecorded image database and a pan, tilt, and zoom camera," in *Proc. IEEE Workshop on Visual Surveillance*, pp. 10–17, Bombay,, India, January 1998.

[24] S. Hat, M. Saptharishi, and P. K. Khosla, "Motion detection and segmentation using image mosaics," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 3, pp. 1577–1580, New York, NY, USA, 28 July–2 August 2000.

[25] A. Puri and T. Chen, *Multimedia Systems, Standards, and, Networks*, Marcel Dekker, New York, NY, USA, 2000.

[26] C.-W. Lin, Y.-J. Chang, and Y.-C. Chen, "Low-complexity face-assisted video coding," in *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 207–210, Vancouver, British Columbia, Canada, September 2000.

[27] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast blockmatching motion estimation," *IEEE Trans. Image Processing*, vol. 9, no. 2, pp. 287–290, 2000.

Chia-Wen Lin received his M.S. and Ph.D. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1992 and 2000, respectively. Dr. Lin joined the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, in August 2000, where he is currently an Assistant Professor. Before that, he was a Section Manager of the CPE and Access Technologies Department at the Computer and Communications Research Laboratories, Industrial Technology Research Institute (CCL/ITRI), Taiwan. During April to August 2000, he was with the Information Processing Lab, Department of Electrical Engineering, University of Washington, as a visiting scholar. His research interests include video coding and networked multimedia technologies.

Yao-Jen Chang received his B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1996. He is currently pursuing his Ph.D. degree in electrical engineering at the National Tsing Hua University. His research interests include computer vision, computer graphics, and multimedia signal processing.

Chih-Ming Wang received his B.S. and M.S. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1999 and 2001, respectively. He has been with Winbond as an IC design engineer since October 2001. His research interests include computer vision and multimedia signal processing.

Yung-Chang Chen received his B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1968 and 1970, respectively, and the Ph.D. (doctorate in engineering) degree from Technische Universitaet Berlin, Germany, in 1978. He joined the Department of Electrical Engineering at National Tsing Hua University, Hsinchu, Taiwan in 1978, where he is currently a Professor. He was Chair of Department of Electrical Engineering at National Central University, Chungli, Taiwan from 1980 to 1983, and Chair of Department of Electrical Engineering at National Tsing Hua University from 1992 to 1994. Since February 2002, He has been appointed as Professor at the Department of Computer Science and Information Engineering, and Dean of the College of Engineering, National Chung Cheng University, Chiayi, Taiwan. He is Senior Member of IEEE and serves as Chair of CES, Taipei Chapter. His current research interests include multimedia signal processing, digital video processing, medical imaging, computer vision, and pattern recognition.

Ming-Ting Sun received his B.S. degree from National Taiwan University in 1976, the M.S. degree from University of Texas at Arlington in 1981, and the Ph.D. degree from University of California, Los Angeles in 1985, all in electrical engineering. Dr. Sun joined the University of Washington in August 1996 where he is Professor. Before that, he was the Director of the Video Signal Processing Research Group at Bellcore. Dr. Sun has been awarded 7 patents and has published more than 120 technical papers including 10 book chapters in the area of video technology. He is a Fellow of IEEE. He was a conference general co-chair of SPIE VCIP2000 (Visual Communications and Image Processing 2000). He received a Golden Jubilee Medal from the IEEE CAS Society in 2000, and was the Editor-in-Chief of *IEEE Transactions on Circuits and Systems for Video Technology* (from 1995 to 1997) and *IEEE Transactions on Multimedia*(from 1999 to 2001). He was a corecipient of the TCSVT Best Paper Award in 1993. From 1988 to 1991, he served as the Chairman of the IEEE CAS Standards Committee and established an IEEE Inverse Discrete Cosine Transform Standard. He received an Award of Excellence from Bellcore in 1987 for the work on digital subscriber line.