IMPLEMENTATION OF H.323 COMPLIANT VIRTUAL MEETING SYSTEMS

Chia-Wen Lin
Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan, R.O.C.
cwlin@cs.ccu.edu.tw
Wei-Xien Lin, Yung Chang Chen
Department of Electrical Engineering
National Tsing Hua University, Hsinchu, Taiwan R.O.C.
{wxlin, ycchen}@benz.ee.nthu.edu.tw
Ming-Ting Sun
Department of Electrical Engineering
University of Washington, Seattle, Washington, USA
sun@ee.washington.edu

ABSTRACT

This paper presents an H.323 standard compliant video conference system implementation. The proposed system not only serves as an MCU (Multipoint Control Unit) for multipoint connection but also provides the gateway function between the H.323 LAN (Local Area Network) and the H.324 WAN (Wide Area Network) users. The proposed video conference system supports two operation modes. The baseline mode provides a split-screen continuous presence display so that each conferee can see all the others simultaneously. The advance mode provides more user-friendly object compositing and manipulation features including 2-D video object scaling, re-positioning, rotating, and dynamic bit-allocation in a 3-D virtual environment. A segmentation scheme based on pre-stored background information is proposed for real-time segmentation of the foreground video objects at the client side. Chroma-key insertion is used for object extraction to facilitate video object manipulation. We also present a geometric transformation scheme and its real-time implementation for video objects manipulation. We have implemented a virtual conference system prototype to demonstrate the feasibility of the proposed methods.

1. INTRODUCTION

With the rapid growth on multimedia signal processing and communication, virtual meeting technologies are getting mature. 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, etc.

ITU-T H.323 [1] is a standard that defines the components, procedures, and protocols necessary to provide audio-visual communications over LANs. H.323 can be used in any packet-switched network, regardless of the ultimate physical layer. Fig. 1 shows the logical view of an H.323 terminal. As shown in Fig. 1, 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.

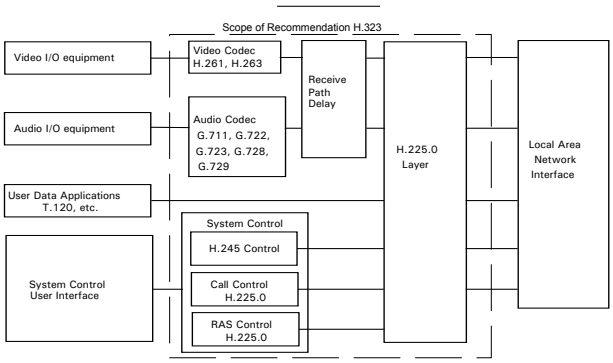


Fig. 1. A logical view of an H.323 terminal

In this paper, we address the implementation of a virtual meeting system based on the H.323 platform. We propose an H.323 Personal Presence Multipoint Control Unit (PPMCU) which adopts the H.263 [2] video coding standard. The proposed virtual meeting system involves 2-D natural video objects and 3-D synthetic environment. We use chroma-key-based object extraction and manipulation schemes so that the developed techniques can be adopted in H.263 compatible systems. The

concept can be easily extended to MPEG-4 [3] based systems.

The rest of this paper is organized as follows. In Section 2, we discuss the architecture of the proposed H.323/324[4] compatible video conference system. In Section 3, we propose a pre-stored background based video object segmentation scheme for real-time segmenting out the conferees in a video conferencing. In addition, a chroma-key based object extraction scheme is also presented for facilitating video object compositing and manipulation. In Section 4, we present an object manipulation scheme based on geometrical transformation, and the real-time implementation of the virtual meeting system. Finally, conclusions are given in Section 5.

2. PROPOSED SYSTEM ARCHITECTURE

The proposed PPMCU is a PC-based prototype for multipoint video bridge and gateway which performs the multimedia communications and conversion and protocol translation among multiple LAN and WAN terminals. Here the Ethernet and ISDN are selected as the target networks for LAN and WAN users respectively. The conceptual model of the proposed PPMCU is shown in Fig. 2. The LAN users can establish a multipoint video conference call with the WAN (ISDN) users via the proposed PPMCU. Typically a gateway is 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 multipoint operation. However in real environments, using two separate devices to perform gateway and conference control respectively is expensive and undesirable. The proposed PPMCU covers both the MCU and the gateway functions. This MCU function can be considered as the extension of Multipoint Controller (MC) and Multipoint Processor (MP) to enhance the capability of multipoint communication over LAN and ISDN. The MC and MP are used for protocol conversion and bandwidth adaptation respectively.

Fig. 3 depicts the architecture of the proposed PPMCU. As shown in this figure, the proposed PPMCU not only serves as a Multipoint Control Unit but also plays the role of a gateway to interwork between the H.323 (for LAN) and H.324 [2] (for WAN) equipment. The video conference equipment is divided into two sides: the server side and the client side. The client side is basically an H.323/H.324/I video terminal, which generates an H.323/H.324/I-compliant bit-stream with integrated audio, video, and data content. The PPMCU server receives and terminates the bit-streams from the H.323 and H.324/I client terminals as well as performs the MC and MP functions. In fact, the PPMCU server also includes the complete functions of the H.323/H.324/I client side terminals. Several advance video transcoding techniques proposed in [5-6] are performed in the PPMCU server for rate adaptation and video quality enhancement.

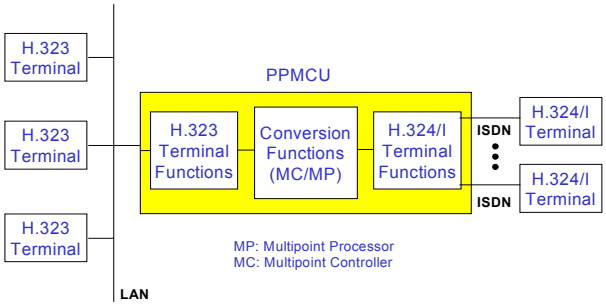


Fig. 2. The conceptual model of the proposed H.323 PPMCU.

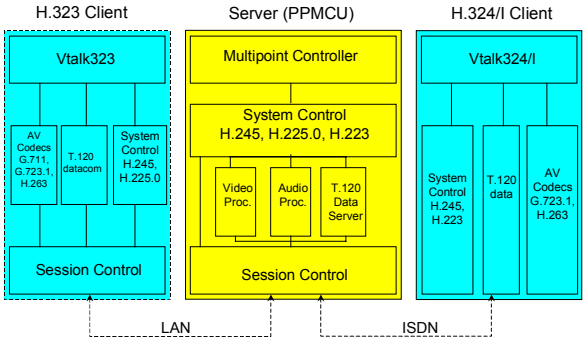
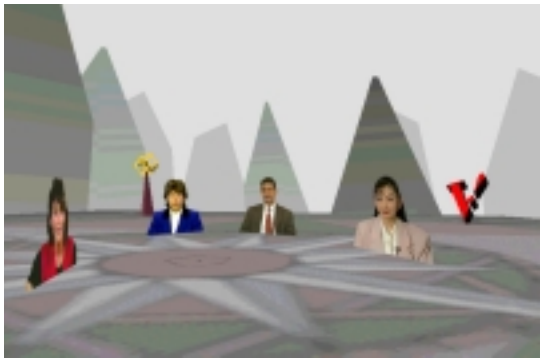


Fig. 3. The proposed PPMCU architecture.

Fig. 4. The baseline-mode user-interface of the PPMCU.

Fig. 4 shows the baseline-mode user-interface of the PPMCU. It supports the split-screen display format so that four sub-windows can be seen simultaneously in a continuous presence fashion. To balance the bandwidth requirement of one incoming and four outgoing video

streams, a dynamic bit-allocation video transcoding scheme described in [5] is used to allocate bit-rates to the sub-windows by taking into account the joint spatio-temporal complexity of each sub-window.



(a)

(b)

Fig. 5. A virtual meeting with 2-D video objects manipulated in two different 3-D virtual environments.

Fig. 5 illustrates a type of virtual meeting which involves 2-D natural video objects (from the remote conferees) in a synthetic 3-D virtual environment. In order to make this possible, several key technologies need to be incorporated. For example, we need to be able to segment out the video objects in the video streams from remote locations and to compose them together so that these video objects appear interacting in the same environment. Video object segmentation and manipulation are both very computationally intensive.

Fig. 6 shows the proposed architecture for the implementation of an H.263 compatible virtual conference prototype system. In a 3-D virtual environment, the location of each conferee is known, so are the relative positions of all the conferees. At the client side, the video object of each conferee is first segmented out. Then a chroma-key is filled in the background. After chroma-key insertion, the client video accompanied with the 3-D position parameters are sent to the sever. The sever extracts all the client video objects using chroma-key, transcodes the video objects to adapt to the available band-width and the user demands, and inserts the chroma-key again. Then the server sends back the transcoded video objects filled with chroma-keyed background and their corresponding 3-D position parameters to each client. The clients compose and render the 2-D video objects in a pre-stored 3-D synthetic environment according to the their 3-D position parameters.

In the process described above, object segmentation, extraction, and manipulation are in general the most computationally demanding tasks. To meet the real-time requirement, as described in the following, we propose a fast object segmentation scheme which uses the 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 video objects in a pre-stored 3-D 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 decoders drastically.

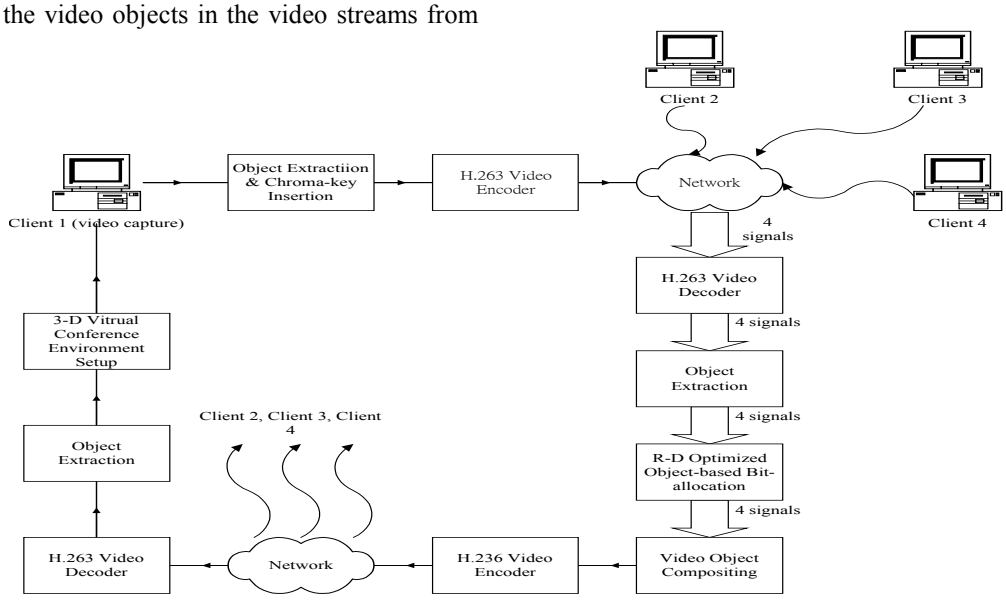


Fig. 6. The proposed virtual meeting architecture

3. VIDEO OBJECT SEGMENTATION AND CHROMA-KEY BASED OBJECT EXTRACTION

3.1 Object Segmentation Based on Still Background Subtraction

Extracting moving objects from a video sequence is a fundamental and crucial problem in many digital video applications, such as video surveillance, video editing, traffic monitoring and human extraction for virtual video conferencing or human-machine interface. Still background subtraction is a simple yet efficient method to discriminating moving objects from the still background [7]. The idea of background subtraction is to subtract the current image from a reference image, which is acquired from a still background during a period of time. After subtraction, only non-stationary or new objects are left. This method is especially suitable for video conferencing applications, since in a video conference, the backgrounds for the conferees in general remain unchanged during the conference time. Should the background be changed, the users can capture and store the new background information again for object segmentation.

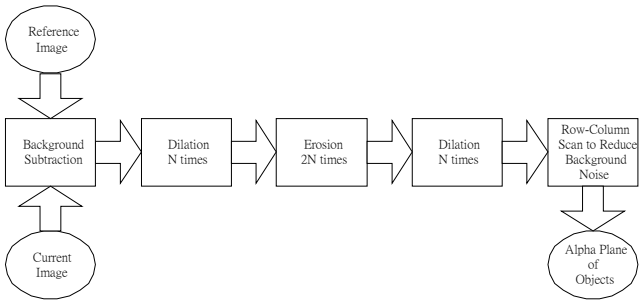


Fig. 7. The proposed object segmentation procedure

The proposed object segmentation procedure with still background subtraction is depicted in Fig. 7. In background subtraction method, the background needs to be analyzed over several seconds of video. Each pixel value over a period of time may change due to camera noises and illumination fluctuations caused by light sources. The background scene is modeled by representing each pixel with two parameters, the mean and the standard deviation, during the training period. Each pixel in the current frame is first classified as either a background or a foreground class using the pre-calculated background model. The criterion to classify pixels is described as follows:

$$
\begin{aligned}
&\textit{if } (\left|C(x)-\text{mean}(x)\right| > k \times \text{std}(x)) \\
&\quad x \in \textit{ foreground} \\
&\textit{else} \\
&\quad x \in \textit{background}
\end{aligned}
$$

where

$$\text{mean}(x) = \frac{1}{N}\sum_{n=1}^{N} R_n(x)$$

$$\text{std}(x) = \sqrt{\frac{1}{N}\sum_{n=1}^{N} R_n^2(c) - M^2(x)}$$

where x presents the index of pixels in the whole frame. $C(x)$ and $R_n(x)$ are luminance values of the pixel x in the current frame and the reference frame respectively. mean(x) and std(x) represent the mean and the standard deviation of the luminance values of the pixel x during the N reference frames.

Background subtraction can roughly classify pixels of background and foreground, but the resulting segmentation result still may be noisy due to the selection of the threshold value. We propose to use the morphological opening and closing operations to reduce small granular noises. Morphological image processing is a type of processing in which the spatial form or structure of objects within an image are modified. Dilation and erosion are two fundamental morphological operations. With dilation, an object grows uniformly in spatial extent, while with erosion an object shrinks uniformly. Eight-Neighbor Dilation and Eight-Neighbor Erosion are used for removing noises in our system.

Generally speaking, finding a satisfactory combination of erosion and dilation steps is quite difficult, and no fixed combination can work well for all natural images. In our experiments, two iterations of dilation are used first to eliminate two-pixel thick noises inside the foreground. Then four iterations of erosion are used to compensate the previous dilations and eliminate two-pixel thick background noises, and two iterations of dilation are used finally. After morphological operations, two-pixel thick noises in the frame would be eliminated.

At the final step of object discrimination, a row-column scan method is used to reduce large noise around object. In video conferencing applications, the head-and-shoulders type human body is usually the main object, and it is usually larger than background noises in row or column direction. Using the proposed row-column scan method to the morphologically segmented alpha-plane can reduce the noises around object in video conferencing applications. In this method, the alpha-plane is scanned row-wise and column-wise to find the separated lines of ‘1’s of each row and each column, and these lines are sorted to find the longest line. The values of memory location corresponding to the longest ”1”s lines of each row and each column of the alpha plane are kept unchanged (e.g., “1”s), while others are set to zero.

After using the row-column scan method, the alpha-plane mask is replaced with the new alpha-plane. And most surrounding noise in this new alpha-plane will be eliminated. The processing flow of the row column scan method is illustrated in Fig. 8.

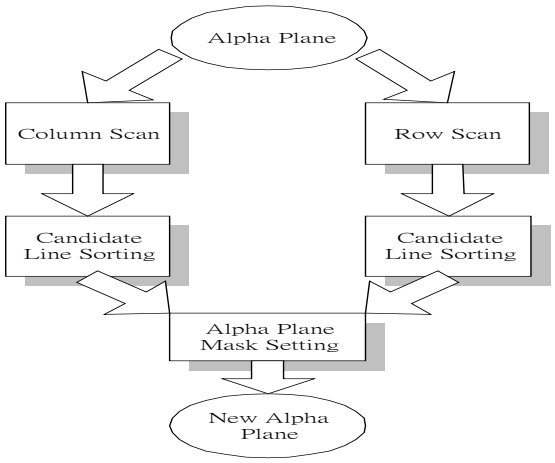
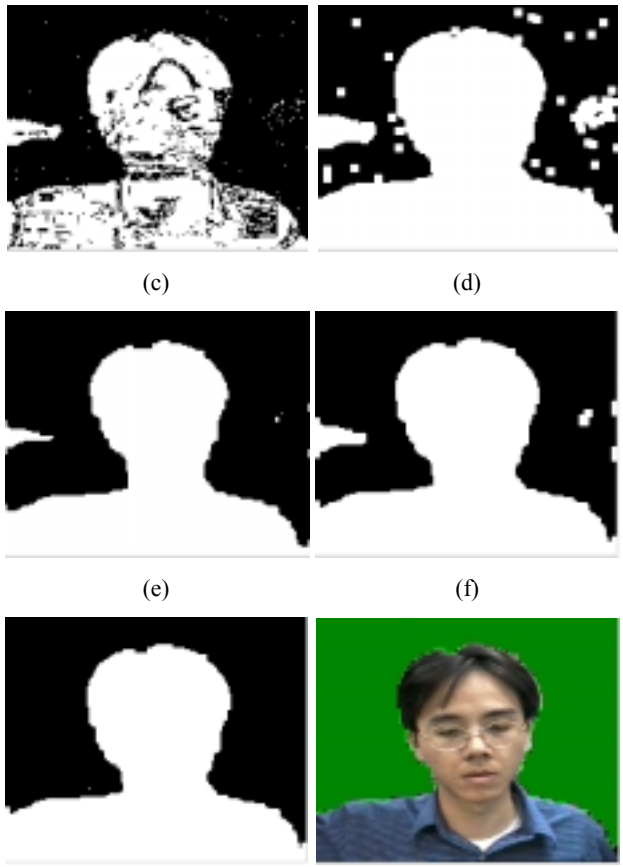


Fig. 8. The processing flow of the propose row-column scan method.

Fig. 9 shows the simulation result of the propose object segmentation scheme. Fig. 9 shows that the segmentation noises outside and inside the object can be effectively removed. Figs. 9 (a) and (b) show the pre-captured background and the image including the foreground object respectively. Fig. 9(c) depicts the rough segmentations results after performing the still background subtraction scheme. The rough segmentation is still quite noisy. The effects of the morphological operations are illustrated in Fig. 9(d)-(f) step-by-step. The small granular noises can be effectively eliminated using the morphological filtering process as shown. In Fig. 9(b), the noise with a larger area (a palm not belonging to the main object) on the left-side the main video object was intentionally added, and it can be removed by the row-column scan scheme. From these results, we see that the proposed method can segment the video objects well from works well for typical video conferencing image sequences in real-time.

(a) (b)



(c) (d)

(e) (f)

(g) (h)

Fig. 9. The simulation result of the proposed object discrimination: (a) One of background frames; (b) frame including foreground object; (c) alpha plane obtained by background subtraction; (d) alpha plane obtained after 2 times of dilation operations; (e) the alpha plane obtained after 4 times of erosion operations; (f) alpha plane obtained after 4 times of erosion operation; (g) alpha plane obtained after the row-column scan method; (h) result of object discrimination.

3.2 Chroma-key Based Video Object Manipulation

To provide the user with more user-friendly features such as object-scaling, repositioning and other object manipulations, and support other features for the virtual meeting environment (e.g., those described in [8-9]) with low computational complexity, we propose to use chroma-key-based video object manipulation scheme. Chroma-key [10] provides an efficient means of object extraction and manipulation which has been widely used in movie and television production. Since chroma-key-based region/object coding does not need to send the contour information such as shape information (e.g., alpha-plane in MPEG-4) and its perceptually lossless representation of region contours, it has been adopted in H.263+ and MPEG-4 (referred to as “material key”) standards as an option for region/object representation.

Consider a video sequence $f(\mathbf{x}, n)$, where $\mathbf{x}$ denotes the spatial coordinate $\mathbf{x} = (x_0, x_1)$, and n denotes the temporal index. Assume there are M foreground regions/objects in a frame: $R_1(n)$, $R_2(n)$, ⋯, $R_M(n)$, and $R_{\text{back}}(n)$ represents the background. In the chroma-keying technique, the background of the image is replaced with a specific color C_0 as follows:

$$g(x,n) = \begin{cases} f(\mathbf{x}, n), & \text{if } \mathbf{x} \in R_m, \quad m = 1, \ldots, M \\ C_0, & \text{if } \mathbf{x} \in R_{\text{back}}. \end{cases} \qquad (1)$$

C_0 can be sent to the decoder as a side information for object extraction in chroma-keyed video in the H.263+ and MPEG-4 standards, thus it can be dynamically changed. For H.263, however, since no such overhead channel is provided, a pre-determined color which is known *a-priori* to the decoder should be used (blue or green colors are often used). Fig. 10 shows an example of a chroma-keyed image.

(a)

(b)

Fig. 10. An example of an image with chroma-keying: (a) original image; (b) chroma-keyed image by replacing the background with a blue color.

Using chroma-key, object extraction in the server side becomes relatively easy. The server classifies each pixel into either the foreground objects or the backgrounds by comparing it with the pre-defined chroma-key pixel value, and filters out the chroma-key-like noises in the video objects using a morphological filter by exploiting the single-connected property.

4. VIDEO OBJECT COMPOSITING USING GEOMETRICAL TRANSFORMATION

Fig. 11 shows the proposed server-client architecture mentioned for virtual meeting presentation. After extracting the video objects, the server transcodes the video objects using the dynamic object bit-allocation method described in [6] and inserts a chroma-key background. The resulting 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 2-D video objects against a 3-D virtual environment according to the 3-D location information of each conferee received from the server.

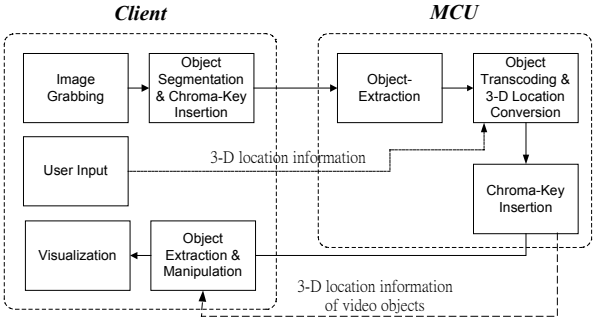


Fig. 11. The proposed server-client architecture for virtual meeting presentation

In a virtual videoconferencing, we need to place different persons at where we wish, against a new 2-D background or a 3-D environment which can even be artificial, allowing even overlap between the people. We would also need to scale the segmented objects, so that a more realistic visualization of conferencing can be achieved. To achieve even more visually pleasing effects, we can tilt the 2-D video objects in 3-D. These features can implemented by using the geometrical transformation discussed below.

We have implemented three operations, 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. Since we allow overlap, we need to fix priorities for the different streams. This priority dictates the overlap to be followed in object compositing.

Translation is achieved by simply changing the 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 look-up 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 3-D. This can be achieved by

"pasting" the frame to an imaginary cube in 3-D and then rotate this cube in 3-D by the two rotation angle parameters, and then take either an orthographic or perspective projection of the 3-D object to 2-D again. However we can approximate this procedure by the simple "rotation" procedure in 2-D itself as described below.

Assume we are given an image and we have to rotate it as shown in Fig. 12. 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',y') of the original figure will be given by the following simple mathematical equations;

$$y' = y - x \times Tan\theta$$
$$x' = \frac{StretchedLength}{OriginalLength}\sqrt{x^2 + (y')^2}$$

with the stipulation that $x>0$. A picture showing the result (incorporating scaling, rotation and translation) against a 2-D virtual background is shown in Fig. 13.

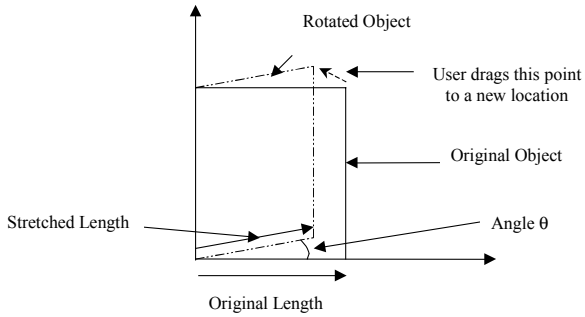


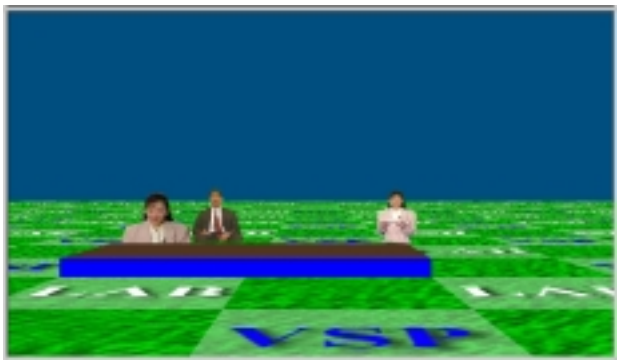
Fig. 12. Illustrating the simple plane image rotation scheme

Fig. 13. Compositing with user control of scaling, repositioning and rotation.

In our implementation, the above simple model adequately achieves the effect of an actual 3-D 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 look-up table. This approach is very computation efficient since the look-up 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 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 further speed up the system, the above-mentioned object manipulation functions can be implemented using the OpenGL™ technologies which are supported by most of the commercial 3-D display card at the client side. The computational load is thus shared by the graphic chips on the display card, thereby most of the computing power of the client terminals can be dedicated to the object segmentation and video encoding and decoding. Fig. 14 illustrates an example of 2-D video objects composited in a 3-D virtual environment by using the OpenGL™.



(a)

(b)

Fig. 14. Two snapshots of the proposed virtual meeting with natural 2-D objects in a synthetic 3-D environment using the OpenGL™ technologies.

5. CONCLUSIONS

We have described the implementation of an H.323/H.324 compatible 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 proposed PPMCU. The proposed PPMCU provides an integrated platform for video, voice, and data communications, and is fully compatible to the H.323/H.324 standards. The proposed PPMCU not only supports the split-screen continuous presence format but also provides advanced personal presence controllable object processing features such as scaling, re-positioning, rotating, and dynamic bit-allocation. We have presented efficient methods for implementing video object segmentation and the user-friendly object processing features. We have implemented a real-time virtual conference system prototype to demonstrate the feasibility of the proposed methods.

6. REFERENCES

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

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

[3] ISO/IEC JTC1/SC29/WG11 "Coding of moving pictures and associated audio MPEG98/W2194". (MPEG-4), Mar. 1998.

[4] ITU-T Recommendation H.324, "Terminal for Low Bit Rate Multimedia Communication".

[5] C.-W. Lin, T.-J. Liou, and Y.-C. Chen, "Dynamic rate control in multipoint video transcoding," *Proc. IEEE Int. Symp. on Circuits and System*, II.17~20, May 2000, Geneva, Switzerland.

[6] 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," *IEEE Int. Conf. Consumer Electronics*, June 2000, Los Angeles.

[7] 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'l Conf. Automatic Face and Gesture Recognition (Nara, Japan)*, pp. 222-227, Los Alamitos, CA, 1998.

[8] M. E. Lukacs and D. G. Boyer, , and M. Mills, "The personal presence system experimental research prototype," *IEEE Int. Conf. Comm*., vol. 2, pp. 1112-1116, Jun. 1996.

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

[10] T. Chen, C. T. Swain, and B. G. Hsakell, " Coding of subregions for content-based scalable video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 256-260, Feb. 1997.