A NOVEL MULTIMEDIA SYNCHRONIZATION MODEL AND ITS APPLICATIONS IN MULTIMEDIA SYSTEMS

Herng-Yow Chen, Nien-Bao Liu, Chee-Wen Shiah, Ja-Ling Wu, Wen-Chin Chen and Ming Ouhyoung
Communication & Multimedia Laboratory
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

Synchronization problem, which always arises when sounds, videos, motion pictures and other media are brought together and integrated into a computer system, is one of the most important issues in multimedia communications and applications. In the time-sharing and multi-process environment such as UNIX operating system, the traditional synchronization mechanism results in two fatal defects, namely, audio discontinuity and out of synchronization between audio and video. In this paper, a novel media synchronization model in a multi-process environment is proposed. Based on this model, a continuous media playback module was implemented and served as the key component of the two multimedia systems developed in the Communication and Multimedia Laboratory of National Taiwan University. One is the multimedia authoring system which provides interactive VCR-like operations and audio/video editing facilities. The other is a prototype of the VOD (Video On Demand) system which provides video browsing facility. Both systems show that the performance of the proposed synchronization model is quite satisfactory.

1. Introduction

A multimedia system combines various information sources, such as text, voice, audio, video, graphics and images, into a wide range of application. It suggests a wide variety of potential applications such as remote learning [1], multimedia mailing system [2], collaborative work systems [3], multimedia communication systems (video phone, conference system [4] and information on demand system [5]), to name just a few. Nevertheless, the complexity of these multimedia systems introduces a number of new technical problems in the field of computer science . It worths noting that these problems basically result from the different features among different media. To solve these problems is one of the major research problems in multimedia research.

The synchronization problem arises mainly due to the fact that the computer systems use hard disks as their storage devices to store all types of data in one-dimensional form. One thus has to expand (digitize) the media source data, which is stored in two-dimensional form in analog storage, into the one-dimensional form in hard disks so that they can be processed in digital systems. The problem of synchronization among media arises consequently when sound, motion pictures and other media are all stored and processed together in the above forms in the computer system.

The purpose of this paper is to develop a robust media synchronization model in application layer and to provide some media synchronization playback modules. These modules serve as the key components of the media playback applications and have been used to implement a variety of multimedia applications in the UNIX environment, including multimedia authoring tool, VOD (Video On Demand) system, and so forth. In the following sections, the basic synchronization principle and the traditional synchronization mechanism is discussed first. Then a

Revised manuscript received December 13, 1994

multi-process (multi-thread) synchronization model will be proposed. Next, the system implementation and some related application systems are presented. Some experiments to show the superiority of the proposed model are then described. Finally, a brief conclusion is given.

2. The Synchronization Principle

The problem of synchronization arises mainly when several related media are to be played back in their corresponding temporal constraints. In analog systems, such as VHS, the rotational speed of tape is used as the reference for timing information. On the other hand, in digital system, the timing information can be obtained intuitively from the size of the audio segment in the digital form (cf. figure 1). To solve the synchronization problem, many different techniques have been proposed in different platforms (PC with DOS or MS-Windows, Workstation with UNIX and X-Windows) [6][7][8][9]. All of these synchronization schemes are based on the idea of aligning the physical location of audio/video data on the storage. Model I describes a simple synchronization scheme proposed in [7].

Model I : [Synchronization by location alignment]

```
loop {

    /* estimate the audio waiting time in advance */
    estimate audio_waiting_time;    /* cf. Eqn. (1) */

    /* show the related audio and video frame */
    play_audio_segment();
    play_video_frame();

    /* waiting for audio data consumed completely by
    the audio device */
    sleep(audio_waiting_time);

} until end_of_playback
```

Eqn. (1) :

$$audio\_waiting\_time = \frac{audio\_segment\_size}{audio\_sampling\_rate} + overhead;$$

$$overhead = data\_access\_time + system\_overhead.$$

Ideally, the audio_waiting_time equals to the size of audio segment divided by the sampling rate of the audio device. Moreover, some overheads include the media accessing time and the instruction execution time should be taken into account in the practical system implementation. In single process environments such as DOS, the interrupt service routine of OS kernel can also be a system overhead. In addition to the above system overhead, the overhead of process context switching is important but is difficult to be predicted in the multiple process environments such as UNIX. As pointed out in [7] (cf. Eqn. 1), the estimation of the data accessing time and system overhead is critical in Model I. If the next playing time interval (i.e. audio_waiting_time) can not be estimated precisely enough, two undesirable phenomena may occur:

(i) Audio discontinuity (cf. Figure 2): If the estimated time interval (i.e. audio_waiting_time) from Eqn.(1) is longer than the real one, it would be too late to write the next audio data segment to the audio device in time. Therefore, the previous audio data segment in the audio buffer will be exhausted before the next audio segment arrives. In the meanwhile, there is no audio data to be played back between these two audio segments. This results in the discontinuous audio output.

(ii) Out of synchronization (cf. Figure 3): If the estimated time interval is shorter than the real one, it would be too early to write audio data to the audio device. In this case, the residual part of previous audio segment is still in the audio device buffer, but the corresponding video frames will be immediately played out by the video devices such as JPEG or MPEG compression hardware. Thus the audio device buffer could be almost full and the phenomenon of "out of synchronization" occurs.

These two shortcomings make the work in [6][7][8] incomplete.

3. The Synchronization Model

In order to overcome the problems existed in Model I, a novel multi-process (multi-threads) media synchronization model (Model II) is presented as follows. Some operating systems, such as UNIX, do not support multiple threads in a process. The others, such as MACH, do support. In our model, a thread could be treated as a process if the multi-thread facility is not supported in the operating system. We used "process" instead of "thread" for the convenience of the explanation.

Model II : [multi-process (threads) model]

(1) Each process is responsible for playing back one medium.

(2) The parent process plays the role of monitoring its child processes and playing back the highest priority medium.

(3) The child processes play back the lower priority media.

(4) The responsibilities of the parent process are :

- pre-calculate the vital synchronization information.
- fork (generate) the child processes before the playback starts.
- kill (terminate) the child processes after the playback ends.

(5) Synchronization mechanism among different media processes: Two approaches can be adopted in the proposed model.

- ***relative synchronization***: [complex method] According to the pre-calculated synchronization information (some synchronization points), media processes can synchronize with each other through some well-known internal process communication (IPC) techniques, such as share memory, sucket, and pipe.
- ***absolute synchronization***: [simple method] According to the pre-calculated information (some time table), each medium process synchronizes with the global system clock .

To make the proposed model clear, we give a simple example in the following, which shows how the synchronization between audio and video is done in our model.

(1) A time-stamp table, which is pre-calculated from the audio segment size, used as the check-points for synchronization. Each element of the table records the starting point (start_time) and the ending point (dead_time) of an audio segment. All of the video frames and their associated audio segments should be played back during their corresponding time interval. (cf. Figure 4.)

(2) Generate a child process for each medium by using system call "fork" [17] from parent process. As shown in figure 5, an audio parent process can generate a child video process and a child text process. Each child process inherits the synchronization information from its parent.

(3) Perform synchronization mechanism described in step (5) of Model II within those child processes. The following pseudo algorithm is used for video playing :

```
Procedure Play_Video
begin
 /* can get current_time from system absolute clock */
 loop
     /* check dead time */
     if (current_time > dead_time of frame i) {
         jump to next appropriate frame j;
         i = j;
     }
     /* check start time */
     if (current_time < start_time of frame i) {
         wait until (current_time = start_time
                         of frame i);
     }
     display video frame i;
 end loop
end Procedure
```

(4) The audio child process should be busy with sending audio data to audio device. The following is its pseudo code :

```
Procedure Play_Audio
begin
    for i = current_frame to end_frame
        play audio data of i-th frame
end Procedure
```

(5) If the user presses the "stop" or "pause" button, all the active child processes are killed by their parent process.

(6) If the user presses the "play" or "continue" button later, steps (1)-(5) are repeated.

In this audio/video synchronization example, one tricky technique, rather than time stamp or internal process communication technique, is used in our implementation of audio process. The synchronization can be achieved due to the following reasons. The output rate of audio device is constant and equals to the input rate of the audio device. Thus, the audio buffer is almost full while audio process is busy with writing audio data. The playback speed of audio data can thus keep pace with the time axis due to the constant output rate of audio device. Therefore, only video process needs the time stamp table to check whether the corresponding video frame should be played back or not. The following shows its checking rules for synchronization (cf. figure 4): the corresponding video frame should

1. be dropped if current_time is earlier than dead_time;
2. wait if current_time is earlier than the start_time;
3. be played back if current_time is in the time interval between start_time and dead_time.

The advantages of the proposed model include:

(a) **Easy to program and debug:** The employment of multi-process (thread) model simplifies the program. It is more intuitive to program each media function than Model I does.

(b) **"Audio discontinuity" is eliminated:** Since the audio process is constantly busy with writing audio data, audio device buffer is always almost full.

(c) **The phenomenon of "out of synchronization" never happens:** The estimation of video-waiting-time in Model I is no longer needed.

(d) **Different media priority can be supported:** The media priority can be supported by assigning different priorities to different processes (threads) in the applications. For example, in the lip synchronization application, it is obvious that the priority of audio is higher than that of video because perception of human is more sensitive to audio than to video. In some subtle applications such as slide presentation (foreground slides and background music), the priority of image or graphic media may be higher than that of the audio data.

(e) **System is robuster and more flexible:** Multimedia applications based on this proposed model become more flexible and adaptive than those in the traditional synchronization model. For example, in our proposed model, an application can kill or suspend some less important media processes (threads) when the system overload is heavy. These processes can be restarted or resumed when system overload becomes light.

4. Implementation

We have implemented a media player module based on this proposed synchronization model. A JPEG based hardware board is used to compress and decompress the video data in real-time. The built-in audio device provides the record/playback audio function in 8 KHz sampling rate. To provide a high disk access speed and large storage space, a disk array is used as the local disk.

The system is currently developed in the UNIX environment using the X Toolkits of X-Windows system [14][15][16] as the graphical user interface. To provide VCR-like interactive operating facility, the UNIX alarm signal (in X Toolkit Intrinsic: XtAddTimeOut) rather than UNIX sleep [17] system function is adopted. Figure 6 shows the hardware and software architecture of the system.

5. Applications

On the basis of the proposed synchronization model, several multimedia systems have been developed in the Communication and Multimedia Laboratory of National Taiwan University. One is a powerful multimedia authoring system, which provides digital VCR-like video operations and KTV facilities in which audio, video and text media can be synchronized. This system provides a friendly and functional complete environment allowing users to do their audio/video editing works. Table 1 summarizes the authoring functionalities provided in this system. Figure 7 shows a photo of the authoring system.

A prototype of VOD (Video On Demand) system has also been developed on the Ethernet network, which is capable of handling simple content-based media queries. Moreover, using a DCT-based video scene detection technique [18] and some video parsing technique [19], a multi-layer video-shot browser was developed for the VOD clients. When a client user requests a movie from the VOD server, he can browse rapidly a number of pre-processing video shots provided by the VOD server to decide whether or not he wants to see the movie. Figure 8 gives a snapshot of the prototype of VOD system.

6. Experiment

To show the superiority of the proposed model, experiments are carried out to test the performance of the model in UNIX workstations with X windows for two critical situations which yield bad performance in the traditional synchronization model. One is in the case of I/O burst situation, such as when the system executes a program with high disk I/O demand. The other one is in the case of CPU burst situation, such as when the system executes many CPU burst processes concurrently. The experiment results show that the performance of the proposed model is quite good in both cases. No audio discontinuity and out-of-synchronization phenomena are observed, even though some video frames are skipped under heavy system load.

7. Conclusion

We proposed a synchronization model that is suitable in both multi-user and multi-process UNIX like environment, and multi-thread MACH like environment. Compared to the traditional approach, the new model enjoys a number of advantages which have been discussed above. Above all, the proposed model is insensitive to the I/O and CPU bursts situations in which the traditional synchronization method does not perform well.

As discussed previously, conventional UNIX system does not support real-time applications because its kernel is non-preemptive and its process scheduling criterion adopts the multi-level feedback with round robin policy. As pointed out in [10][11], the conventional UNIX environment for workstation computing, although useful for many applications, may not be suiitable for high-performance multimedia computing. The main contribution of this paper includes: (1) a novel synchronization model and some related implementation experiences for multimedia computing are presented; (2) a general model is proposed in the non real-time operating system (such as UNIX) to achieve the media synchronization requirement; and (3) based on the proposed synchronization model, a media playback module is developed and has been used as the key component of several multimedia systems .

8. Acknowledgement

This research was supported in part by the National Science Council of Republic of China under the contract no. NSC 83-0425-E-002-140.

References:

[1] Roger C. Schank, "Active Learning through Multimedia", IEEE Multimedia Magazine, Spring 1994 , pages 69-78.

[2] Ming. Ouhyoung, Wen-Chin. Chen, et al.,"The MOS Multimedia E-mail System", Proceeding of IEEE Multimedia, 1994, pages 315-324.

[3] Earl Craighill, Ruth Lang, Martin Fong, and Keith Skinner, "CECED: A System For Information Multimedia Collaboration", Proceeding of ACM Multimedia, 1993, pages 437-446.

[4] William J. Clark, "Multipoint Multimedia Conferencing", IEEE Communications Magazine, May 1992, pages 44-50.

[5] P.Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan, "Designing An On-Demand Multimedia Service", IEEE Communications Magazine, July 1992, pages 56-64.

[6] Shyi-Bang Wey, Chee-Wen Shiah, and Wen-Chin Chen, "Synchronization of Audio and Video Signals in Multimedia Computing Systems", Proceeding of 1992 International Computer Symposium , November 1992, pages 665-669.

[7] Chun-Chuan Yang, Jau-Hsiung Huang, and Ming Ouhyoung, "Synchronization of Digitized Audio and Video in Multimedia System", HD-Media Technology and Applications Workshop, November 1992, pages 2-6.

[8] Yuong-Wei Lei, Ming Ouhyoung, "A New Architecture For A TV Graphics Animation Module", IEEE Transactions on Consumer Electronics, Vol. 39, No. 4, November 1993, pages 797-800.

[9] Lawrence A. Rowe and Brian C. Smith, "A Continuous Media Player", Proc. 3rd Int. Workshop on Network and OS Support for Digital Audio and Video, San Diego CA, November 1992, pages 328-335.

[10] Dick C. A. Bulterman, Guido van Rossum, and Dikter, "Multimedia Synchronization and UNIX", EurOpen Conference, Autumn 1991.

[11] Dick C. A. Bulterman, "Synchronization of Multi-Sourced Multimedia Data for Heterogeneous Target Systems", Proc. 3rd Int. Workshop on Network and OS Support for Digital Audio and Video, San Diego CA, November 1992.

[12] XVIDEO: User's Guide, Parallax graphics Inc., 1991.

[13] XVIDEO: Software Developer's Guide, Parallax Graphics Inc., 1991.

[14] An OPENLOOK at UNIX: a developer's guide to X, by John David Miller, M&T Inc., 1990.

[15] X Toolkit Intrinsics Programming Manual, by Adrian Nye and Tim O'Reilly, O'Reilly & Associates Inc., 1990.

[16] X Toolkit Intrinsics Reference Manual, O'Reilly & Associates Inc., 1990.

[17] SUN MicroSystem "Programmer's Language Guides".

[18] Farshid Arman, Arding Hsu, and Ming-Yee Chiu. "Image Processing On Compressed Data For Large Video Databases", ACM Multimedia, 1993, pages 267-272.

[19] AI Lab, University of Michigan. "Knowledge Guided Parsing in Video Databases", ACM Multimedia Tutorial, 1993.

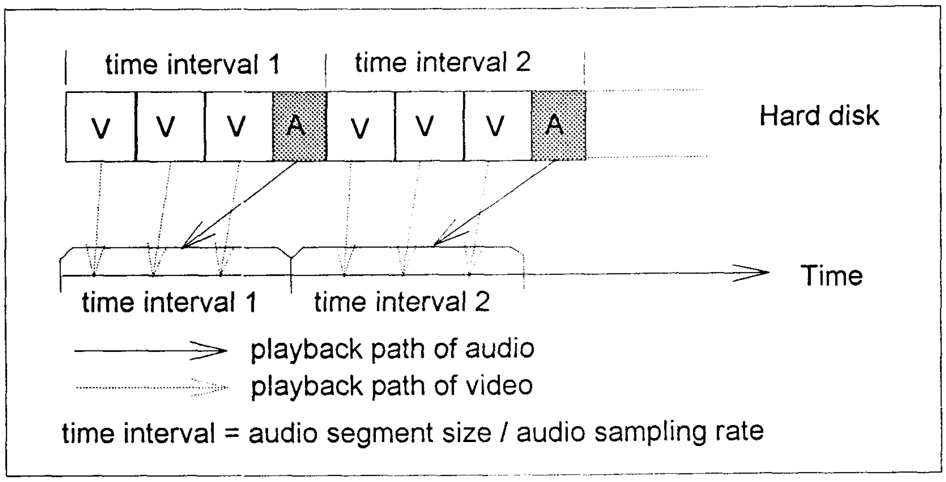


Figure 1. The basic principle of audio/video synchronization.

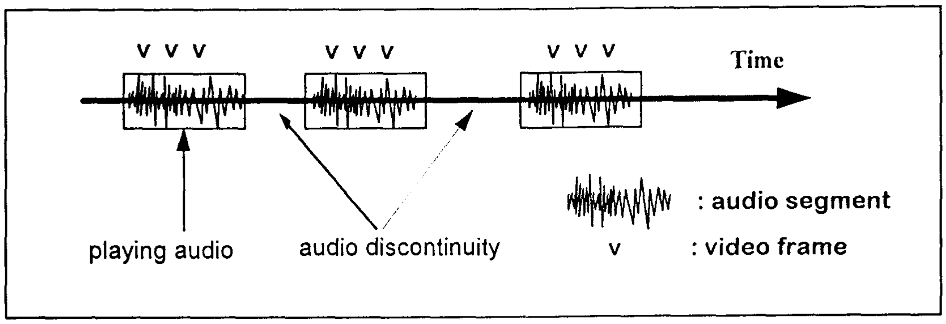


Figure 2. The phenomenon of audio discontinuity due to the fact that the estimated time is longer than the real one.

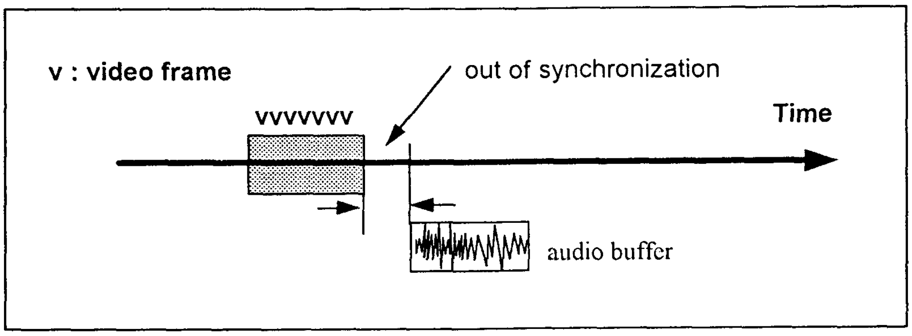


Figure 3. The phenomenon of "out of synchronization" due to the fact that the estimated time is shorter than the real one.

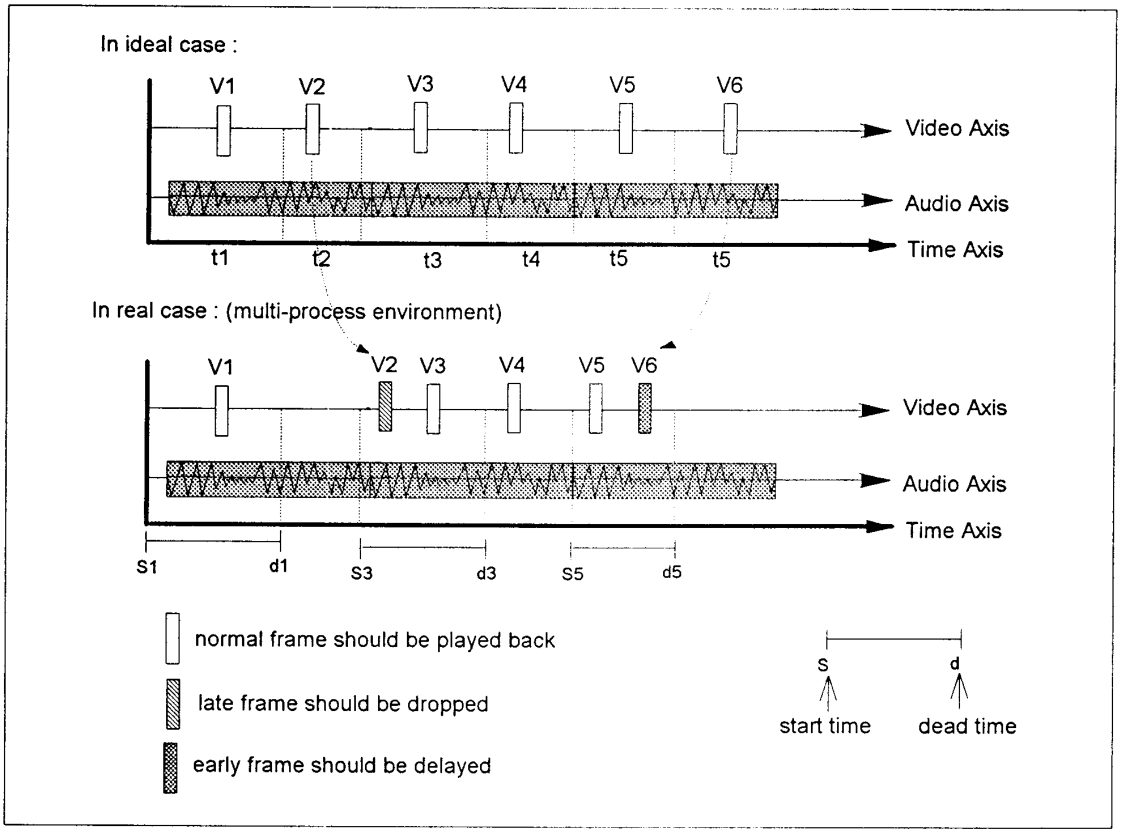


Figure 4. Using the pre-calculated table of time stamp and system clock for audio/video synchronization.

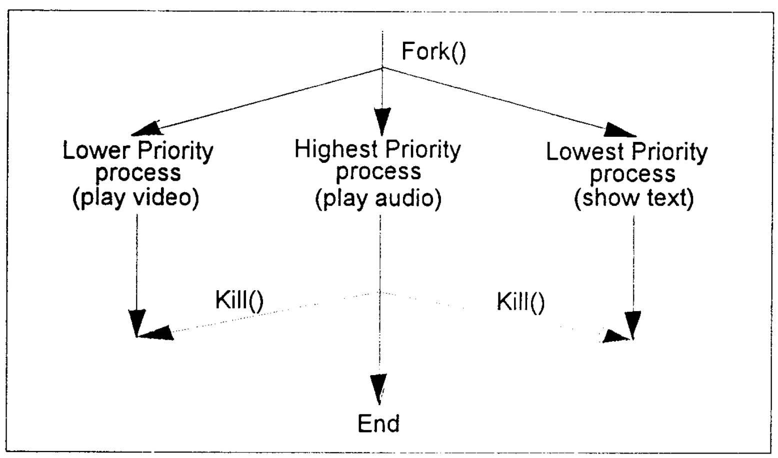


Figure 5. The multi-thread synchronization model

FUNCTIONS	EXAMPLES
Video creation	video record, format convert, etc.
Multi-sources combination	picture in picture, video-graphic composite, etc.
Video concatenation	fad in, fad out, etc.
Visual effect	dissolve, door open, rotation, etc.
Video frame edition	image processing, draw, paint, etc.
Video sequence edition	reverse, cut, paste, copy, search, etc.
Video playing speed adjustment	play, forward, slow, backward, etc.
Video display factor adjustment	hue, saturation, brightness, etc.
Video file management	copy, rename, load, etc.
Video scene browser	decompose video into shots of scenes
Text media synchronization with A/V	KTV facility

Table 1. Functionalities of multimedia authoring system

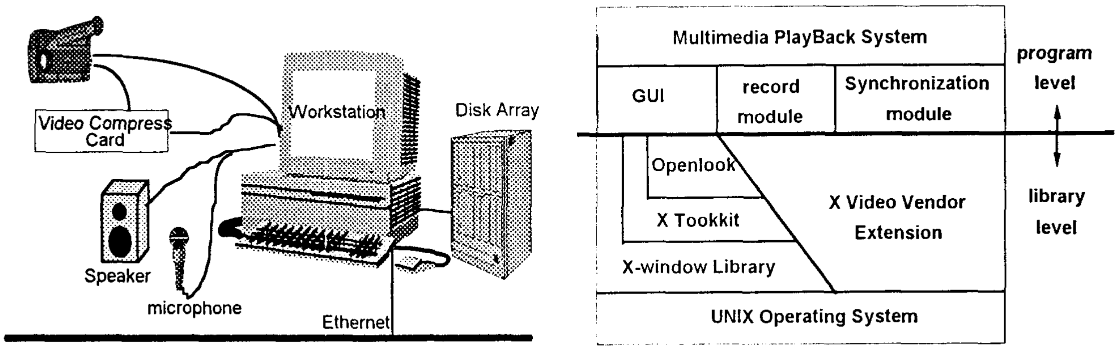


Figure 6. The hardware and software architecture of the proposed media player module

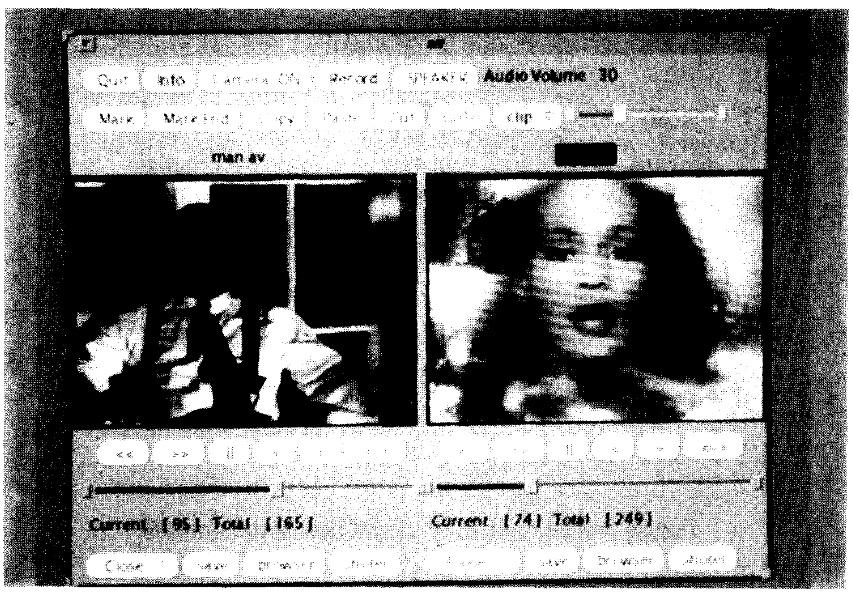


Figure 7 A snapshot of the multimedia authoring system

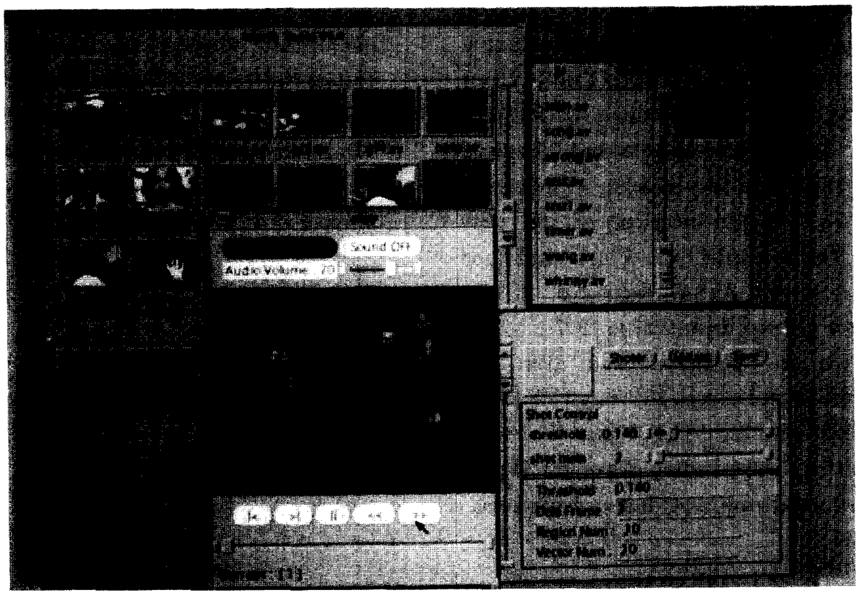
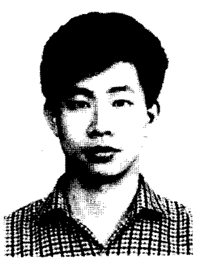


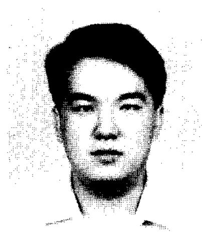
Figure 8. A snapshot of the proposed video on demand system

Biographies



Herng-Yow Chen received the B.S. degree in Computer Science and Information Engineering form Tamkang University, Tamshoei, Taiwan, ROC. He has been a Ph.D. student in the department of Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan, ROC, since 1992.

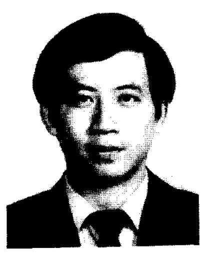
His research interests include Digital Signal Processing, Image Coding, Data Compression, Multimedia Synchronization Modeling and Multimedia Systems. He is a student member of IEEE.



Nien-Bao Liu is a master student in the department of Computer Science and Information Engineering at National Taiwan University.

He received the B.S. degree in Computer Science from the Tatung Institute of Technology in 1991. His research interests include Multimedia Database and Multimedia Synchronization Modeling.

Chee-Wen Shiah is a Ph.D. student in the Department of Computer Science and Information Engineering at National Taiwan University. His research interests include Multimedia Database, Electronic Classroom, Real Time System Design, and Computer Graphics.
He received the B.S. degree in Computer Science form National Chiao-Tung University in 1990, and the M.S. degree in Computer Engineering from National Taiwan University in 1992.



Ja-Ling Wu was born in Taipei, Taiwan, on November 24, 1956. He received the B.S. degree in Electrical Engineering from the Tamkang University, Tamshoei, Taiwan, in 1979, the M.S. and Ph.D. degree in Electrical Engineering from the Tatung Institute of Technology in 1981 and 1986, respectively.

From 1986 to 1987 he was an Associate professor of the Electrical Engineering Department at Tatung Institute of Technology, Taipei, Taiwan. Since 1987 he has been with the Department of Computer Science and Information Engineering, National Taiwan University, where he is presently a Professor.

Prof. Wu was the recipient of the 1989 Outstanding Youth Medal of China and the Outstanding Research Award sponsored by the National Science Council, from 1987 to 1992.

Prof. Wu has published more than 100 technical and conference papers. His research interests include Neural Networks, VLSI Signal Processing, Parallel Processing, Image Coding, Algorithm Design for DSP, Data Compression, and Multimedia Systems.

Wen-Chin Chen received the B.S. degree in Mathematics from National Taiwan University in 1976, and the ScM and the Ph.D. degree in Computer Science from Brown University in 1981 and 1984, respectively

In 1987 he joined the faculty of National Taiwan University where he is currently a Professor of Computer Science and Information Engineering. His research interests include Design and Analysis of Algorithms, Multimedia, and Database systems. He is the co-author of a book entitled "The Design and Analysis of Coalesced Hashing", which was published by Oxford University Press in 1987.

Ming Ouhyoung received the B.S. and M.S. degree in Electrical Engineering from the National Taiwan University, in 1981 and 1985, respectively. He received the Ph.D. degree in Computer Science from the University of North Carolina at Chapel Hill in 1990. He was a member of the technical staff at AT&T Bell Laboratories, Middle-town, during 1990 and 1991. Since August 1991, he has been an Associate Professor in the Computer Science and Information Engineering Department, National Taiwan University. He has published papers on Signal processing and Computer Graphics. He is currently engaged in research in the area of Computer Graphics, Virtual Reality, and Multimedia System. He is a member of ACM and IEEE.