AN AUDIO/VISUAL TOOLKIT FOR SUPPORTING CONTINUOUS MEDIA APPLICATIONS

Herng-Yow Chen[1,2], Yen-Sheng Fu[2], and Ja-Ling Wu[1,2]
[1]Department of Computer Science and Information Engineering
National Chi-Nan University, PULI, Nantou, 545, Taiwan, R.O.C.
[2]Communication & Multimedia Lab.
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, 106, Taiwan, R.O.C.

Abstract

This paper proposes an Audio/Video (A/V) toolkit which can assists programmers to develop continuous media (CM) applications. A deliberate A/V file format and a set of programming interface which are beneficial to random retrieval, film editing and A/V synchronization are proposed as the kernel of the toolkit. Based on the presented A/V toolkit, some practical CM applications such as **A/V Player, A/V EFXer**, and **A/V Browser**, have been developed in the Communication and Multimedia Laboratory of National Taiwan University. These CM applications are easy to be integrated as the complicated multimedia systems.

I. Introduction

Over the last 10 years, the information technologies such as CPU computing power, network bandwidth, storage capacity and compression algorithms have been significantly improved. The advancements of these technologies present rich possibilities of continuous media (CM) applications which demand the critical supports of storage, retrieval, display and manipulation of CM data (e.g., audio and video). The following are some typical CM application:

1. Video phone and video conferencing systems [1]: allow two or more users to hold audiovisual conversation.
2. Computer Supported Cooperative Work (CSCW) systems [2]: allows two or more users to collaborate on work, such as editing a group of documents.
3. Video on Demand (VoD) system [3]: provides digital movie delivery over network.
4. Multimedia E-mail [4] and News Systems [5]: allow users to communicate with each other via multimedia documents.
5. Audio/Video editing [6] and special effecting systems [7]: allow users to make movie/film production.

In the CM applications addressed above, the following three activities (or stages) may be present: (1) ***query*** stage, (2) ***authoring*** stage, and (3) ***presentation*** stage. For example, an interactive scenario in a VoD application may involve both query and presentation activities. Another example is the authoring scenario in a multimedia authoring system. Query stage may be involved first, the authoring and presentation stages are involved after and repeatedly until the satisfactory document is created. Figure 1 illustrates the relationships among these three CM stages involved in CM applications.

In contrast to discrete media (DM) such as text and image, CM are considered much more difficult to maintain and process in digital systems. One reason is that volume of basic access unit (e.g. a video frame) of CM is far larger than that (e.g. a character) of DM, a tiny editing operation of CM data may take a lot of time due to large data movement. Another reason is that temporal constraint of CM data are stricter than that of DM data. Moreover, CM

Manuscript received June 13, 1997

data usually require additional decoding and encoding processes. In general, developing applications on CM data is harder than that on DM data. A well-designed CM toolkit which provides some essential CM operations (such as query/retrieval, playback, editing) for simplify the developing complexity is necessary to CM application developers.

In our previous work, **query/browsing** [8], **authoring** [6] and **presentation** [9] techniques of CM data for the these CM stages have been addressed. In this paper, we focus on the A/V toolkits which have been used for (1) developing CM query/browsing tools (e.g. A/V Browser) which can assist the end-users to find the interested CM data more efficiently, (2) developing user-friendly CM authoring tools (e.g. A/V EFXer) which can help users to compose multimedia documents conveniently, (3) developing robust CM presentation tools (e.g. A/V Player) suitable for multi-tasking environment. A deliberate A/V file format which are beneficial to CM operations such as "random access", "non-linear editing", "efficient reading", "query/browsing", and "media synchronization" have been designed to assist programmers develop various CM applications.

II. A/V Format and Basic Operations

The proposed A/V format consists of three parts: ***A/V Header***, ***A/V Body*** and ***A/V Tail***. For example, a sample video clip named CTV-News.av is anatomized in the Figure 2.

A/V Header stores common information of A/V data stream. Fields of header include a magic ID identifying the file as an A/V clip (i.e. "A/V." string), video resolution (i.e. video width and height), compression information (i.e. quantization factor), other miscellaneous information (such as total frames and total presentation time).

A/V Body follows A/V Header, and stores continuous sequences of video frames and audio segments in an interleaving way (i.e. V1, A1, V2, A2, and so on). This frame-based storage mechanism is adopted based on the following three important reasons:

1. *Effective for retrieval and transmission*: A single sample or several samples (e.g., 20 samples) of audio data is too small a unit to processing or transmission. Not enough audio samples put to the audio device may cause the device buffer underflow during playback, in this case, the annoyed audio-break occurs. Hence, certain amount of audio samples (e.g., 300 sample bytes) are grouped into an segment which should be time-aligned to the video frame occurred at the same time interval.
2. *Easy to achieve the synchronization control and fancy user-interaction:* CM playback requires more precise accuracy in synchronization than that of DM display. Each video frame (e.g. V1) and its associated audio segment (e.g. A1) are tightly time-aligned and placed in the adjacent location, this achieve synchronization control and VCR-like interaction (such as fast-forward, pause, and reverse) easily.
3. *Easy to achieve frame-based processing:* A/V editing or special effect processing are basically frame-based operations.

A/V Tail consists of a table of the retrieving information (location and volume size) for each video frame and each audio segment. In order to save the volume of the tail part, we don't store the recording time for each video and audio unit for synchronization. They can be completely re-constructed by equation (1) according to the size of the associated audio segment and the audio recording sampling rate. For example, a 1000 bytes audio segment under the sampling rate 8000 bytes/sec should have 1/8 sec playback time duration. That is,

playback_duration =audio_size / sampling_rate ...eqn.(1)

The reconstructed timing information is very important for the process of audio/video synchronization.

III. Comparison with other formats

Comparing to the traditional media format, our proposed A/V format consists of three parts rather than two parts in traditional one. One may wonder "Is the tail part of A/V necessary ?" or "Why don't you merge the retrieval and timing information of the tail part into the header part ?". The proposed A/V format has the following two advantages over other formats:

1. *It facilitates random access and VCR facilities:* To illustrate the advantage of our A/V format over the traditional ones, let's consider the MPEG format as an example. The MPEG does not support retrieval information so that random access of MPEG stream is very difficult. This explains why in a MPEG playback system, the fast forward operation is implemented by skipping the whole group of pictures (GOPs). In our design, the retrieval information of media unit is recorded, so the playback system such as A/V Player can easily support random access and some VCR functions such as fast forward or backward.

2. *It facilitates editing and special effects operations:* The editing usually means assembly editing, which consists of the process of deletion or insertion. The special effects generating is a process of modifying CM data, such as scene transition (e.g., fade-in and fade-out) and composition (e.g., chroma-keying). Both the editing and the special effects processing will induce changes of frame sequence number. If the retrieval information is stored in the header part (as done in the traditional A/V formats), a tiny editing operation (e.g., just cuts out one frame or inserts a new frame) will cause tremendous movements of the data. This is clearly very inefficient. In the proposed A/V format, data movements resulted from the editing and the special effects generating are very limited.

The following two paragraphs will present the advantage and feasibility of A/V format for editing process.

1. *Cut operation:* In the proposed A/V format, as the user perform an cut operation on a video sequence, data of the cut out frames, in the proposed A/V format, are not really deleted immediately. Instead, Instead, only the retrieval information of the cut out frames in the tail part is modified to disable the cut frame (the user can not access these cutout frames thereafter). Hence, only a little retrieval information, in the tail part, should be shifted accordingly. Figure 3 illustrates this operation.
2. *Insert operation:* When the user perform an insert operation, the data of the inserted frames do not insert to the location according to the traditional frame sequence number. Instead, the data is inserted to the in-between of the body part and the tail part. Only the new retrieval information of the tail part is updated. As a result, the data movements are limited. Figure 4 illustrates this operation.

Special effects generating process sometimes induces additional transition frames (such as in the case of fading and dissolving). These transition frames are treated as the inserted frames and the same mechanism can be applied.

IV. Software/hardware architecture

In the initial stage, the A/V toolkit was designed and implemented on the SPARC 10 workstation under UNIX and OpenWindows environments. A JPEG-based hardware, the XVIDEO Parallax board, was used to capture and compress the video stream (with 640 x 480 resolution) in real time (25 frames/sec). The system can decompress and display the encoded video frames faster than real time (40 frame/sec). The built-in audio device, with 8 KHz sampling rate u-law audio was used to provide record/playback functions. All the graphical user interface was developed on OPENLOOK and X-Windows Figure 5 shows the hardware and software architecture of the A/V Toolkits.

V. A/V Player: A CM presentation tool

Figure 6 shows the user interface of the A/V Player. The buttons at the bottom of the video display window

provide corresponding VCR controls: play forward and backward, stop, fast forward and fast backward. In addition, a digital slider below these buttons allow direct access to any position in the video. About sound control, a sound ON/OFF toggle and a volume slider is given in the upper part of the display window. The proposed A/V toolkit provided the following two key functions for supporting A/V Player:

1. *Periodical playback with user interaction:* The simplest way to simulate playback A/V in periodic is using a looping statement (such as for-loop or while-loop in C) to enclose the retrieving, playback, and synchronization controls. However, the looping statement is too strict to release the control right for user interaction. To provide the interactive facilities (such as fast forward, rewind and pause in the A/V Player), instead of using the traditional looping statement, the UNIX alarm signal (in X Toolkit intrinsic: XtAddTimeOut) rather than the UNIX sleep() system function was included into the control process so as to return the input control right (such as mouse input focus) to the window manager.
2. *Synchronization control:* In our design, the audio and video data are stored in an interleaving way (c.f. Figure 2) and can easily be retrieved and played. Inter-media synchronization problem has thus been simplified. It is because that the corresponding audio segment and video data are presented at almost the same interval. However, intra-media synchronization problem still have to be investigated. Yang and Huang [11] proposed an intuitive synchronization mechanism which intend to keep the equality of production rate and consumption rate in audio buffer. Unfortunately, in a multi-process environment, Yang's mechanism is feasible in a normal situation, but *"anomaly of synchronization"* may occur when system load becomes heavier (e.g., when running a job with some very intensive computation (CPU bound) or disk access I/O bound) concurrently. In such case, two kinds of perspective synchronization anomaly: *audio-break* and *out-of-sync* between audio and video will occur. A novel synchronization model, Multi-Sync model, had presented in our previous [9], and been implemented into the A/V toolkit. The experiment result shows that our synchronization model performs better than Yang's approach in the heavy load situations.

VI. A/V EFXer: A CM authoring tool

Figure 7 shows the user interface of the A/V EFXer. It simulates the functions of a typical editing machine with two editing channels. We regard each channel as one A/V Player which can open and play some A/V files, individually. In addition, the A/V EFXer provides the function of live recording. The "Camera ON" button is responsible for detecting and digitizing the analog A/V signal (such as NTSC or PAL) from the input device (such VCR). When the input audio and video signal sources are ready, user can press a "Record" button to compress the video by hardware (such as Motion-JPEG hardware), and to digitalize the audio by audio device (such as SUN audio device). All the captured data (audio segments and video frames) are stored in an interleaving way.

In addition to recording, the A/V EFXer is capable of doing various primitive functions. The two channels share an editing buffer- the ClipBoard, and provide some basic edit functions: (1) Cut, (2) Copy, and (3) Paste. The Cut/Copy instruction will move/copy a sequence of frames (including audio and video) into the Clipboard. The boundary of the cut/copied frame sequences are defined by the mark and mark-end buttons. This cut/copy/paste scenario is similar to the clipboard mechanisms used in MS-Windows 3.1/95/NT. Figure 8 illustrates the editing scenario of the proposed A/V EFXer.

The A/V EFXer also provide several special effects. Initially, an effect generating scripts, authored by an experienced user, will guide the A/V EFXer to perform the off-line special effects process. All the special effects are processed on the RGB domain. That is, all the candidate frames will be decompressed first (i.e., in RGB domain) prior to the application of the special effects generating

algorithms (such as fade, dissolve and wipe), and then the processed frames are compressed in Motion-JPEG format. We believe this is the simplest and most intuitive architecture. Figure 9 illustrates this architecture. The following pictures show some snapshots of special effect generated by the A/V EFXer. Figures 10 and Figure 11 show the zoom-in-zoom-out special effect for a circular object and the wipe effect where the second shot flying into the first shot from four corners, respectively.

VII. A/V Browser: A CM browsing tool

The facility of video browsing is necessary for many video services, e.g., the VoD application. Figure 12 shows the possible client-server interaction scenario. Figure 13 shows a snapshot of the proposed A/V Browser. The A/V Browser shows the pre-classified digital movies located in the remote server by a small image icon. The client user may click the desired movie icon to play the contents. If user is interested in many movies, A/V browser can represent the selected movie as many video shots, and deliver these shot icons to users. In addition, granulation of video shots (i.e., shot number) can be dynamically updated by adjusting the shot threshold, controlled by the user.

In our previous work, this multi-layer browsing mechanism provides a scaleable representations of the contents of a video sequence. It can assist the user to rapidly find the movies interested. The A/V toolkit has implemented two key technologies for CM browsing: a DCT-based scene-change detection [8] for efficient video partitioning (in server), and a dynamic shots browsing mechanism (in client).

VIII. Conclusion

We investigated the important issues on the **query/browsing**, **authoring** and **presentation** stages in typical CM applications, and designed a CM Toolkits as a kernel library to help the development of CM prototype systems (**A/V browser**, **AV-EFXer** and **A/V Player**) corresponding with the focused issues on three CM stages (query, authoring, and presentation). Figure 14 illustrates the hierarchical relationships among the key technologies (or the CM libraries), the primitive CM applications, and the advanced multimedia systems. The implementation of these techniques has significantly improved the quality of CM services to various CM applications.

References

[1] Clark, W. J., "Multipoint Multimedia Conferencing," *IEEE Communications Magazine*, pp. 44-50, May 1992.

[2] Craighill, E., et al., "CECED: A System for Information Multimedia Collaboration," *Proceedings of ACM Multimedia*, Anaheim, California, pp. 437-446, 1993.

[3] Ahanger, L. G., Folz, R.J., et al., "A Digital On-Demand Video Service Supporting Content-Based Queries," *Proceedings of ACM Multimedia*, pp. 427-436, 1993.

[4] Ouhyoung, M., and Chen, W.-C., et al., "The MOS Multimedia E-mail System," *Proceedings of IEEE Internal Conference on Multimedia Computing and Systems*, pp. 315-324, 1994.

[5] Chang, K. N., et al., "The MOS Multimedia Bulletin Board System," *Proceedings of International Conference on Consumer Electronics*, Chicago, pp. 315-324, 1995.

[6] Chen, H. Y., et al., "A Novel Audio/Video Synchronization Model and Its Application in Multimedia Authoring System," *Proceedings of International Conference on Consumer Electronics*, Chicago, pp. 176-177, 1994.

[7] Alpert, S. R., et al., "The EFX Editing and Effects Environment," *IEEE Multimedia Magazine*, Vol. 3, No. 1, pp. 15-29, 1996.

[8] Chen, H. Y., and Wu, J. L., "A Multi-Layer Video Browsing System," *IEEE Transaction on Consumer Electronics*, Vol. 41. Num. 3, pp. 842-850, 1995.

[9] Chen, H. Y., Wu, J. L., "MultiSync: A Synchronization Model for Multimedia Systems," *IEEE Journal on Selected Areas in Communications*, Vol. 14, Num. 1, pp. 212-225, January. 1996.

[10] Yang, C. C., et. al, "Synchronization of Digitized Audio and Video in Multimedia System," *HD-Media*

Technology and Application Workshop, Taipei, pp. 26-31, 1992.

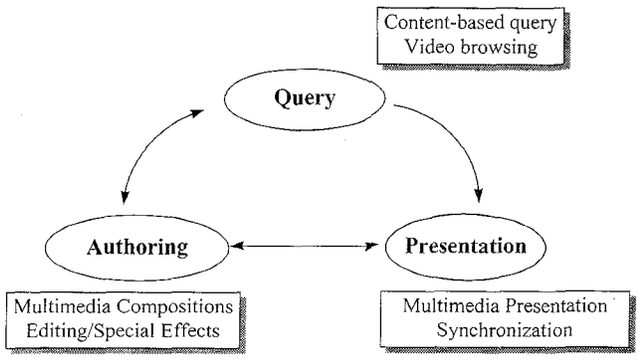

Figure 1. The relationships among the three stages of CM applications.

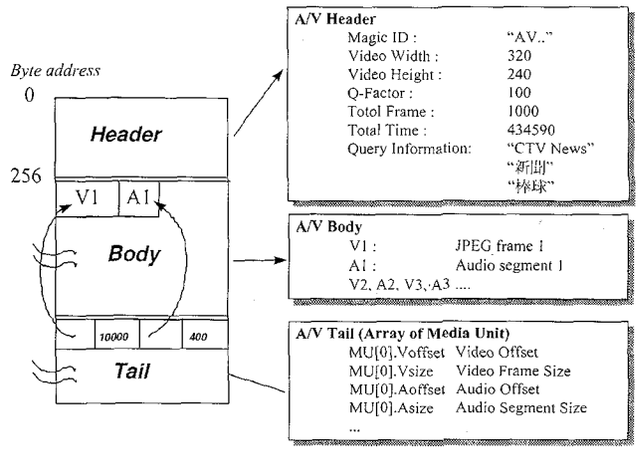

Figure 2. The proposed A/V file format: Header, Body and Tail.

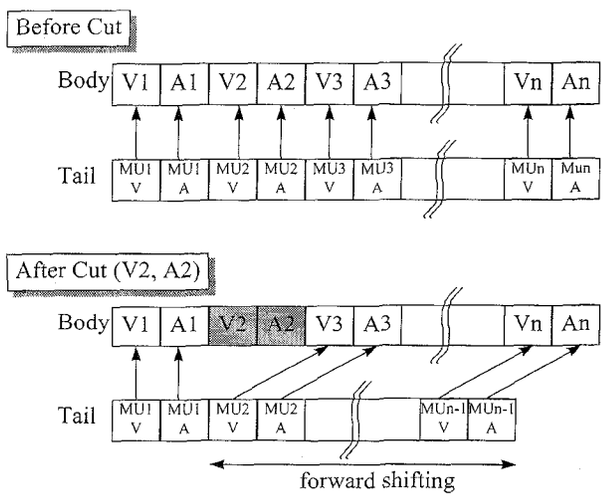

Figure 3. Cutout frames in the proposed A/V format.

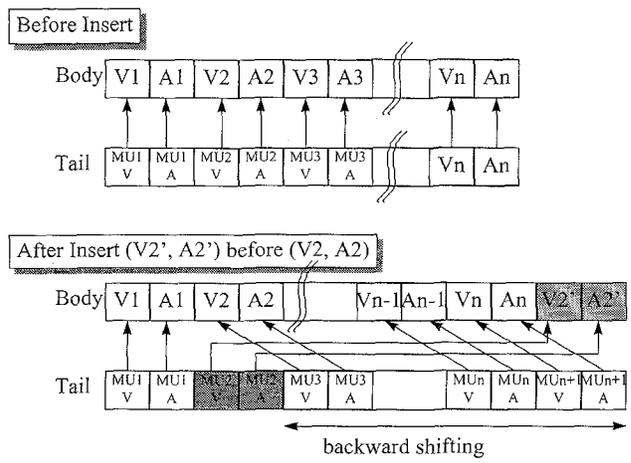

Figure 4. Insert frames in the proposed A/V format.

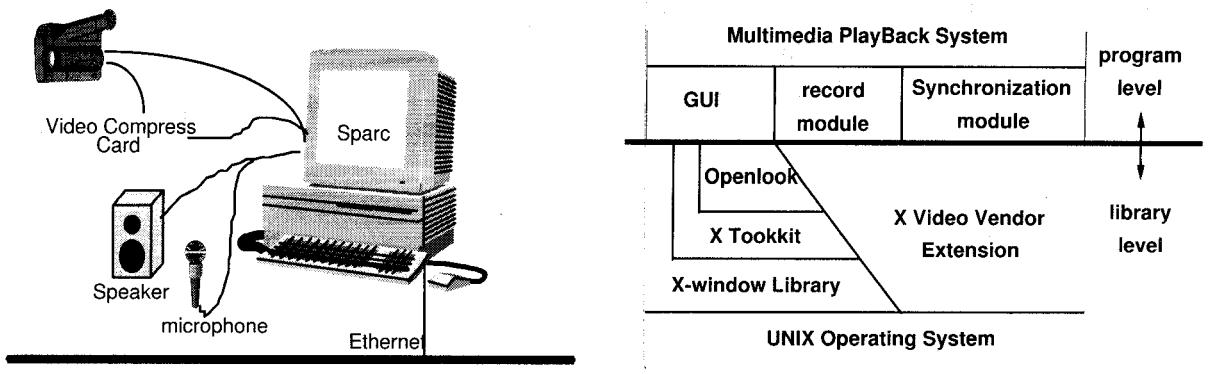

Figure 5. Hardware/Software component of the proposed A/V toolkit .

Figure 6. A prototype of the A/V Player system.

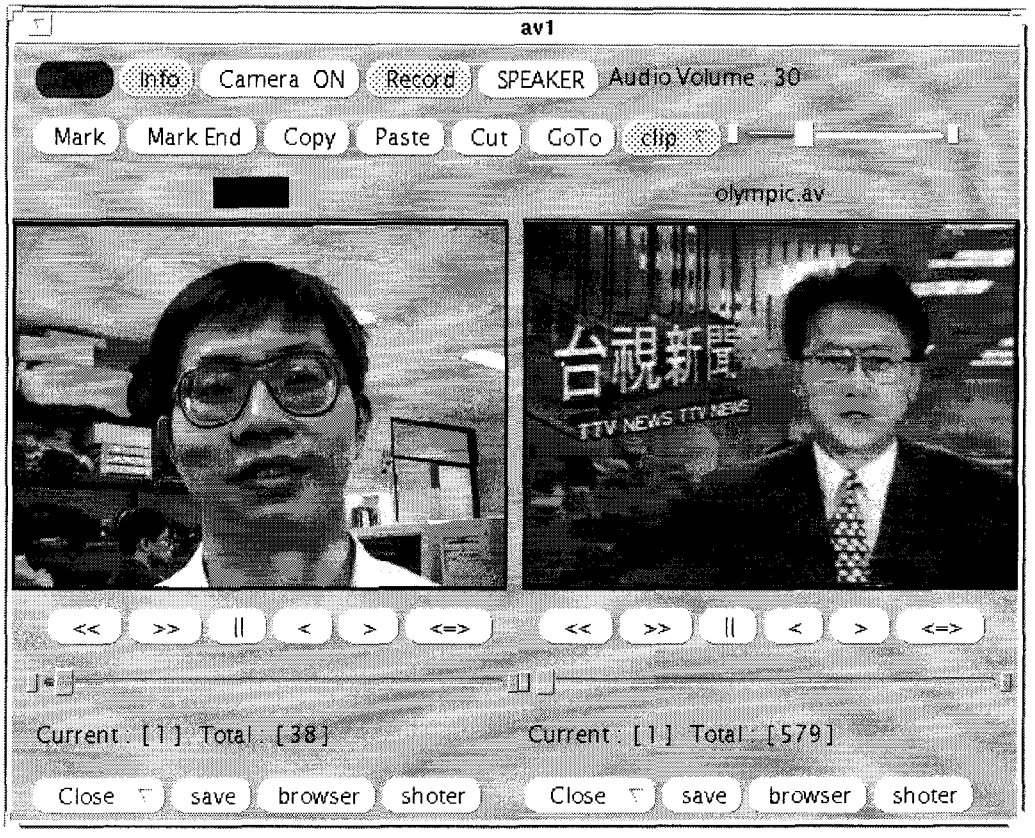

Figure 7. A snapshot of the A/V EFXer system.

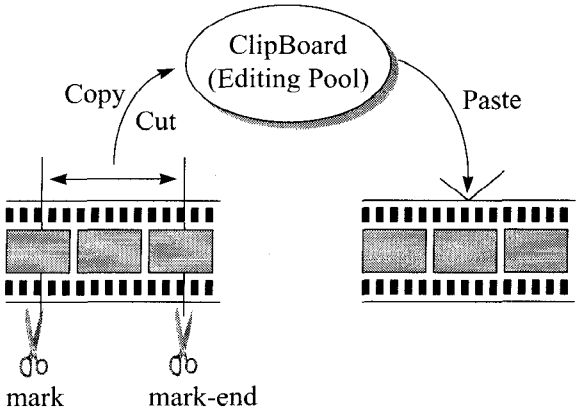

Figure 8. ClipBoard acts as a editing buffer .

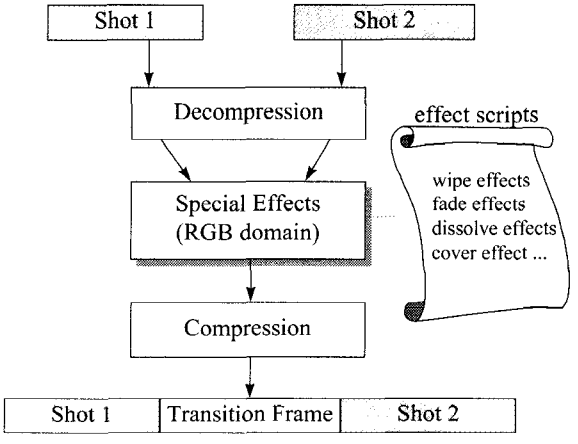

Figure 9. Block diagram of the special effect generation.

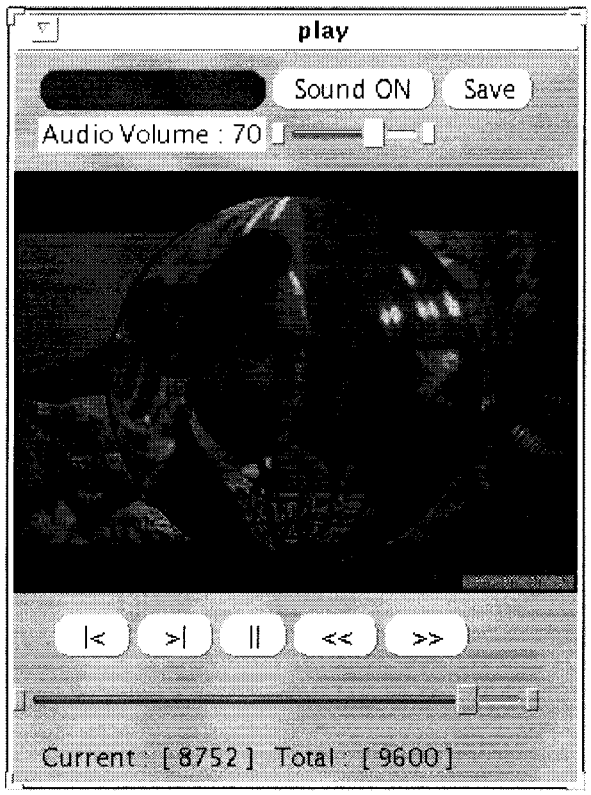

Figure 10. Circle Zoom-In/Zoom-Out special effect.

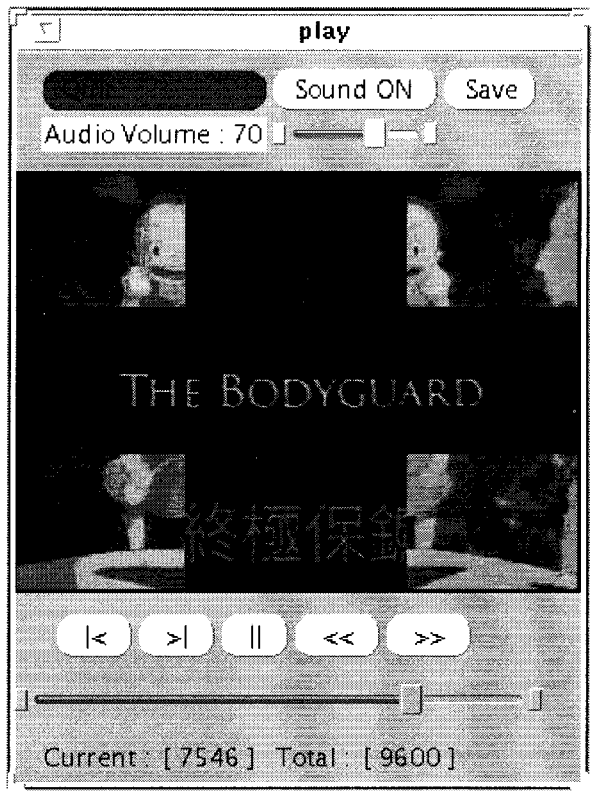

Figure 11. "Flying from the corner" special effect.

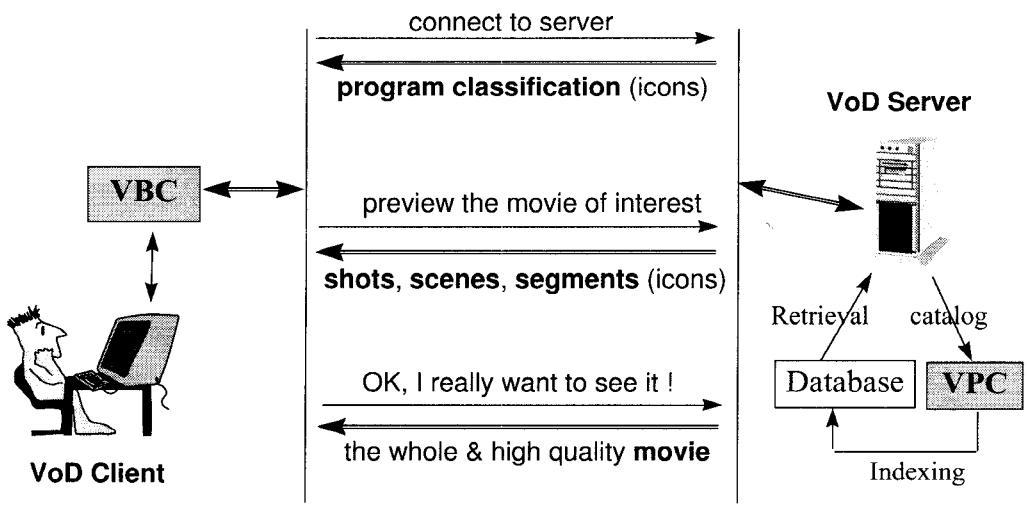

Figure 12. The interactive behavior between a VoD client and server. VBC acts for Video Browsing Component, VPC acts for Video Partition Component.

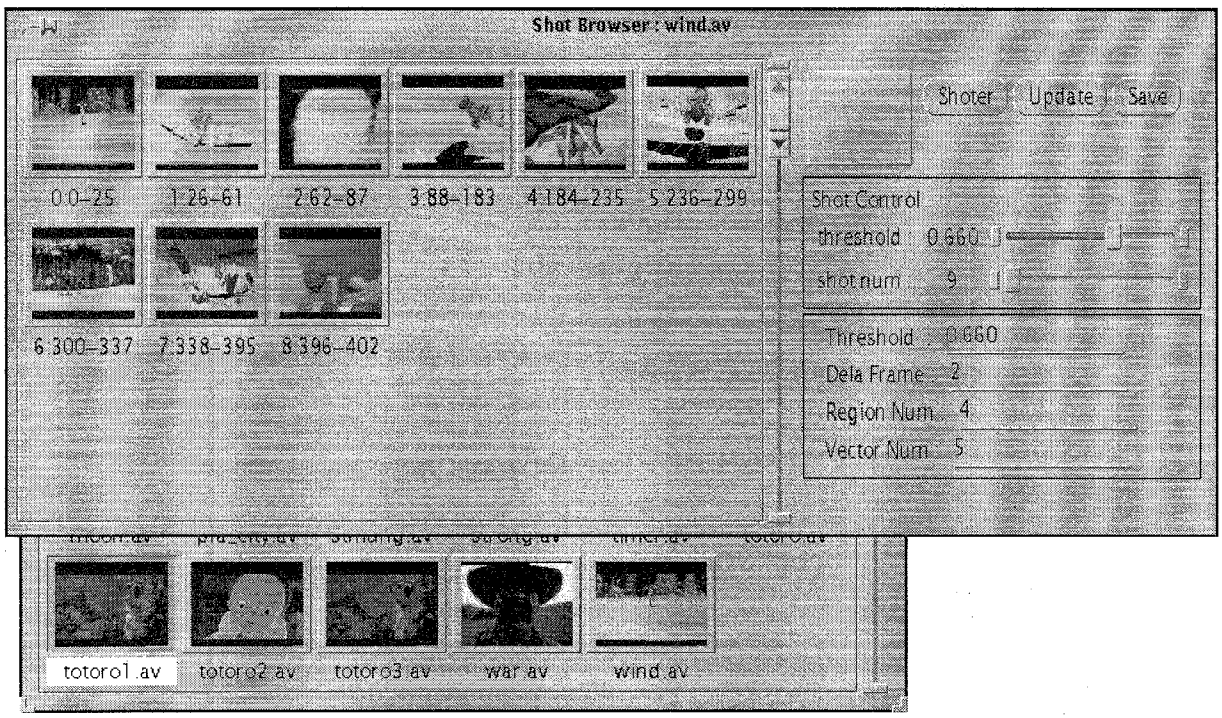

Figure 13. The A/V Browser: dynamic video shots adjust for video browsing.

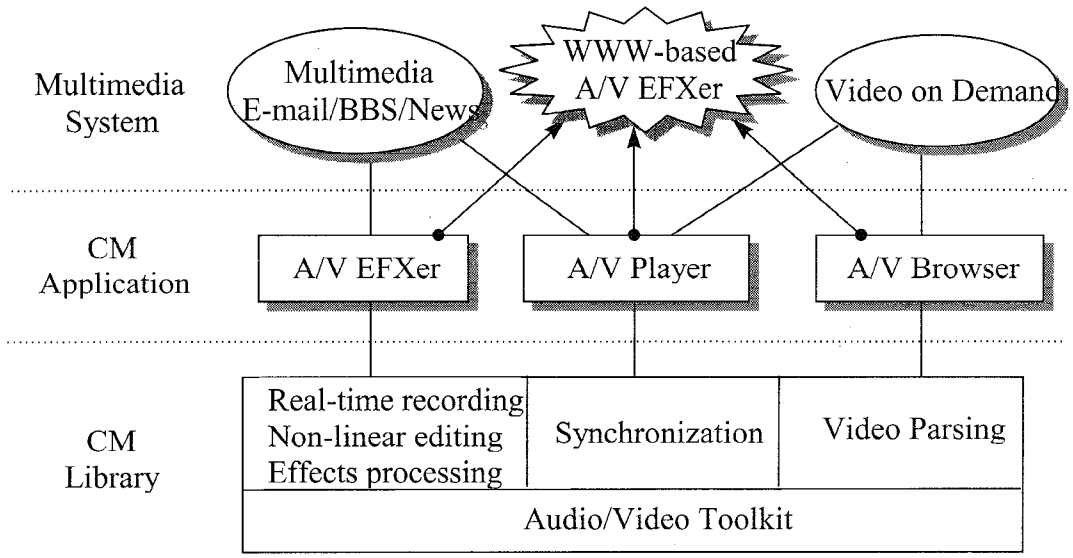

Figure 14. Hierarchical relationships among the CM Library, CM Applications and Multimedia Systems.

Biographies

Herng-Yow Chen was born in Taiwan on June 1969. He received the Ph.D. degree in Computer Science and Information Engineering (CSIE) from National Taiwan University (NTU), Taiwan, R.O.C., in 1997, and B.S. degree in CSIE from Tamkang University, Taiwan, R.O.C., in 1992. He is currently an Associate Professor in the Dept. of CSIE at the University of National Chi-Nan, PULI, Nantou, Taiwan, R.O.C. His research interests include multimedia system, digital signal processing, image coding, and data compression,

Jen-Sheng Fu received the M.S degree in CSIE from NTU in 1997, and B.S. degree in CSIE from National Chiao Tung University in 1995. His research interests include multimedia system, database system and computer network.

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. He 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.