

A LOW-COST EDITOR FOR MPEG-1 SYSTEM STREAMS

Jiun Shiu, Shuo-Chia Cheng, Ja-Ling Wu, Chun-Hung Lin and Jiun-Jie Huang
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, 10764, Taiwan, Republic of China

ABSTRACT —

An A/V (Audio/Video) editing system for MPEG-1 system-layer bitstreams is implemented and described in this paper. By utilizing such a system, users can cut, copy, and paste portions of MPEG-1 coded bitstreams to rearrange the A/V sequences.

1 INTRODUCTION

The international video coding standard MPEG-1 has made it possible to manipulate and store the moving pictures and the associated audio data as one kind of computer data, and to transmit and receive the data over existing and future networks. For constructing versatile applications, an MPEG-1 coded bitstream has been designed to support a number of operations including random access, fast search, reverse playback, error robustness, and editing [1, 2]. A simple application, which is required to produce a high quality video and audio, is to remove some seriously degraded portions in an MPEG-1 coded bitstream before it is put on CDs or distributed in other forms. These degradations might be caused by the encoder, the decoder, or the error in the coded bitstreams. It is surely that one wants to find out the exact source of degradation; however, in a practical situation, it will be much more expensive than the simple cutting methods described in this paper. Moreover, the proposed editing system also provides the copy and the paste functions. With such a system, one can produce his own MPEG-1 A/V files from a library of A/V clips which are stored as MPEG-1 coded bitstreams.

For highly efficient compression, MPEG-1 exploits the temporal redundancy between neighboring frames.

However, the dependence between the video frames in an MPEG-1 video stream prevents an easy implementation of an efficient MPEG-1 editor. Two approaches are possible. The first commonly used approach is to edit in the units of GOP's (Group-of-Picture's). In other words, one can cut or paste GOP's instead of frames. Sometimes, such an approach might be less useful when one wants to cut or paste precisely. Another approach is to decode the MPEG-1 stream, edit the decoded video, and encode it back to an MPEG-1 stream. In this approach, the precise editing is possible. However, several drawbacks occur in the same time. First of all, the encoding process is still slow or expansive. Enormous volume of the decoded video implies a lot of storage and more I/O bandwidth required. It is also expected to have more quality lost in such kind of systems because the bitstreams have been encoded more than once. In our approach, we adopt another method. We don't re-encode the bitstream but support more precise editing functions by considering the dependence between the pictures.

Because the system complexity of an MPEG-1 editor is very high, the MPEG-1 editing systems are rare [1, 3]. In these systems, only the edition of moving pictures is available, and usually only I-pictures are allowed in the video sequences. Nevertheless, our editing system support the A/V editing functions, and all the three picture types (I-, P-, and B-pictures) are allowed in the video bitstreams.

The remainder of this paper is structured as follows. In section 2, the system architecture and the user interface of our MPEG-1 editor are introduced. The im-

plementation methods of each editing function are discussed in section 3. In section 4, the conclusion is given.

2 SYSTEM ARCHITECTURE

Our editing system is implemented in the Microsoft Windows environments on a PC-486 machine. To decode and play the MPEG-1 A/V data, a low-cost MPEG decoding board is required. Because no expensive equipment is needed, the cost of the overall system is low.

Fig. 1 shows the user interface of our MPEG editor. The system provides the typical A/V editing functions, including copy, cut, paste, and play. The users can easily edit their MPEG-1 bitstreams by such a user-friendly editing environment. Several bitstream files can be loaded into the system for editing in the same time. It is required that they have the same frame resolutions, picture rate, audio bitrate, and audio sampling frequency.

The block diagram of overall implementation is depicted in Fig. 2. An MPEG-1 system stream is first scanned for creating some supporting side information if necessary. These side data can be stored in separated files and used in later editing sessions. All editing operations are performed over these supporting data files. In this way, complicated and time-consuming adjustments to the original MPEG-1 system stream can be postponed till the time that one wants to save the edited video. When one wants to store the edited video, a new system stream is created from the original ones and those supporting data files. Some fields of system layer are also adjusted including SCR, PTS and DTS to ensure the correct A/V synchronization at playback.

3 EDITING POLICIES

Due to the dependence between the difference frames in MPEG-1 bitstreams, it is impossible for the editor to cut or paste precisely without the expensive decode-and-re-encode procedure. To cope with this difficulty, the system must adjust the editing range specified by the users. There are two possible strategies which can be used in the editing systems. The first one is an ex-

pansion strategy, and the second one a shrink strategy. For the expansion strategy, the system will extend the editing range to ensure the decoding of the residual portion does not depend on the selected pictures. While in the shrink strategy, some specified pictures will be excluded from the editing range to avoid the dependancy. Fig. 3 gives an example for a cut operation to show the difference of these two editing strategies. In this paper all the picture sequences are depicted in the display order. In Fig. 3(a) a range for cutting is specified. The results for the two strategies are shown in Fig. 3(b) and (c), respectively. In our current implementation, all the editing functions are implemented based on the expansion strategy.

For the convenience of describing the editing policies for different editing operations, several mathematical notations are defined as follows.

Def-1: A video sequence S is a set of pictures, that is $S = \{F_0, F_1, F_2, \dots, F_{N-1}\}$, where $N > 0$ is the picture number of S .

Def-2: A video sequence S can be divided into several sequence portions. It is represented as $S = S_0 \cup S_1 \cup \dots \cup S_{M-1}$, and $S_i \cap S_j = \phi$, for $0 \leq i < j \leq M - 1$.

Def-3: A sequence portion A depends on the sequence portion B , which is represented as $A \rightarrow B$, if there are some pictures in A referring to some other pictures in B .

Def-4: If a sequence portion A can be decoded without referring to the sequence portion B , then it is represented as $A \overset{\times}{\rightarrow} B$.

Def-5: For a video sequence S , its picture number is represented as $|S|$.

Def-6: Assume F is a coded picture contained in a video sequence, two weighting functions W_1 and W_2 are defined as,

$$W_1(F) = \begin{cases} 0, & \text{if } F \text{ is a B-picture,} \\ 1, & \text{if } F \text{ is either an I- or P-picture,} \end{cases} \quad (1)$$

$$W_2(F) = \begin{cases} 0, & \text{if } F \text{ is either a P- or B-picture,} \\ 1, & \text{if } F \text{ is an I-picture.} \end{cases} \quad (2)$$

3.1 Cut/Copy Operation

Sometimes it will be necessary for users to cut damaged or improper portions of the video sequences. Assume there is a video bitstream S which contains N pictures $\{F_n | 0 \leq n \leq N - 1\}$ in the display order, and S_2 contains the pictures within the user specified cutting range $\{F_n | i \leq n \leq j - 1\}$ as shown in Fig. 4(b). The sequence portion S_2 should be extended to be S'_2 if there is any dependence between S_1 and S_2 , or between S_2 and S_3 . The sequence portion S'_2 should satisfy the following conditions,

- (1) $S_2 \subseteq S'_2$,
- (2) $S'_1 \xrightarrow{X} S'_2$ and $S'_3 \xrightarrow{X} S'_2$,
- (3) minimal $|S'_2|$.

To satisfy the above required conditions, S'_2 should be extended from S_2 as follows.

- (1) If $\sum_{F \in S_2} W_1(F) = 0$, then $S'_1 = S_1$, $S'_2 = S_2$, and $S'_3 = S_3$.
- (2) If $\sum_{F \in S_2} W_1(F) \neq 0$, then $S'_1 = S_1 - \{F_n | k \leq n \leq i - 1\}$, $S'_2 = S_2 + \{F_n | k \leq n \leq i - 1 \text{ or } j \leq n \leq l - 1\}$, and $S'_3 = S_3 - \{F_n | j \leq n \leq l - 1\}$, where, $W_1(F_{k-1}) \neq 0$, $\sum_{k \leq n \leq i-1} W_1(F_n) = 0$, $W_2(F_l) \neq 0$, and $\sum_{j \leq n \leq l-1} W_2(F_n) = 0$.

In other words, the sequence S_2 will not be extended if it contains only the B-pictures. Otherwise, the cut range should be modified according to the above policies to make sure the retained bitstream can be decoded correctly.

When a sequence portion is cut from the original sequence, it is moved to a clipboard implemented in the editing system. The sequence portion might be pasted to another sequence from the clipboard if it is not yet overlapped by another sequence portion.

In addition to the cut operation, users can copy a sequence portion and then paste it to another sequences. When a sequence portion is copied, it is duplicated and the copy is placed in the clipboard. There is no additional modification should be performed on the user specified copy range.

3.2 Paste Operation

When users select the paste operation, the sequence portion in the clipboard is pasted to the destination. For the paste operation, both the source sequence which contains the clip sequence portion in the clipboard and the target sequence where the clip sequence portion will be inserted should be taken into considerations. The clip sequence portion could include more pictures from the source sequence to make the pasted sequence portion be decoded correctly. Moreover, all the dependancy between the pictures in the target sequence should be maintained after the clip sequence portion is inserted. Although the sources of the clip sequence portions might come from two different kinds of operations, the copy or paste operation, they don't have different paste operating methods.

Assume a clip sequence portion S_2 will be pasted to a sequence, then S_2 should be extended to S'_2 (as shown in Fig. 5). The extension of pasted pictures should satisfy the following conditions.

- (1) $S_2 \subseteq S'_2$,
- (2) $S'_2 \xrightarrow{X} S'_1$ and $S'_2 \xrightarrow{X} S'_3$,
- (3) minimal $|S'_2|$.

The sequence S'_2 can be generated by extending the sequence S_2 according to the pasting policies,

- (1) If $W_2(F_i) = 1$, then $S'_1 = S_1$.
- (2) If $W_2(F_i) = 0$, then $S'_1 = S_1 - \{F_n | k \leq n \leq i - 1\}$, where $W_2(F_{k-1}) = 1$ and $\sum_{k \leq n \leq i-1} W_2(F_n) = 0$,
- (3) If $W_1(F_{j-1}) = 1$, then $S'_3 = S_3$.
- (4) If $W_1(F_{j-1}) = 0$, then $S'_3 = S_3 - \{F_n | j \leq n \leq l - 1\}$, where $W_1(F_{l-1}) = 1$ and $\sum_{j \leq n \leq l-1} W_1(F_n) = 0$,
- (5) $S'_2 = S - S'_1 - S'_3$.

After the paste range has been determined, the editing system must find the insertion point which is expected to be close to the point indicated by the users. Assume the paste target sequence is divided into two sub-sequences by the specified insertion point as shown in Fig. 6(a), then the system must modify the insertion point, which divides the target sequence into sub-sequences S'_1 and S'_2 (as shown in Fig. 6(b)), to satisfy the following conditions,

- (1) $S'_1 \xrightarrow{\Delta} S'_2$,
- (2) $S'_2 \xrightarrow{\Delta} S'_1$.

In our approach, the closest I-picture to the specified insertion point is found and set as the new insertion point. The clip sequence portion will be inserted before the insertion point. The method can be described as,

- (1) If $W_2(F_i) = 1$, then $S'_1 = S_1$ and $S'_2 = S_2$,
- (2) If $W_2(F_i) = 0$, then $S'_1 = S_1 + \{F_n | i \leq n \leq j - 1\} - \{F_n | k \leq n \leq j - 1\}$, $S'_2 = S_2 - \{F_n | i \leq n \leq j - 1\}$, where $W_2(F_j) = 1$, $\sum_{i \leq n \leq j-1} W_2(F_n) = 0$, and $\sum_{k \leq n \leq j-1} W_1(F_n) = 0$.

It should be noted that the neighboring B-pictures before the insertion point will be cut to cope with the necessary condition (1).

3.3 Editing of Audio Frames

When the edited pictures are determined, the corresponding audio frames should be edited. In an MPEG-1 system stream, there is no exact positional relation between a video picture and its corresponding audio frames. The synchronization of these two media is controlled by the presentation time stamp (PTS) and the decoding time stamp (DTS) which are encoded in the packet headers. In our work, the audio frames with the PTS values most closed to that of the first and the last pictures in the selected range will be choosed as the first and the last audio frames for an editing operation.

By modifying the associated PTS and DTS values in the packet layer headers, the edited video can be synchronized with the edited audio.

For the time related audio/video packets before which several packets have been cut by the editing system, their PTS and DTS values must be decreased. On the other hand, the PTS and DTS values should be increased for those packets if some other packets have been inserted before them.

Because the time duration of a video pictures and its corresponding audio frames would not be identical, the positions of the audio frames and the video pictures would not be aligned. Suitable audio frames are selected from the packets that the time duration of the

audio portion is nearly equal to or a little less than the time duration of the video portion. To make sure the synchronization of the retained audio/video packets will not be affected by the editing operations, the PTS values of the video packets will be increased to make the retained packets aligned as they do before the editing operations. Fig. 7 gives an example. Because human beings are more sensitive to the audio signal, the slight delay of the edited video signal would not be distinguished.

3.4 Modification of Headers

In the MPEG-1 video stream, various kinds of header data are defined. Because the editing unit in our system is a picture, the headers corresponding to the edited picture sequences, including sequence headers and group-of-picture (GOP) headers, should be modified. When a clip video sequence is inserted into a target sequence, to avoid the inconsistency of these two headers between these sequences, the headers associated with the clip video sequence should be inserted before the clip sequence, and the headers associated with the target sequence are copied and inserted after the clip sequence such that the last portion of the target sequence can be decoded with the correct header information. However, we do not include the above operations for saving the processing time, because the sequence headers and the GOP headers are usually the same in the general MPEG-1 video bitstreams. In other words, we assume that all decoded video sequences will have the same headers.

Another field that must be modified after the editing operations is the temporal reference numbers in the picture layer headers. When the pictures are either inserted or deleted in a GOP, the temporal reference numbers for the pictures in this GOP should be rewritten.

3.5 Preview of the Edited Bitstreams

The edited bitstream files which locates on the disk are not modified when they are being edited. All the editing effects are recorded in the side information file which

specifies the bitstream structure. When the edited bitstreams would be saved to the file disks, the editing system refers to the side information and gets the required data from the original bitstream files to form the new bitstream files.

The editing system provides a preview-before-save function for building a practically useful system. Similar to the save function, the editing system generates packets from the side information and the original bitstream files and delivers them to the decoding hardware for playing. After various editing operations, the resultant bitstream might be composed of various portions from different bitstream files. Such an access pattern to the hard disk implies tremendous seek time which might prevent smooth playback. To cope with such a nonlinear access pattern, three 64 Kbytes look-ahead buffers are used as cache for disk access. These three buffers will hold data from current video packets, current audio packets, and next audio or video packets, respectively. This simulated retrieving of edited bitstream can start from any I-picture in the bitstream. By this feature, we can implement random accesses to the edited bitstream.

4 CONCLUSION

More and more audio/video applications are implemented according to the MPEG-1 standard due to the efficiency and the flexibility of the coding methods. Meanwhile an MPEG-1 audio/video editor is rarely found which is needed for a variety of related applications. A low-cost MPEG-1 audio/video editing system has been implemented and introduced in this paper. The system provides nearly precise audio/video editing functions that facilitates the editing of the MPEG-1 bitstreams. It is not required to decode and re-encode the original bitstreams. The edited bitstreams can also be previewed smoothly in the system to make sure the quality of the editing effect.

References

- [1] *ISO/IEC 11172:1993, Information technology - Coding of moving pictures, and associated audio for digital storage media up to about 1.5 Mbit/s.*
- [2] D.L. GALL, "MPEG : A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 34, No. 4, Apr. 1991, pp. 47-58.
- [3] Phillip Lougher, *MpegUtil version 1.0*, Internet Free Software, Apr. 7, 1995.

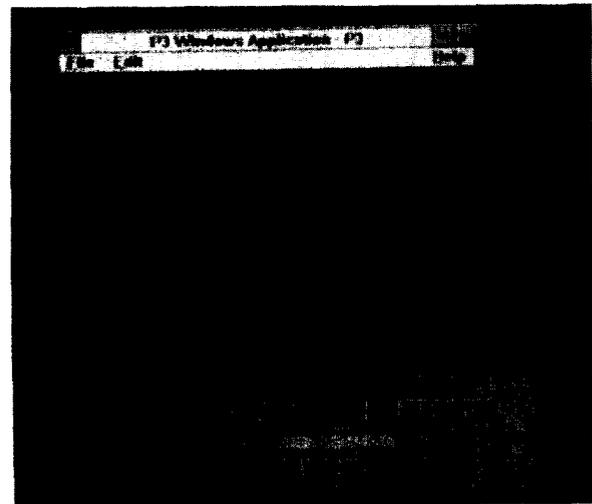


Figure 1: MPEG editor user interface.

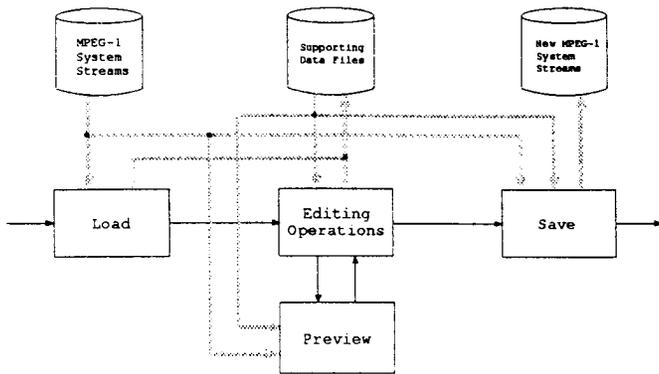


Figure 2: The block diagram of the MPEG-1 system stream editor.

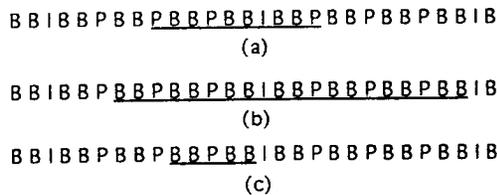


Figure 3: Comparisons of different editing strategies, (a) the cut range specified by the user, (b) the result of the expansion strategy, (c) the result of the shrink strategy. (Each character represents a picture in the bitstream, and the cut range is marked in the figure by the underline.)

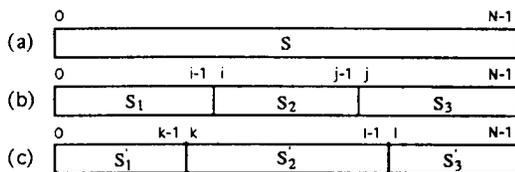


Figure 4: Modification of the cutting range. (a) S represents the video bitstream to be edited. (b) S_2 is specified by the users and S is divided into three sub-bitstreams. (c) S_2 must be extended to be S'_2 , such that S'_1 and S'_2 can be decoded while S'_2 being cut.

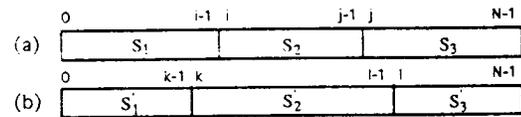


Figure 5: Modification of the paste range. (a) S_2 is specified by users to be pasted to another portion of the sequence. (b) The extension of the specified paste range.

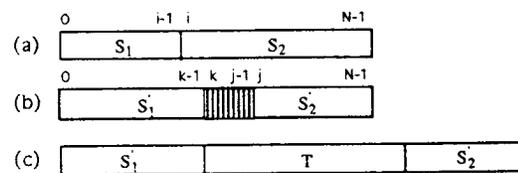


Figure 6: Selection of the insertion point. (a) The insertion point specified by the users. (b) The modified insertion point will satisfy the dependence constrains. (c) A sequence T is inserted into the target sequence.

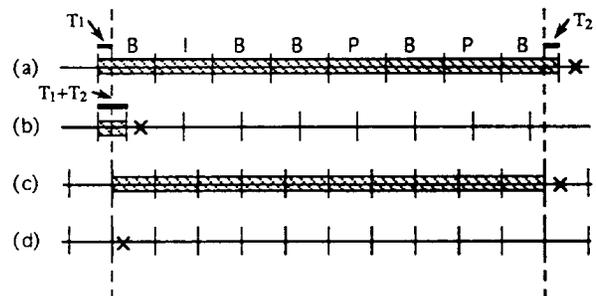


Figure 7: Modify the PTS values of the video pictures to align the retained portions of the both media, (a) the video pictures before cutting, (b) the video pictures after cutting, (c) the audio frames before cutting, (d) the audio frames after cutting.