

## AN INTERACTIVE MPEG-1 BROWSING SYSTEM

Jiun-Jie Huang,<sup>1</sup> Chun-Hung Lin,<sup>1</sup> Ja-Ling Wu,<sup>1,2</sup> Jiun Shiu,<sup>1</sup> Yuh-Ming Huang,<sup>1</sup> and Chiu-Ting Hsu<sup>1</sup>  
<sup>1</sup>Communication and Multimedia Lab., Department of Computer Science and Information Engineering  
National Taiwan University, Taipei, 107, Taiwan, R.O.C.  
<sup>2</sup>Department of Information Engineering, National Chi-Nan University, Puli, 545, Taiwan, R.O.C.

### ABSTRACT —

In this paper, a personal computer (PC) based interactive MPEG-1 browsing system is proposed. The users can get the general information of MPEG video bitstreams, such as the encoding structure and the header parameters of each encoding layer via this browsing system. The users can also get the detailed information of each decoding step. With the aid of the proposed browsing system, users will have the ability to analyze any MPEG-1 bitstream easily. In addition, this system also provides the capability to encode MPEG-1 video bitstreams and to adjust the encoding parameters interactively when the quality of the encoded video is not good enough. The browsing system can also be used as an useful tool during the development stage of an MPEG encoding or decoding system.

## 1 INTRODUCTION

Nowadays, in the multimedia era, MPEG is the widespread industrial standard of video compression [1]–[3], and gradually prevailed in our daily life. The MPEG codec and the related products have become a popular topic. During the developing process of these products, various kinds of bugs might occur. It is very difficult to find where the bugs reside sometimes. A convenient and powerful debugging tool is therefore required for shortening the developing time. In the proposed browsing system, users can get the header information and the encoding structure of each layer to analyze the MPEG-1 coded bitstreams. Moreover, they can get the decoding data of each decoding step to debug their MPEG-1 codec step by step. By this way, users can find and fix the bugs quickly.

Although more and more MPEG video programs are available, it is possible that the quality of the presented video programs is too rough to be accepted by the picky spectators. The video quality of the encoded bitstreams depends entirely on the selection of encoding parameters. Therefore, the decision of the encoding parameters plays an important role in MPEG encoding. However, the decision rule is usually unalterable and general for the whole video in the encoding process because most of the available MPEG encoding systems are not subtle enough to adjust the parameters. Therefore, the rule might not fit all the conditions. In the proposed browsing system, users can encode MPEG-1 video by setting different arguments. The users can also adjust the encoding parameters of the macroblocks for each video frame interactively. The visual result of the modified macroblocks can be shown on the screen immediately; therefore, users can choose the appropriate parameters to improve the video quality. The adjustment of parameters compensates the defects of the encoded video sequences caused by inappropriate selection of encoding parameters. The browsing system can also be used as an useful tool to explore the relationship between the encoding parameters and the video quality.

## 2 MPEG-1 BROWSING SYSTEM

The system diagram of the proposed MPEG-1 browsing system is depicted in Fig. 1. The system consists of five functional units: **video encoder**, **structure analyzer**, **parameter analyzer**, **video player**, and **encoding parameter adjusting unit**.

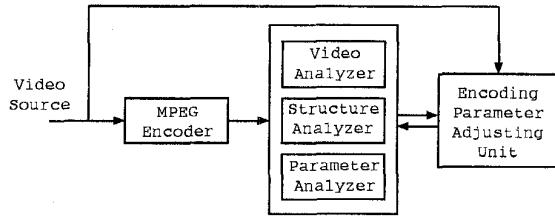


Figure 1: The system diagram of the MPEG-1 browsing system.

These functional units, except the encoding parameter adjusting unit, can be independently operated. For example, users can only use the structure analyzer to analyze the structure of their video files. Therefore, the system can be used for different purposes.

If users use the MPEG-1 browsing system as a debugging tool, they can take advantages of the video player, structure analyzer and parameter analyzer to analyze the MPEG-1 bitstreams. For the necessity of debugging, these functional units all support the feature of random access.

When users want to generate MPEG-1 video bitstreams, they can use the MPEG encoding unit of the system to do this, and then use the video player of the system to examine the video quality. If there are some defects in the coded video frames, users can adjust the encoding parameters interactively to refine the video quality.

The features of each functional unit are described as follows.

### Video Player

The video player unit can be used to examine the video quality of MPEG-1 bitstreams and to check which video frame should be response for the problems appeared.

### Structure Analyzer

The structure analyzer unit shows the structure of the bitstreams as a tree structure and reports the encoding type of each tree node; therefore, users can understand the structure of each encoding layer. These information can be used to check the validness of a given MPEG-1

bitstream. Users can also move the scope from one tree node to another through the embedded structure.

### Parameter Analyzer

The parameter analyzer unit shows the header parameters of each layer and dumps the block data for each decoding process. In the debugging process, users can easily locate the decoding/encoding step where the bugs occurred by checking the decoded parameters.

### MPEG Encoder

The MPEG encoder unit of this browsing system can encode an MPEG-1 video sequence according to different parametric arguments.

### Encoding Parameter Adjusting

This unit provides users an interactive capability of adjusting the encoding parameters according to the visual results.

## 3 ADJUSTMENT OF ENCODING PARAMETERS

The encoding parameter adjusting unit is a very important component in the browsing system. An interactive user-interface should be provided in this unit. Moreover, unnecessary macroblock re-encoding has to be avoided. The development of the unit is based on the kernel of MPEG encoding and decoding techniques. On the other hand, the properties of MPEG bitstreams are also taken into consideration. Fig. 2 shows the flow chart of the encoding parameter adjusting unit.

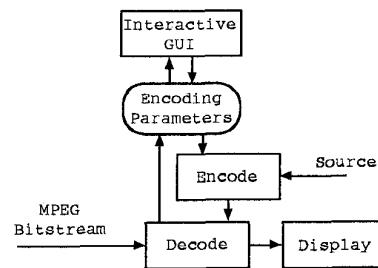


Figure 2: The flow chart of encoding parameter adjusting.

By the interactive user-interface, users can easily find the macroblocks that need to be modified and understand the encoding parameters of each macroblock. In our implementation, the encoding parameters of each macroblock in the current frame are stored in a database. When users modify the encoding parameters, the associated values in the database are modified accordingly. Meanwhile, the associated macroblocks are re-encoded, re-decoded, and shown on the associated parts of the screen. Users can thus realize the effect of the adjustment.

While the encoding parameters are modified, the associated macroblocks have to be re-encoded. The original bitstream has to be decoded before the macroblocks are re-encoded, and the corresponding header parameters are replaced by the new ones, such that the original bitstream structure can be maintained. For the macroblocks that don't need to be re-encoded, the original data streams can be placed directly on the new bitstream. The block diagram of the macroblock re-encoding is given in Fig. 3.

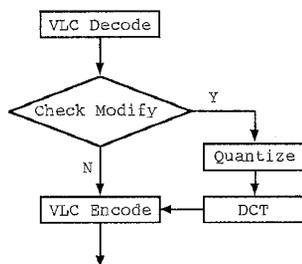


Figure 3: The block diagram of macroblock re-encoding.

For the sake of motion compensation [4], if the modified frame is an I-Picture or a P-Picture, it might have been referenced by the other frames. Those frames that reference the modified frame have to be re-encoded. The system have to find out how many macroblocks of the referencing frames should be modified. These macroblocks are re-encoded according to the new modified referenced frame.

In order to know whether a macroblock is affected by the referenced frame, three flags are set for each

macroblock: **forward reference flag**, **backward reference flag**, and **current frame flag**. The current frame flag of a macroblock is set if that macroblock is modified by users. The forward reference flag and the backward reference flag are set if the forward referenced macroblock and the backward referenced macroblock of the current macroblock are modified respectively. A macroblock has to be re-encoded when one of the three flags is set.

## 4 SYSTEM FUNCTION AND APPLICATIONS

The feature of the proposed browsing system is that the video bitstreams can be browsed at arbitrary positions. Users can check out the information and play video frames from arbitrary positions of the video bitstreams. The system functions of different functional units are detailed in this section.

### 4.1 Structure Analyzer

The structure of the neighboring areas of the specified byte address can be dumped on the screen via the structure analyzer unit. In the "Structure" menu item, users can select the structure layer, including "Pack", "Packet", "GOP" (Group Of Picture), "Picture", "Slice", "Macroblock", and "Block" layers. For example, the picture structure of the loaded bitstream near the byte address 141508 is shown on the screen, as in Fig. 4.

File Structure	Parameter	Playback	Dump	Color	Encode	Adjust
Packet						Byte Address : 141508
GOP						
Slice						
Macroblock						B Picture: Temporal Reference=5
Block						B Picture: Temporal Reference=7
						B Picture: Temporal Reference=8
						P Picture: Temporal Reference=12
						B Picture: Temporal Reference=10
						B Picture: Temporal Reference=11
						P Picture: Temporal Reference=15
						B Picture: Temporal Reference=13
						B Picture: Temporal Reference=14
						Sequence Header
						GOP Header
						I Picture: Temporal Reference=2
						B Picture: Temporal Reference=4
						B Picture: Temporal Reference=1
						P Picture: Temporal Reference=5

Figure 4: Structure analyzing.

It can be seen from the figure that the starting byte address of each structure item is shown before that item. The byte address specified by users are listed on the upper right corner of the screen, and the structure item that contains the specified byte is displayed in reversed color. Users can change the specified address by pressing the arrow keys *UP* (“↑”) and *DOWN* (“↓”). The structure layers can be shifted by pressing *LEFT* (“←”) and *RIGHT* (“→”), or selecting the new layers via the menu items. Thus, the structure analyzer unit provides users a very convenient interface. Users can analyze the layer structure on any specified address of the bit-streams.

In addition to pressing the arrow keys, users can also press a hot key to specified the new address via a dialog box.

### 4.2 Parameter Analyzer

The parameter analyzer unit displays the header parameters and other related information of a structure layer which the specified byte address belongs to. Users can select the structure layer in the “Parameter” menu item, as shown in Fig. 5. The associated information will be shown on the screen. The structure layers can also be shifted by pressing *LEFT* and *RIGHT*.

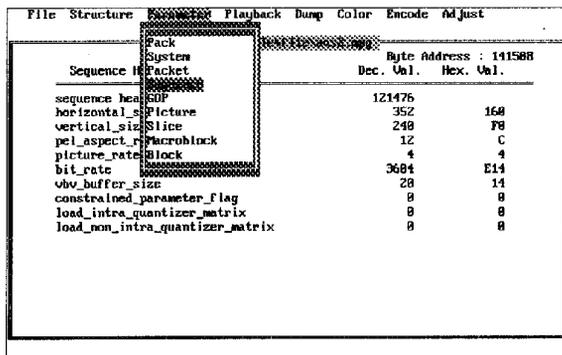


Figure 5: Parameter analyzing.

When the “Block” layer is selected, the contents of the block which contains the specified byte, are shown on the screen in addition to the encoding parameters. Users can observe the change of the block coefficients

in different processing stages as shown in Fig. 6. This will be very helpful in the development stage of MPEG encoding/decoding systems.

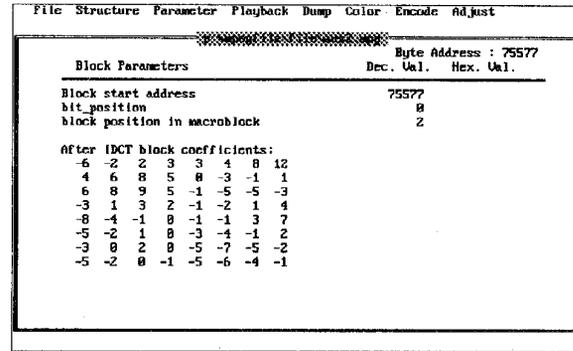


Figure 6: The coefficients of the block after IDCT.

The contents of the coefficients can be switched via hot keys. The hot keys are defined as,

- <F2>: the coefficients after Huffman decoding,
- <F3>: the coefficients after inverse quantization,
- <F4>: the coefficients after inverse discrete cosine transformation (IDCT),
- <F5>: the contents of the referenced block,
- <F6>: the final result of the block coefficients.

### 4.3 Video Player

There are two kinds of video playing functions provided by this browsing system. They are “video sequence playing” and “video frame displaying”, which can be selected from the “Playback” menu item. The “video sequence playing” function can play the whole video bitstream on the screen, and the “video frame displaying” function can display the frame that contains the specified byte on the screen.

The video can be played from a specified frame or within a portion of that video. That video portion can be marked by pressing *Ctrl+S* (start) and *Ctrl+E* (end). Users can select the video portion by the structure analyzer. The video player will play the video portion only.

While the video is played, users can press `<space>` to pause the frames, and press another hot key to display next frames one at a time.

The maximum resolution provided in this system is  $640 \times 480$  pixels, and two kinds of color modes, 256 colors and true colors, are supported. The color modes can be switched via the "Color" menu item.

#### 4.4 Video Compression

When users have to generate an MPEG-1 bitstream, they might want to set up the encoding parameters themselves. The system provides a dialog box for the input of the encoding parameters. The operation steps of video compression are listed as follows:

1. Press `Alt+E` to select the "Encode" menu item.
2. Select the "File I/O" item in the "Encode" menu item, then a pop-up dialog box will show up for inputting the basic information, as shown in Fig. 7.
3. Select the "Parameter" item in the "Encode" menu item to modify the encoding parameters, as shown in Fig. 8.
4. Select the "Motion Estimation" item in the "Encode" menu item, and select one of the motion estimation algorithms.
5. Select the "Start Encoding" item in the "Encode" menu item to generate the MPEG-1 bitstream.

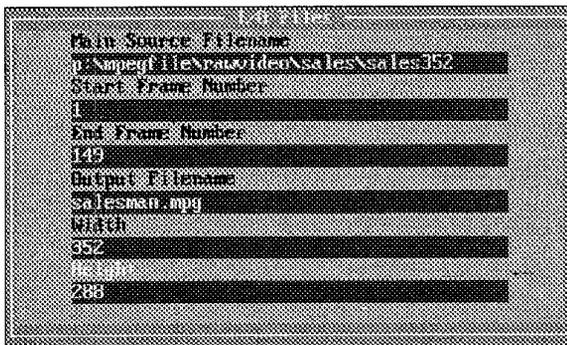


Figure 7: The "I/O Files" dialog box.

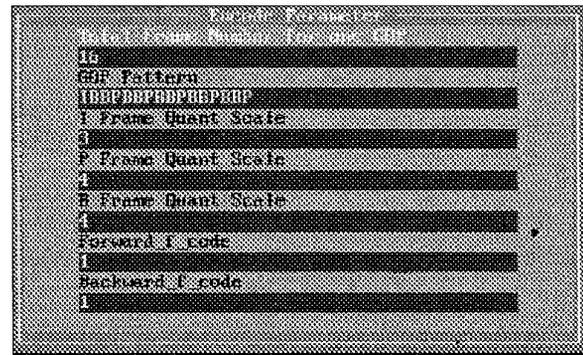


Figure 8: The "Encoding Parameter" dialog box.

#### 4.5 Parameter Adjusting

When the quality of some frame is not good enough, users can improve the quality via this parameter adjusting function. The result can be directly shown on the screen for finely adjusting the video quality. The operation steps are listed as follows:

1. Press `Alt+F` to select the "File" menu item, and select the "Open" item to load the MPEG bitstream file.
2. Specify the byte address of the frame that needed to be fixed.
3. Press `Alt+A` to select the "Adjust" menu item, and select the "Source" item to input the filename of the original video source.
4. Set the color mode to true color mode.
5. Select the "Parameter" item in the "Adjust" menu item, then the frame that needed to be fixed will be shown on the screen as in Fig. 9

The square box in the frame indicates the macroblock that is fixed currently. Users can press the arrow keys to move the square box and fix the other macroblocks. The quantization scale and the motion vectors of the current macroblock specified by the square box are shown on the upper left corner of the screen.

There are lots of functional keys defined as follows,

**Page Up/Down:** increase/decrease the quantization scale,

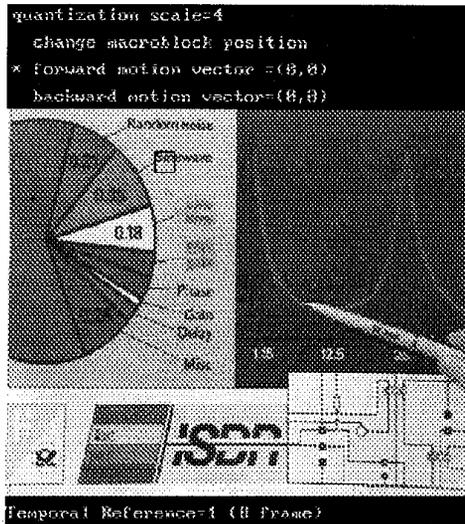


Figure 9: The video frame that is not fixed yet.

**Arrow keys:** move the square box or change the motion vectors,

**<Tab>:** switch the functions of arrow keys,

**<F1>:** change the motion compensation mode to forward prediction,

**<F2>:** change the motion compensation mode to backward prediction,

**<F3>:** change the motion compensation mode to interpolation prediction,

**<F4>:** switch the intra/non-intra encoding mode,

**<F5>:** display the original frame,

**<Esc>:** restore the modification of the encoding parameters.

For example, the specified macroblock is blurred in Fig. 10. It is known from the information shown on the upper left corner of the screen that the specified macroblock is originally encoded in intra mode, and the quantization scale is 31.

When the quantization scale is changed to 1, the visual quality is improved as shown in Fig. 11.

In addition to changing the quantization scale, the macroblock encoding type can be also modified. The

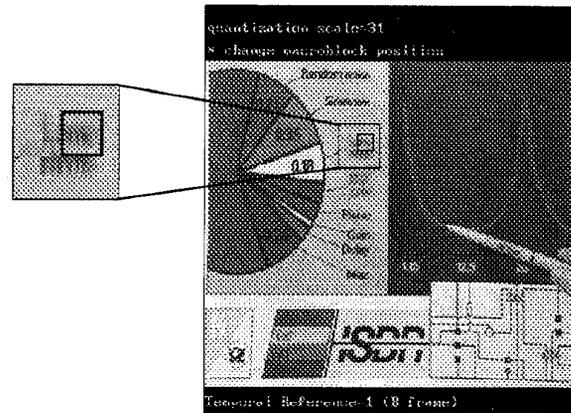


Figure 10: The video frame with blurred macroblocks.

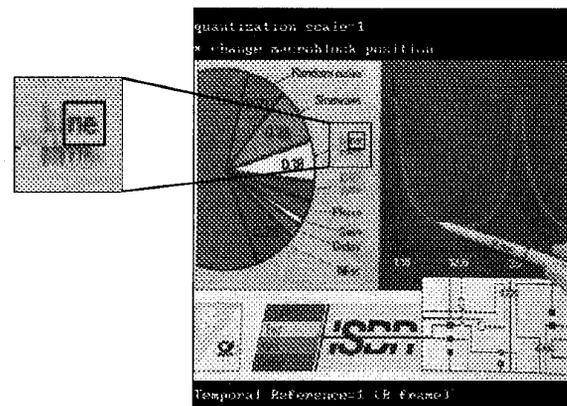


Figure 11: The result of decreasing the quantization scale.

encoding type is changed from intra mode to forward prediction mode in Fig. 12.

While comparing with Figs. 10 and 12, one can find that the quality of the macroblocks encoded in different types might be different even though their quantization scales are the same. The quality of the macroblocks encoded in non-intra mode is usually better than that of the ones encoded in intra mode. It is suggested that users can change the macroblock encoding type at first. If the result is still not acceptable, the quantization scale can then be changed. However, the drawback is that the bit rate will increase a little.

Users can press the *<Enter>* key when a frame has been fixed. Because of motion compensation, those macroblocks that reference the fixed macroblocks

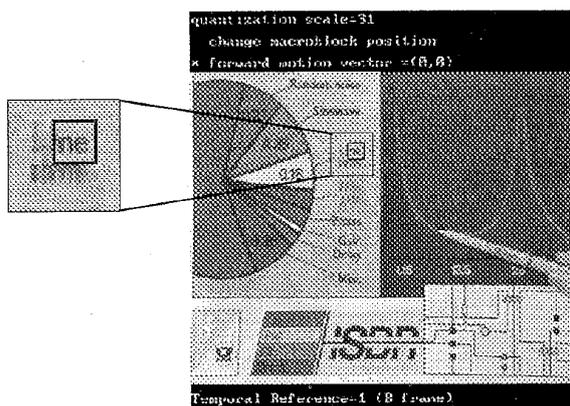


Figure 12: The result of changing the macroblock encoding type.

should be re-encoded. The system will detect all the frames that reference the fixed frame. The frames that needed to be fixed are displayed on the screen for modification. If users don't want to fixed the referencing frames, they can press  $\langle F9 \rangle$ , then the macroblocks that needed to be fixed will be re-encoded by using the original encoding parameters.

## 5 CONCLUSION

The proposed MPEG-1 browsing system provides all necessary information of MPEG-1 bitstreams. Users can check the validness of MPEG-1 bitstreams and debug their MPEG-1 codec with the aid of these information. The system also provides an interactive environment for adjusting the encoding parameters. Users can improve the video quality and explore the relationship of the visual result and the encoding parameters. It is believed that the proposed browsing system is a very useful tool for developing MPEG-1 video related applications.

## 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," *Communica-*

*tions of the ACM*, Vol. 34, No. 4, Apr. 1991, pp. 47-58.

- [3] R. Aravind, G. L. Cash, D. L. Duttweiler, H.-M. Hang, B. G. Haskell, and A. Puri, "Image and Video Coding Standards," *AT&T Technical Journal*, Jan./Feb. 1993, pp. 67-89.
- [4] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, Vol. COM-29, Dec. 1981, pp. 1799-1808.