LOGO INSERTION IN MPEG TRANSCODER

Krit Panusopone, Xuemin Chen and Fan Ling

Advanced Technology Department,
Broadband Communications Sector,
Motorola, San Diego, CA 92121

ABSTRACT

Various studies [1,2] show that a basic architecture of MPEG-2 transcoder consists of a cascaded decoder/encoder, shown in Figure 6. Such a transcoder re-uses the motion vectors (MVs) and minimizes the changes of macroblock (MB) mode. One interesting feature that can be added on this type of transcoder is to insert a translucent logo into the transcoded bit-stream.

1. INTRODUCTION

A logo often occupies a small area of a picture. It is static from frame to frame. For MPEG-2 video [3], we assume that the logo is in rectangular form.

The basic requirements for inserting a logo into a coded MPEG bit-stream are

- The quality degradation is minimized.
- The insertion is translucent.

A translucent logo in a frame is shown in Figure 7. In this figure, $p(x, y)$ denotes a pixel value of the video frame and $l(x, y)$ denotes an original pixel value in the logo. The a is the composition parameter and $0 < a < 1$.

The following issues of translucent logo insertion for MPEG video transcoding need to be considered.

- For a desired insertion area, finding an insertion position such that the logo will effect a minimum number of coded MBs.
- For the MBs which is effected by the logo insertion, how to refine the motion vectors and quantization steps such that the coding is more efficient and the impact on coding algorithm is small (which means low complexity).

In order to insert a logo to a video we need to know the desired position, and the width and the height of the hypothetical box that bounds the logo tightly (any single row or column in the box are part of the logo). We position the box up to the MB level so as to minimize the number of MBs containing by the logo. The exact position of the box on a video then equals to the desired position in pixel rounded down to the nearest mulitple of 16. In other words, the logo is aligned to the nearest top left MB on a video. In this paper, all MBs inside the hypothetical box will be called logo MBs.

We will present two methods for translucent logo insertion in the MPEG transcoder in this paper. The first method adds the logo directly at the input of the encoder loop and re-uses MVs decoded from the bit-stream as the operations of a normal transcoder. This method encounters the problem relating to motion compensation due to mismatch (addition of the effect of logo). The logo effect degrades the coding performance. The second method blends the logo content onto the video and the encoder part of the transcoder will generate the bit-stream from this video with logo. In this scenario, the encoder has to deal with the MV of the MBs surrounding the logo in an attempt to compensate the effect of logo insertion. We will describe the architecture and operation of both methods in the next section.

2. TRANSCODING OF THE VIDEO WITH A LOGO INSERTION

Transcoding of the video with a translucent logo requires a slightly different operation from the normal transcoder. The normal transcoder generates the bit-stream that provides the similar content as in the input bit-stream and hence the coding parameter of the input and output bit-stream are consistent with each other. The additional of the translucent logo may differ the input and output coding parameters, especially, in the MB surrounding the logo. Two methods are developed for the transcoder to enable the insertion of the logo to the final output.

2.1. Method 1

Configuration of the MPEG transcoder with a translucent logo insertion is shown in Figure 8. This solution re-uses MVs decoded from the input bit-stream as the operations of a normal transcoder. The residual from MC(2) will include the logo effect and will be corrected by the encoder loop. The advantage of this solution is that the mode and MV of the output bitstream will be consistent with the input bit-stream. However, the coding efficiency will be reduced because of the inaccurate MVs for the logo MBs and 'wrong' reference MBs for the MBs that have MVs pointing to the logo MBs.

The detail operation of the method 1 can be described as the following:

- Generate the translucent logo by calculating $a\{l(x, y) - p(x, y)\}$.
- Add the translucent logo to the video at the appropriate position in 'insert'.

- Buffer in 'MC(2)' contains the video with the translucent logo.
- 'DCT' will encode the video with the translucent logo.

2.2. Method 2

The second method shares the same architecture as the first one (fig. 3). In this solution, the logo is inserted translucently into a specified position before being subtracted by the reconstruction output from MC(2). The effect of insertion will be compensated by the motion compensation (MC(2)) and DCT process in the encoding part of the transcoder. Coding parameter, such as mode, MVs and DCT coefficients will have to be modified in order to improve the quality and coding efficiency.

The normal transcoder re-uses the MV from the input bit-stream. However, this MV may not be the pointer to the best match of MBs that are close to the insertion area since part of the content (logo) is always static (zero MV, (0,0)). It is reasonable to choose either the MV' decoded from the input bit-stream or zero MV for those logo MBs. Since logo MBs compose of the video (background) and the logo (foreground). To maintain a high coding efficiency, MV of logo MBs have to follow the characteristics of the motion compensated residual, i.e., zero MV for logo MB which is dominated by logo content and MV' from the input bit-stream for logo MB which is dominated by video content. We propose the thresholding concept to determine the MVs fo the logo MBs as follows:

MV(x,y)=(0,0) when a is greater than or equal to some adjustable threshold, e.g., $a = 0.5$.

MV(x,y)=MV'(x,y), otherwise. That is using MVs decoded from input bit-stream.

There is a need to check the validity of the coding mode when the MV is changed. It will be inefficient to use a complex mode when the logo MB is dominated by logo. To keep the complexity of the system low, a logo MB will be coded using forward prediction mode when MV(x,y) is re-set to zero MV. If this MB is originally intra-coded, its mode will be changed to a forward-prediction only MB with zero MV.

Another situation is that the MVs decoded from the bit-stream may not be the correct MVs after logo insertion. The MVs decoded from the bitstream that points into the logo MB does not expect to pick up the logo content which is added in MC(2). This problem can be viewed as a reversed way of the previous case. The encoder can code MBs which has MVs pointed to logo MB with zero MV or MVs decoded from the bit-stream based on the following criteria:

MV(x,y)=(0,0), when a is less than or equal to some adjustable threshold, e.g., $a = 0.5$.

MV(x,y)=MV'(x,y), otherwise. That is, using MVs decoded from input bit-stream.

3. FURTHER IMPROVEMENTS

Constant quality of the logo may be the desired property in some applications. While the quantization parameter (QP) of the normal MB is varied under the rate control mechanism, the logo MB can be forced to use a fixed QP. This QP constant can be fixed at a high value so as to decrease the overall bit-rate or vice versa.

4. EXPERIMENTAL RESULTS

Both methods 1 and 2 are implemented in the transcoder software. The transcoder is implemented according to the architecture in Fig. 1. This transcoder takes any MPEG-2 compliant bitstream as its input. The user is also required to supply the logo input, its alpha factor, a, and the logo coordinate (width, height and position). Rate control mechanism has been disabled in this experiment so that the effect of logo insertion is not diluted. Four MPEG-2 test bit-streams (main profile at main level) i.e., Cheer (5 Mbps), Bus (5 Mbps), Football (5 Mbps) and Vides-home (6 Mbps), are used in this experiment. The simulation results are shown in Table 1 and 2. These results show the number of bytes of the output (with logo inserted) bit-stream. Figures 4 and 5 show the total bits changes for both method at different a values. Figures 1 shows the decoded frame (frame 790) from 'vides-home' bit-stream (This bit-stream is used as one of the input to the transcoder). Figures 2 and 3 show the same decoded frame when transcoded with method 1 and 2 ($a = 0.5$). From experimental results, it can be seen that overall performance of method 2 is better than that of method 1. Also notice that method 1 requires a larger bits when adds a more dominant logo (higher a) while this effect is relatively stable for method 2.

Table 1. Total number of bytes generated by Method 1 at different alpha values.

Sequence	Cheer	Bus	Football	Vides-home
0	624745	3184243	6187732	33369924
0.1	625695	3278338	6375026	35704147
0.5	634251	3687269	7039509	40566292
0.9	644212	3980031	7442067	43156554

Table 1. Total number of bytes generated by Method 2 at different alpha values.

Sequence	Cheer	Bus	Football	Vides-home
0	624745	3184243	6187732	33369924
0.1	635051	3418719	6377657	34176823
0.5	622306	3383432	6481521	34162602
0.9	608269	3338601	6511191	33927290

5. CONCLUSIONS

This paper describes two methods for logo insertion in the MPEG transcoder. Both methods have a slightly different architecture from the normal transcoder. The logo will be inserted translucently on the video before being subtracted by the reconstruction output. To maintain a high coding efficiency of the system, two compensation techniques for coding the effect of logo insertion are discussed. We experience from the simulation that the logo insertion may increase or decrease the overall bit-rate depending on the relative complexity of the logo and the video. This paper also presents the approach to position the logo in such a way that the affected MB will be minimum.

Figure 1: Frame 790 of the decoded MPEG-2 bit-stream 'Vides-home'

Figure 2: Frame 790 with logo inserted ($a = 0.5$) using the proposed method 1

Figure 3: Frame 790 with logo inserted ($a = 0.5$) using the proposed method 2

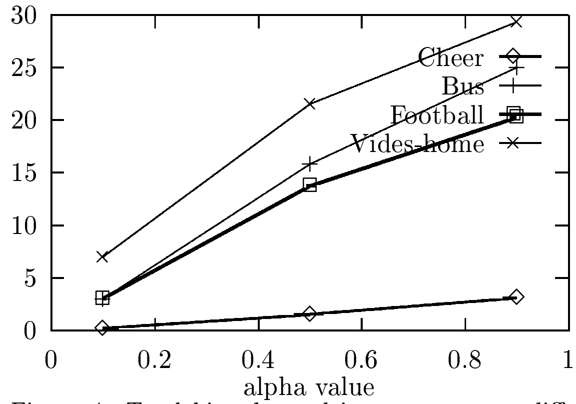

Figure 4: Total bits changed in percentage at different a values for method 1

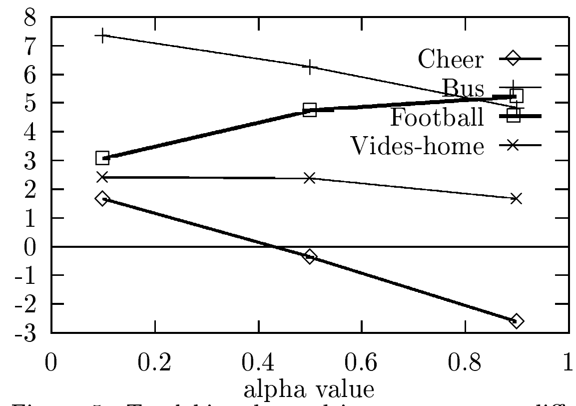

Figure 5: Total bits changed in percentage at different a values for method 2

6. REFERENCES

[1] Limin Wang, Ajay Luthra and Bob Eifrig, Adaptive rate control for MPEG transcoder, Proc. Intl. Conf. on Image Processing, Kobe, Japan, pp. 266–270, Oct. 1999.

[2] G. Keesman, R. Hellinghuizen, F. Hoeksema and G. Heideman, Transcoding of MPEG bitstreams Signal Processing : Image Communications, vol. 8, pp. 481–500, 1996.

[3] ISO/IEC 13818-2, Generic Coding of Moving Pictures and Associated Audio Information: Video International Organization for Standardisation, Nov. 1994, Draft International Standard (MPEG-2 Video)

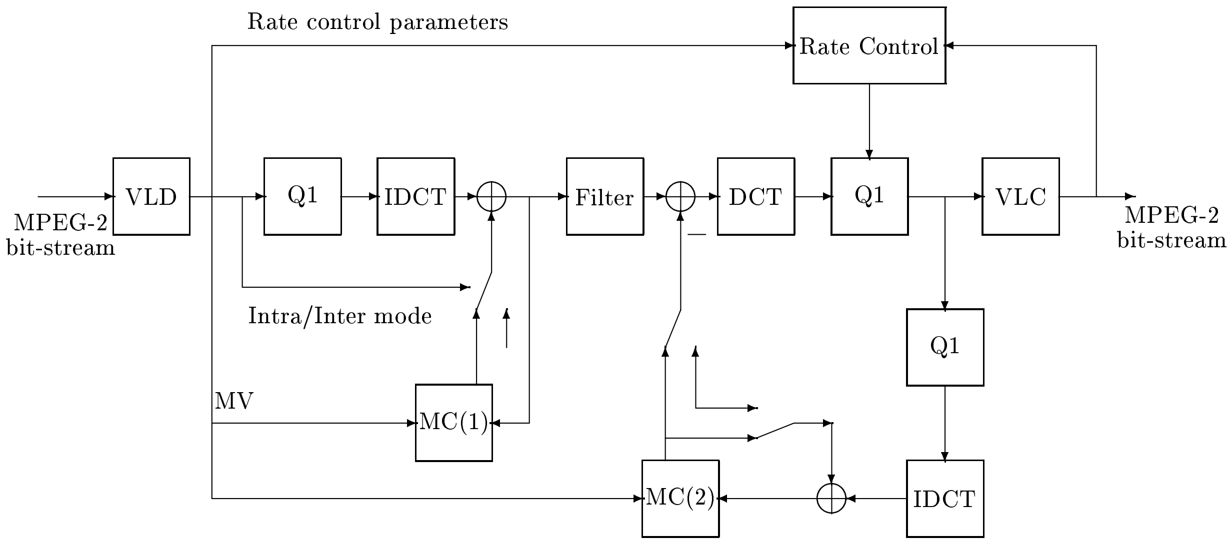

Figure 6: Block diagram of binary shape decoding

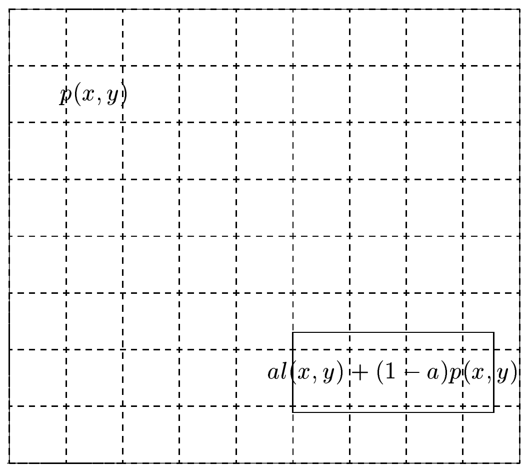

Figure 7: Direct mode of interlaced video

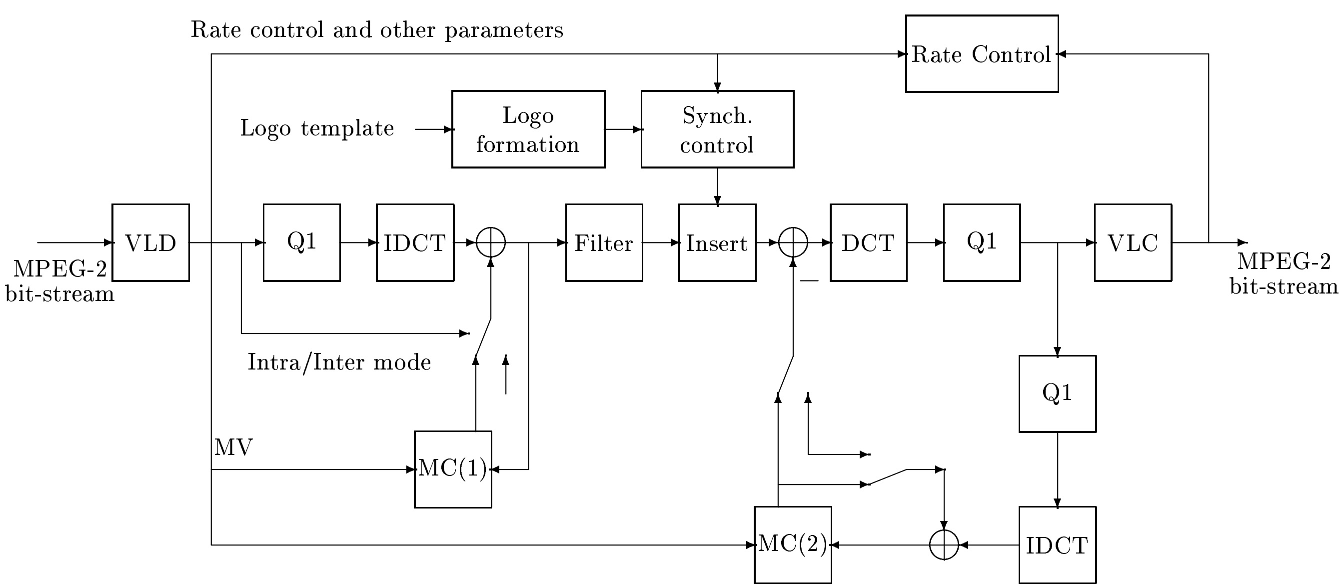

Figure 8: Direct mode of interlaced video