A FAST FRAME-ACCURATE H.264/MPEG-4 AVC EDITING METHOD

Akio Yoneyama, Yasuhiro Takishima, and Yasuyuki Nakajima
Multimedia Communications Laboratory
KDDI R&D Laboratories Inc. Saitama 356-8502, JAPAN
E-mail:{yoneyama,takisima,nakajima}@kddilabs.jp

ABSTRACT

H.264/MPEG-4 AVC has become widely used in various applications and services because of its high coding efficiency. With the spread of H.264/MPEG-4 AVC contents, it has been strongly required to edit the H.264/MPEG-4 AVC coded contents with less computational complexity since the compression of H.264/MPEG-4 AVC requires far more processing power than other existing formats such as MPEG-4, MPEG-2.

In this paper, we propose a fast frame-accurate editing method for H.264/MPEG-4 AVC contents. In the proposed method, we employed "Adaptive bit allocation and GOP-length modification" and "Re-quantization scheme with drift noise management" The proposed editing method can realize frame-accurate cut and splice editing regardless of picture coding types I, P, and B.

In the experiment, we could also confirm the effectiveness of the proposed method in terms of processing power and the picture quality of the edited contents when compared with conventional editing.

1. INTRODUCTION

Efficient editing of H.264/MPEG-4 AVC ("H.264/AVC" hereafter) contents has been greatly in demand as H.264/AVC contents increases. This requirement is much stronger than those to MPEG-2 or MPEG-4, for H.264/AVC editing involves far larger computational complexity.

As for the hybrid-coding format which employs motion compensation and orthogonal transform, there is a fast editing method provided that the editing is performed in GOP-based. Since there is no dependency between consecutive GOPs when closed GOP, we can easily cut and splice contents without decoding the originally coded stream. (In detail, the buffer size and buffer fullness condition should be investigated.) However, considering that a GOP typically consists of more than 10 frames, temporal editing accuracy is limited.

As opposed to GOP-based editing, frame-accurate editing cannot be achieved by extending GOP-based editing scheme because of interframe coding. The most straightforward approach for frame-accurate editing is "re-encoding" method with tandem connection of frame-based decoder and re-encoder. Though this approach has no limitation on application, it may cause degradation in terms of picture quality and may result in time-consuming operation.

There have been various approaches which do not require full-decoding and re-encoding. To minimize processing cost for editing, they employ "picture-type conversion at editing point", and "compressed-domain re-quantization for subsequent frames". The basic idea of re-quantization scheme is to maintain the most part of the originally coded stream. Though this approach has limitation on applicable conditions, fast editing can be achieved compared to straightforward approach. On the other hand, picture quality may be degraded because of the lack of drift noise handling.

To solve the above mentioned problems, we propose a fast frame-accurate editing method. This scheme has following advantages: i) high speed editing, and ii) less degradation in picture quality. In section 2, we briefly review previous works related to H.264/AVC frame-accurate editing. In section 3, we propose a fast editing scheme for H.264/AVC. Then, the proposed method is compared with the conventional approach which employs straightforward method in section 4.

2. REVIEW OF PREVIOUS WORKS

Tandem connection of a decoder and a re-encoder has been widely used as a frame-accurate editing processing. Since every frame is reconstructed, the editing operation results in a simple decoding and encoding operations. There have been proposals to improve both editing speed and quality of edited pictures. Basic idea of these methods is to utilize the coding information stored in the original compressed video such as Picture-type, MB-mode, and motion vector at the re-encoding stage. Especially, the re-use of motion vector information greatly contributes to increase the overall editing speed.

Another editing approach is compressed domain editing technique. For example, [1] proposes a method to concatenate MPEG segments while maintaining VBV

0-7803-9332-5/05/$20.00 ©2005 IEEE

condition. Picture-type conversion method in compressed domain for MPEG content trimming at arbitrary frame is explained in [2]. These compressed domain editing aims the operation without full-decoding the original coded stream. This approach has possibility to achieve fast editing since full-decoding and re-encoding process is not included. However, the compressed domain re-quantization method may introduce picture quality degradation because a modification in a referred frame introduces error propagation to subsequent frames which refer the modified frame without error feedback loop. Moreover, it is required to minimize the number of frames to be modified at the editing process to increase editing speed.

3. PROPOSED FAST EDITING METHOD

Figure 1 shows an example of cut only editing operation using two H.264/AVC streams *A* and *B*. Stream *A* and *B* are trimmed at the points *CPa1*and *CPb1*, then these trimmed streams are spliced and edited stream *C* is obtained. In the figure, "I" and "P" represent IDR (Instantaneous Decoding Refresh) picture and predictive picture. Note that there is no temporal dependency over the IDR-picture. We denote three groups of pictures GOP-A: the last GOP just before cut point *CPa1* in Stream *A*, GOP-B1: the first GOP just after cut point *CPb1*, and GOP-B2: the next GOP to GOP-B1, respectively as shown in Fig.1.

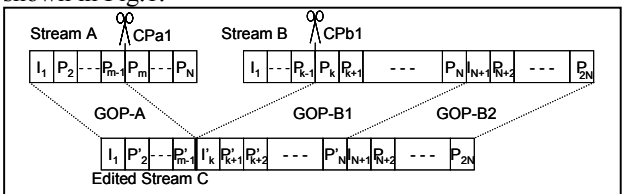

Fig.1 Frame-accurate editing (Case-1)

The proposed method contains mainly seven functions listed below:

i) Frame-level coding mode modification
ii) Bit budget determination for GOP-A
iii) Bit budget determination for GOP-B1
iv) GOP structure determination
v) MB-level Re-quantization operation
vi) Re-quantization with drift noise reduction
vii) Control for multiple reference frame

In the following subsections, the detail of each function is explained.

3.1. Frame-level coding mode modification

At the frame-accurate editing operation shown in Fig. 1, Picture (P_k) needs to be changed to Intra-coded picture I'_k in the edited stream C. Here, we re-encode the P_k picture as I-picture (I'_k) with same quantization parameters as those in P_k. to minimize introduction of propagation error in the subsequent frames in the same GOP.

Generally, consumed number of bits for I'_k is more than that of the original picture P_k due to intra coding. The increase in bit consumption at I'_k introduces two problems. i) It may cause buffer underflow due to the insertion of I'_k. ii) It may cause buffer underflow at the subsequent frames to I'_k. Consequently, we need to save bit budget for the increased bit consumption of I'_k in both "precedent" and "subsequent" frames. Figure 2 shows an example of bit budget modification in the subsequent frames when the GOP has seven frames and the cut point is between P_3 and P_4. In this sample, the increased bit budget in I'_4 is counterbalanced by bit budget reduction in the subsequent three frames P'_5, P'_6, and P'_7. Moreover, we need to modify the bit budget at the precedent frames which is described in the following subsection.

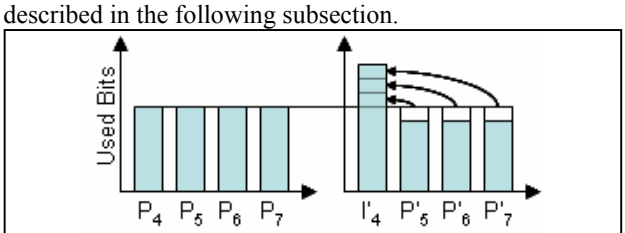

Fig.2 Bit Budget determination

3.2. Bit Budget determination for GOP-A

Since we already know the data size of I'_k which is the first frame from Stream *B*, it is possible to check the buffer underflow/overflow at the presence of I'_k in the edited Stream *C*. To avoid buffer underflow at the presence of I'_k, it is required to keep the buffer occupancy to some level at the end of GOP-A. We determined the level: *L* as follows:

$$L = B(I'_k) + MinBuffLevel \quad (1)$$

, where $B(I'_k)$ and *MinBuffLevel* represent used bit of frame I'_k and minimum buffer margin, respectively. Assume that the buffer level at the end of GOP-A is denoted as $Buff(P'_{m-1})$. If the $Buff(P'_{m-1})$ is lower than *L*, we need to re-encode pictures belonging to GOP-A with smaller bit consumption to keep buffer occupancy at the end of GOP-A more than *L* as shown in Fig.3.

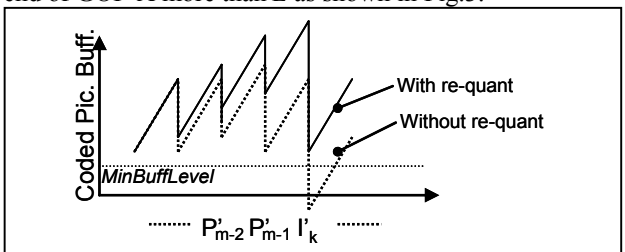

Fig.3 CPB Buff Level investigation

3.3. Bit Budget determination for GOP-B1

The increased bit consumption in I'_k needs to be counterbalanced by other frames which belong to the

same GOP. We determined the new bit budget for subsequent pictures in the GOP as shown in Eqs.(2) to (4). to modify all frames in the GOP equally in terms of picture quality.

$$B_{tar}(P'_n) = B_{oh}(P_n) + B_{coef}(P_n) * BRR \qquad (2)$$

, where

$$BRR = -\left(\sum_{j=k+1}^{N} B_{coef}(P_j) - B_{diff} \right) \Big/ \sum_{j=k+1}^{N} B_{coef}(P_j) \qquad (3)$$

, where

$$B_{diff} = B_{bits}(I'_k) - B_{bits}(P_k) \qquad (4)$$

In the above three equations, each parameter represents as follows:

k: Frame number in original GOP which is the first frame in the edited GOP (GOP-B1).

N: Number of frames in original GOP-B1 and GOP-B2.

BRR: "Bit Reduction Ratio" of bit consumption for transformed coefficients of subsequent pictures in the same GOP.

$B_{tar}(P'_n)$: Target bit for edited picture P'_n.

$B_{oh}(P_n)$: Used bit for overhead of picture P_n.

B_{diff}: Increased bit consumption at the conversion from P_k to I'_k.

In order to reduce the bit budget of the subsequent pictures P'_k, P'_{k+1}, and P'_{k+2}, we employ re-quantization method described in subsection 3.5.

3.4. GOP Structure determination

Generally, the increased bit budget in changing from P-pic to I-pic is counterbalanced by all other frames within the same GOP, However, when the cut point is near to the next I-picture I_{N+1}, it should introduce severe degradation in terms of picture quality in the subsequent frames in the same GOP. This is because it forces larger bit budget reduction in the subsequent frames than that in the Case-1. In such case, we connect the short GOP with the next GOP as shown in Fig. 4. Due to this GOP reconstructuring, picture quality degradation can be avoided since most of the bit budget change B_{diff} is compensated by the subsequent picture type change from I_{N+1} to P'_{N+1}.

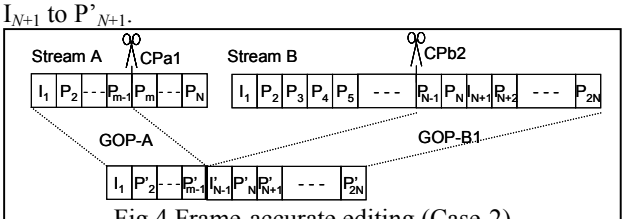

Fig.4 Frame-accurate editing (Case-2)

Here, we need to determine whether GOP reconstructuring is needed or not. For this purpose, we evaluated *BRR* in Eq.(2) since it can directly indicate the bit reduction. If the *BRR* is less than a threshold, GOP reconstructuring is performed.

3.5. MB-level re-quantization

By the above sections from 3.1 to 3.4, we described methods for a conversion from P_k to I'_k and target bit budget determination for P_{k+1} to P_N. Then we need to modify pictures from P_{k+1} to P_N according to the required bit budget modification. In this step, MB level re-quantization method is employed since much faster processing than the baseband transcoding method while maintaining the picture quality can be expected when noise reduction mechanism is included as shown in the following.

3.6. Re-quantization with drift noise management

Since all the pictures which belong to GOP-B1 need to be modified according to the editing operation, we need to take into account of drift noise problem which is caused by re-quantization approach.

In the proposed method, the drift noise is appropriately managed using in-loop drift error feedback scheme which is shown in Fig. 5. The transformed coefficient difference between the original and re-quantized is investigated and the difference is stored to frame memory : FM. The difference is also used for motion compensation. The motion compensated difference data is transformed and added to original inverse-quantized coefficients. Finally, the re-quantized coefficients are encoded using VLC.

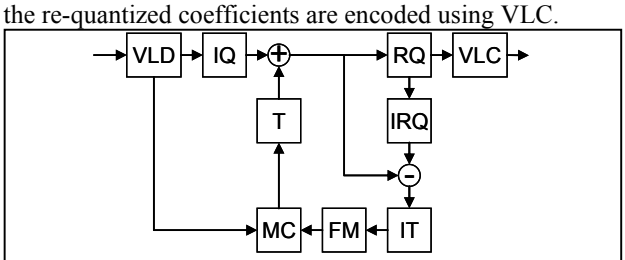

Fig.5 MB-level requantization block diagram

3.7. Control of multiple reference frame

Although reference frame is fixed in other coding standards such as MPEG-2 and MPEG-4, we can completely find reference area for all MBs in P'_{k+1} to P'_N. On the contrary in the case of H.264/AVC, reference area is not limited to the temporally adjacent frame since H.264/AVC introduces multiple reference frame coding mode. In such cases, we re-encode the MBs with motion search operation from the temporally adjacent frame.

The editing process of the proposed method is summarized as follows. Firstly, if the cut point is not just before I-pic, the picture just after the cut point is

converted from P-pic to I-pic with the same quantize parameters as described in subsection 3.3. Secondly, the bit budget increase at the conversion is investigated. The increase is compensated basically by subsequent frames in the same GOP. But if the picture quality degradation is inevitable by the compensation such as the case that the cut point is near the end of GOP, we concatenate the GOP with the next GOP. Finally, we re-quantize the modified GOP by the proposed method as shown in Fig. 6.

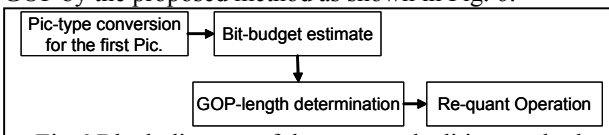

Fig.6 Block diagram of the proposed editing method.

4. SIMULATION RESULT

We have performed frame accurate editing simulation under the following conditions:

- Picture Format : H.264/AVC (Baseline)
- Picture Size : 352 x 240
- Framerate : 30fps
- Bitrate : 512kbps (CBR)
- IDR Interval : every 30-frame
- B-Picture : Not used
- Test sequence : Video captured from TV program

We prepared two image sources *A* and *B* whose length are both 60 seconds. Then, we trimmed the two streams near the beginning frame where is not GOP boundary. Finally, we spliced the contents by the following methods:

Type-0 (Pixel domain editing as benchmark) :
1) Stream *A* and *B* are edited in pixel domain
2) The pixel domain edited stream is encoded using above mentioned condition.

Type-1 (Re-encoding) :
1) Stream *A* and *B* are encoded using above mentioned condition.
2) Stream *A* and *B* are decoded and edited and finally re-encoded using the same encoding condition.

Type-2 (Proposed method) :
1) Stream *A* and *B* are encoded using above mentioned condition.
2) Stream *A* and *B* are edited using our proposed method.

Table 1 shows PSNR result comparison among Type-0, Type-1, and Type-2. As can be seen from the table, the PSNR degradation introduced by re-encoding approach is about 1.5dB. On the contrary, the PSNR degradation introduced by the propose method is suppressed less than 0.12dB. It is clear that the propose method maintains higher PSNR compared to that in fully re-encoding method.

Table 1 PSNR comparison among editing approach

	Type-0 (Benchmark)	Type-1 (Re-encode)	Type-2 (Proposed)
PSNR	37.358	35.904	37.244
ΔPSNR to Type- 0	-	1.454	0.114

Moreover, we tested the effectiveness of the in-loop drift error feedback in the proposed method using various sequence. The PSNR gain by employing the in-loop feedback was from 0.5 to 1.3dB.

Table 2 shows edit processing time comparison between Type-1 and Type-2 using Windows PC (Pentium 4 3.06GHz). As can be seen from the table, editing speed of the proposed method is about four times faster than that of straightforward method.

Table 2 Processing time comparison

	Type-1 (Re-encode)	Type-2 (Proposed)
Processing time (Sec.)	171	43

5. CONCLUSION

In this paper, we propose a fast frame-accurate H.264/AVC editing method. The main advantage of our work is that the proposed method can greatly reduce the processing time and also reduce the picture quality degradation introduced by the editing operation. The picture quality is maintained by "Adaptive GOP-length modification" which adaptively concatenates a GOP to the next GOP, and "Re-quantization scheme with drift noise management" which controls error propagation caused by the editing operation. The proposed method has no limitation on cut point and splice point.

6. ACKNOWLEDGEMENTS

The authors would like to deeply thank to Dr. T. Asami and Dr. S. Matsumoto for their continuous support, and also Mr.T.Aoyagi for the computer simulations.

REFERENCES

[1] R. Egawa, A. A. Alatan, and A. N. Akansu,"Compressed domain MPEG-2 video editing with VBV requirement", *IEEE Proc. International Conference on Image Processing,ICIP*, Vol.1, pp. 1016-1019, 2000.

[2] K. Wang, and J. W. Woods,"Compressed domain MPEG-2 video editing", *IEEE Proc. International Conference on Multimedia and Expo, ICME*, Vol.1 pp.225-228, 2000.

[3] J Meng; and S.-F. Chang, "Buffer control techniques for compressed-domain video editing", *IEEE Proc. International Symposium on Circuits and Systems, ISCAS*, Vol.2,pp.600-603,1996.