Block-based Error Compensation Method for Fast Thumbnail Generation in H.264/AVC Bitstreams

Kyung-Jun Lee and Je-Chang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, Republic of Korea
e-mail: kjlee8812@gmail.com, jjeong@hanyang.ac.kr

***Abstract*—The thumbnail pictures can be generated from the frequency data, which is called H.264 advanced video coding (H.264/AVC) bitstreams, directly without inverse transform process. However, it makes some error caused by rounding of the floating point operation and it will be propagated to neighbor blocks. In this paper, we propose an error compensation method for fast thumbnail generation considering error propagation. It determines the block-based compensation value using the error distribution of intra prediction mode and gives the weighting factor to cover the error propagation.**

Keywords-thumbnail; H.264/AVC; error compensation.

I. INTRODUCTION

H.264/AVC is widely used for digital video processing including both of high and low bit rate applications, such as high definition television (HDTV), internet TV and digital multimedia broadcasting (DMB) services. Moreover, as the development of the personal devices, people can take video contents easily, so both of online and offline storage overflow with a lot of multimedia. In these reasons, some of generating reduced-size images which called thumbnail are needed for file searching and restoring operation. Because thumbnail can give intuitive information of video data, so it can be used for video searching, browsing, and displaying.

The thumbnail extraction from the frequency domain directly is normally used for fast generation. Because DC coefficient in the frequency domain block is considered as the representative value of the block. Therefore, the collection of the DC coefficients of entire images become a thumbnail. Yeo and Liu [1] proposed a method to make DC image consists of DC coefficients of a MPEG-1 frame, and it called DC sequence. Likewise, most thumbnail extracting method from the video streams of the MPEG-1/2 can make various sizes thumbnail images with reduced complexity [2]-[3].

H.264/AVC supports the intra prediction process which predicts and reconstructs the blocks of the image in spatial domain [4]. Chen *et al.* described an intra prediction process as matrix multiplication and proposed a frequency domain prediction method [5]. Kim *et al.* and Yu *et al.* proposed a fast thumbnail extraction method which calculate DC coefficient from the frequency domain directly using Chen's method [6]-[7]. Kim *et al.* proposed another fast thumbnail generation method by substituting multiplications to shifting operations [8]. Yoon *et al.* proposed an error compensation method for thumbnail images [9].

This paper proposes an enhanced error compensation method for thumbnail generation based on [9]. We focused on error propagation which is occurred because of rounding of the floating point operation. We collect the error distribution data from the thumbnail images and set the mean and deviation value. And considering the location of the blocks, we set the weighting factors to determine different compensation values.

This paper is organized as follows. In Section 2, related works are introduced. In Section 3, we describe how to determine the compensation value of the thumbnail image extracting from the bitstreams directly. Section 4 shows the experimental results and Section 5 concludes the paper.

II. RELATED WORKS

A. Fast Thumbnail Extraction

Chen described an intra prediction as a matrix multiplication for 9 modes [5]. Figure 1 shows a 4x4 block. It uses the pixel values in four neighboring blocks for intra prediction. The prediction block of current block $\mathbf{y}_p^m$ can be calculated with matrix form.

$$\mathbf{y}_p^m = \left(\sum_{n=1}^{3}\sum_{i=1}^{4}\mathbf{s}_i\mathbf{x}_n\mathbf{c}_{n,i}^m\right) + \sum_{i=1}^{4}\mathbf{c}_{4,i}^m\mathbf{x}_4\mathbf{s}_i^T \qquad (1)$$

where m, $\mathbf{s}_i$, $\mathbf{x}_n$, $\mathbf{c}_{n,i}^m$ refers to the intra prediction mode, the shift matrix, the neighboring block of current block and the constant matrix, respectively. And prediction for next block, only the boundary pixels of the current block are needed. Therefore, we filtered the current prediction block in both of vertical and horizontal direction in advance. For this filtering process following **v**, **h** matrix will be used [6].

$$\mathbf{v} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{h} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad (2)$$

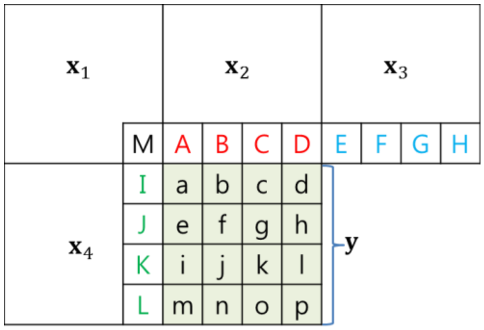

Figure 1. Current block y and the neighboring blocks $\mathbf{x_1}$ to $\mathbf{x_4}$ for intra prediction.

$$V(\mathbf{a}) = \mathbf{va} = \mathbf{a}_v \tag{3}$$

$$H(\mathbf{a}) = \mathbf{ah} = \mathbf{a}_h \tag{4}$$

Equation (3) and (4) are the vertical and horizontal filtering operator, respectively. By using this operator, we can get filtered prediction blocks.

$$\mathbf{y}_{p,v}^{m} = \left(\sum_{n=1}^{3}\sum_{i=1}^{4} V(\mathbf{s}_i\mathbf{x}_{n,v}\mathbf{c}_{n,i}^{m})\right) + \sum_{i=1}^{4} V(\mathbf{c}_{4,i}^{m}\mathbf{x}_{4,\mathrm{h}}\mathbf{s}_i^{T}) \tag{5}$$

$$\mathbf{y}_{p,h}^{m} = \left(\sum_{n=1}^{3}\sum_{i=1}^{4} H(\mathbf{s}_i\mathbf{x}_{n,v}\mathbf{c}_{n,i}^{m})\right) + \sum_{i=1}^{4} H(\mathbf{c}_{4,i}^{m}\mathbf{x}_{4,\mathrm{h}}\mathbf{s}_i^{T}) \tag{6}$$

After calculation, (5) and (6) can be simplified and it is defined as follows:

$$\mathbf{y}_{p,v}^{m} = \mathbf{x}_{1,v}\mathbf{c}_{1,4}^{m} + \mathbf{x}_{2,v}\mathbf{c}_{2,4}^{m} + \mathbf{x}_{3,v}\mathbf{c}_{3,4}^{m} + (\mathbf{p}_{4,v}^{m}\mathbf{x}_{1,h})^{T} \quad \text{where } \mathbf{p}_{4,v}^{m} = \textstyle\sum_{i=1}^{4}(\mathbf{s}_i V(\mathbf{c}_{4,i}^{m})) \tag{7}$$

$$\mathbf{y}_{p,h}^{m} = \mathbf{x}_{1,v}\mathbf{q}_{1,h}^{m} + \mathbf{x}_{2,v}\mathbf{q}_{2,h}^{m} + \mathbf{x}_{3,v}\mathbf{q}_{3,h}^{m} + \mathbf{c}_{4,4}^{m}\mathbf{x}_{4,h} \quad \text{where } \mathbf{q}_{4,h}^{m} = \textstyle\sum_{i=1}^{4}(H(\mathbf{c}_{n,i}^{m})\mathbf{s}_i^{T}) \tag{8}$$

To make DC coefficient, *UNI* operation is needed. It fills the whole components in the matrix with mean value of the block. For this process, following **u** matrix will be used.

$$\mathbf{u} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{9}$$

$$U(\mathbf{a}) = \frac{1}{N \times N}\mathbf{uau} \tag{10}$$

$$\mathbf{y}_{p,uni}^{m} = \sum_{n=1}^{3} \mathbf{x}_{n,v}\mathbf{c}'^{m}_{n,i} + (\mathbf{c}'^{m}_{4})^{T}\mathbf{x}_{4,h} \quad \text{where } \mathbf{c}'^{m}_{n} = \tfrac{1}{16}\textstyle\sum_{i=1}^{4}\mathbf{c}_{n,i}^{m}\mathbf{u} \tag{11}$$

For calculate in the frequency domain, (7), (8), and (11) are transformed by *HT* which is the modified DCT operator in H.264/AVC.

$$\mathbf{Y}_{p,v}^{m} = \mathbf{X}_{1,v}\mathbf{C}_{1,4}^{m} + \mathbf{X}_{2,v}\mathbf{C}_{2,4}^{m} + \mathbf{X}_{3,v}\mathbf{C}_{3,4}^{m} + (\mathbf{P}_{4,v}^{m}\mathbf{X}_{1,h})^{T} \tag{12}$$

$$\mathbf{Y}_{p,v}^{m} = \left(\sum_{n=1}^{3}\mathbf{X}_{n,v}\mathbf{Q}_{n,h}^{m}\right)^{T} + \mathbf{C}_{4,4}^{m}\mathbf{X}_{4,h} \tag{13}$$

$$\mathbf{Y}_{p,uni}^{m} = \sum_{n=1}^{3}\mathbf{X}_{n,v}\mathbf{C}'^{m}_{n} + (\mathbf{C}'^{m}_{4})^{T}\mathbf{X}_{4,h} \tag{14}$$

B. *Error Compensation for Thumbnail Image*

In H.264/AVC, some rounding errors are occurred because of transform, and quantization process. Yoon focus on statistical pattern of truncated errors and set a random variable *r* to compensate them [9]. The compensation value *s* is determined which makes minimum variance of *r*. The matrix form of *s* can be written as follows:

$$\mathbf{s}^{m} = \begin{bmatrix} E[r_{0,0}] & E[r_{0,1}] & E[r_{0,2}] & E[r_{0,3}] \\ E[r_{1,0}] & E[r_{1,1}] & E[r_{1,2}] & E[r_{1,3}] \\ E[r_{2,0}] & E[r_{2,1}] & E[r_{2,2}] & E[r_{2,3}] \\ E[r_{3,0}] & E[r_{3,1}] & E[r_{3,2}] & E[r_{3,3}] \end{bmatrix} \tag{15}$$

In the same way, the representative value which is the mean of distributed error can be written in the matrix form as follows:

$$\mathbf{D}^{m} = \begin{bmatrix} E[D_{0,0}] & E[D_{0,1}] & E[D_{0,2}] & E[D_{0,3}] \\ E[D_{1,0}] & E[D_{1,1}] & E[D_{1,2}] & E[D_{1,3}] \\ E[D_{2,0}] & E[D_{2,1}] & E[D_{2,2}] & E[D_{2,3}] \\ E[D_{3,0}] & E[D_{3,1}] & E[D_{3,2}] & E[D_{3,3}] \end{bmatrix} \tag{16}$$

This predicted error value also will be filtered in vertical and horizontal direction and then it compensates the thumbnail image.

III. PROPOSED METHOD

We propose an enhanced error compensation method using error distribution in each intra prediction modes and set the weighting factors by considering the location of the blocks. When the current block is predicted, it will be filtered for next blocks' prediction. Therefore, the errors also influence next blocks and it will cumulate. For this reason, we must set different compensation value. Each intra prediction modes have a characteristic error distribution, so we use this information to compensate the difference. We calculate the average and the deviation as a representative value of the block in each intra prediction mode.

$$V_{ij} = \mu^{m} + \omega_{k}^{m}\sigma^{m}, if\ i \in b_k \tag{17}$$

 ISBN: 978-1-61208-444-2

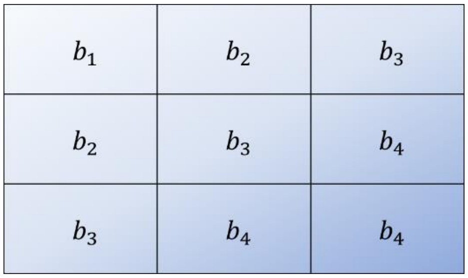

Figure 2. Partitioned image for different weighting factors.

Equation (17) determine the compensation values. The $\boldsymbol{\mu}^{\boldsymbol{m}}$, $\boldsymbol{\sigma}^{\boldsymbol{m}}$ are the average value and the deviation of mode *m*, respectively and we can get this factor experimentally. In order to consider the error distribution, we divide an image into blocks as shown in Figure 2. Finally we calculate the compensation values V_{ij} by using weighting factors ω_k^m.

IV. Experimental Results

The proposed method was evaluated by the following conditions. We randomly choose the test images and the seven unofficial JPEG images (*Album1*, *Album2*, *Hirmer*, *Soccer*, *Beatles*, *TVshow*, and *Flowershop*) were used. The size of the images is shown in Table I. We generate three thumbnail images. Frequency domain (FD) makes the thumbnail by extracting DC coefficients and fast generation method from frequency domain (FFD) makes it by using [6]. Block-based FD (BFD) is the result of proposed method. The weighting factors related to each prediction mode and location of block are set experimentally. We were doing the experiment on the MATLAB program for simple comparison.

TABLE I. THE SIZE OF THE TEST IMAGES.

Image	Size
Album1	600×600
Album2	640×640
Hirmer	720×540
Soccer	900×656
Beatles	1024×768
TVshow	1280×720
Flowershop	4272×2848

Figure 3 and 4 show the comparison of subjective quality of the thumbnail image. It shows almost same at first sight but it has certain difference in detail of the objects. FFD method shows a little bit dark compare to FD result because of errors. Especially, due to error propagation, it become darker when it goes to bottom-right part of the image than upper-left part. BFD method compensates the degraded part effectively. So, it looks more similar to FD result than FFD method.

Table II shows the average peak signal-to-noise ratio (PSNR) of each method. We compute the PSNR of the FFD and BFD with reference to the thumbnail image generated by FD. As shown in this table, proposed method can make higher PSNR for all test images. Especially, *Flowershop* image shows the outstanding improvement.

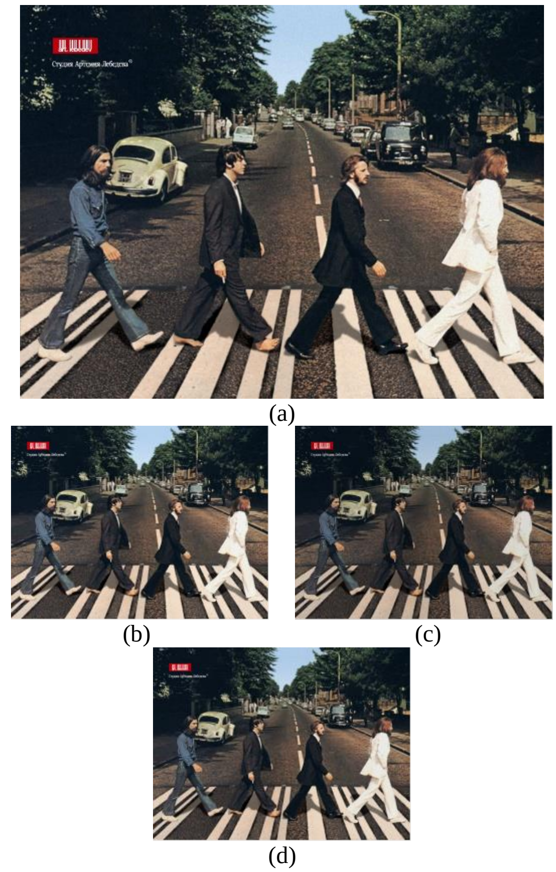

Figure 3. The *Beatles* image (a)Original image, (b)FD, (c)FFD, (d)BFD.

Figure 4. The *TVshow* image (a)Original image, (b)FD, (c)FFD, (d)BFD.

TABLE II. COMPARISON RESULT OF THE PSNR (dB).

Image	FFD [6]	BFD	Δ
Album1	34.24	36.01	+1.77
Album2	33.30	34.98	+1.68
Hirmer	33.69	35.08	+1.39
Soccer	33.74	35.10	+1.36
Beatles	32.45	34.45	+2.00
TVshow	34.45	36.27	+1.82
Flowershop	24.84	28.75	+3.91

V. CONCLUSIONS

We have presented the block-based error compensation method for thumbnail extraction. It uses the error distribution data of each intra prediction modes. It determines the compensation value with the mean and the deviation value of the errors. Also, we divide the image into some blocks and adaptively set the weighting factors by considering the location of the pixels. Thus, it have lower value when it goes to upper-left and higher value when it goes to bottom-right. By using the proposed method, we can successfully compensate the error of thumbnail which is extracted by fast extraction algorithm. It achieves better result both of subjective and objective comparison.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1005) supervised by the IITP(Institute for Information&communications Technology Promotion)

REFERENCES

[1] B. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video," IEEE Trans. Circuits and Syst. Video Technol., vol. 5, no. 6, pp. 533-540, Dec. 1995.

[2] J. H. Song and B. L. Yeo, "Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video," IEEE Trans. Circuits Syst. Video Technol., vol. 9, no. 7, pp. 1100-1114, Oct. 1999.

[3] S. J. Suh, S. S. Chun, M. H. Lee, and S. H. Sull, "Efficient Image Down-Conversion for Mixed Field/Frame-mode Macroblocks," IEEE Electron. Lett., Vol. 39(6), pp. 514-515, Mar. 2003.

[4] ISO/IEC JTC1/SC29/WG11 (MPEG), "Coding of audio-visual objects-Part 10: Advanced Video Coding," International Standard 14496-10, ISO/IEC, 2004.

[5] C. Chen, P. H. Wu, and H. Chen, "Transform-Domain Intra Prediction for H.264," IEEE ISCAS, pp. 1497-1500, May. 2005.

[6] E. S. Kim, T. W. Um, and S. J. Oh, "A Fast Thumbnail Extraction Method in H.264/AVC Video Streams," IEEE Trans. Consumer Electronics, vol. 55, no. 3, pp. 1424-1430, Aug. 2009.

[7] S. J. Yu, M. K. Yoon, E. S. Kim, C. B. Sohn, D. G. Sim, and S. J. Oh, "An Efficient Thumbnail Extraction Method in H.264/AVC Bitstreams," THE KOREAN SOCIETY OF BROADCAST ENGINEERS, vol. 13, no. 2, pp. 222-235, Mar. 2008.

[8] M. H. Kim, H. J. Lee, and S. H. Sull, "Fast Thumbnail Generation in Integer DCT Domain for H.264/AVC," IEEE Trans. Consumer Electronics, vol. 57, no. 2, pp. 589-596, May. 2011.

[9] M. K. Yoon, Y. S. Lee, C. B. Sohn, H. C. Park, C. B. Ahn, and S. J. Oh, "An Efficient Error Compensation Method for Thumbnail Extraction in H.264/AVC Bitstrems," THE KOREAN SOCIETY OF BROADCAST ENGINEERS, vol. 13, no. 5, pp. 622-635, Sep.