

# Simple and Efficient Deblocking Algorithm for H.264/AVC Decoder

Jung-Ah Choi and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)  
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea  
E-mail: {jachoi, hoyo}@gist.ac.kr

**Keywords:** H.264/AVC, video coding, deblocking filter

**Abstract** – Block-based coding in H.264/AVC creates discontinuities at block boundaries. The blocking artifact is regarded as the most visible artifact in block-based video coding. In H.264, a deblocking filter is used to reduce such a blocking artifact. However, it requires a significant amount of complexity and it occupies one-third of the computational complexity of the decoder. In this paper, we propose a simple and efficient deblocking algorithm. It consists of mode decision and filtering. For mode decision, we define three different modes: weak mode, strong mode, and special mode. For filtering, we design two new filters for strong mode and special mode. Experimental results show that the proposed deblocking algorithm provides comparable objective quality, better subjective video qualities, and lower computational complexity, compared to the deblocking filter in H.264.

## 1. INTRODUCTION

The H.264 video coding standard was developed through the Joint Video Team (JVT) from the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG) standardization. H.264 is the most advanced development for video coding [1].

In H.264, block-based discrete cosine transform (DCT) and block-based motion compensated prediction are used to reduce both spatial and temporal redundancies. In block-based coding, each macroblock in the frame is independently coded but the transform does not take into account the correlation between block boundaries; blocking artifact is created at block boundaries. The blocking artifact generates decoded frames unacceptable for human eyes at low bit rates. They also limit the maximum compression performance that can be achieved. Therefore, it is necessary to decrease such a blocking artifact at block boundaries.

In order to decrease the undesired blocking artifact, we employ a deblocking filtering technique after the inverse discrete cosine transform. In H.264, the deblocking filtering process goes through three operations: boundary strength (BS) selection, mode decision, and filter implementation. After we check BS in every  $4 \times 4$  block boundary in each macroblock, we determine the proper filter according to the selected BS value. Then, we apply the suitable filtering adaptively. These processes are carried out repeatedly for each macroblock. The filtering is applied to both encoder and decoder.

The H.264 deblocking filter shows good performance; however, there is a large amount of computational complexity. Specifically, the deblocking filter can easily account for one-third of computational complexity in the decoder [2]. Therefore, a new deblocking filtering algorithm of low complexity is required.

In this paper, we propose a simple and efficient deblocking algorithm for H.264/AVC. The algorithm consists of mode decision and filtering. In the H.264 deblocking filter, there are only two filtering modes: normal mode and special mode. In the proposed algorithm, we define three different modes: weak mode, strong mode, and special mode based on BS parameters. For strong mode and special mode, we design a new deblocking filter of low computational complexity and better subjective quality than the deblocking filter in the H.264 standard.

In order to measure simplicity, we calculated computational complexity of the proposed filter and compared it to that of the conventional filter. Also, we used both objective quality measure and subjective video quality measure to estimate quality of decoded frames. Here, we used a modified Watson's digital video quality metric (MWDVQM) to evaluate subjective video quality based on the human visual system (HVS) [3].

From experimental results, it has been shown that the proposed deblocking algorithm gives similar objective result to the conventional method. However, subjective video quality measured by MWDVQM is improved. The computational complexity of the proposed deblocking method is much lower than the deblocking filter in H.264.

## 2. DEBLOCKING FILTER IN H.264/AVC

In H.264, the deblocking filter is used to decrease blocking artifact at block boundaries. The filtering is done first from left to right vertically and then from top to bottom on the horizontal boundaries. Figure 1 represents vertical and horizontal boundaries in one macroblock. Each square stands for a block of  $4 \times 4$  pixels.

In order to apply a filter to each macroblock, we use filtered pixels at the top and on the left of the current macroblock. Luminance and chrominance components are separately processed. In H.264, the deblocking filtering process consists of three operations: BS selection to determine the filter strength, mode decision to select the suitable filter, and filter implementation [4].

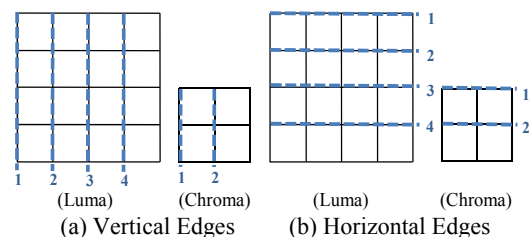


Fig. 1. Sequential order of vertical edges and horizontal boundaries in a macroblock

### 3. PROPOSED DEBLOCKING ALGORITHM

The deblocking filter in H.264 has only two filtering modes: normal mode and special mode. If BS is from 1 to 3, normal mode is selected. Otherwise, special mode is chosen as a filtering mode. However, characteristics of BS equal to 1 or 2 and BS equal to 3 is not same. Thus, we need to separate these two cases. Another drawback of the H.264 deblocking filter is an amount of complexity of filtering. Despite of good performance, cost of computational complexity is too much. Hence, we need to propose a simple deblocking filter. The proposed algorithm is comprised of two stages: mode decision and filtering. For mode decision, we reorganize filtering modes, and for filtering, we design a simple and efficient deblocking filter for strong mode and special mode.

#### 3.1 Mode Decision

The deblocking filter in H.264 consists of two filtering modes, normal mode for low BS and special mode for high BS. However, based on BS parameters, we classify filtering modes into three groups: weak mode, strong mode, special mode. The flowchart of the proposed algorithm is shown in Fig. 2.

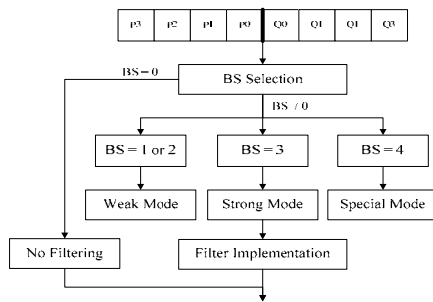


Fig. 2. Flowchart of the proposed algorithm

#### 3.2 Filter Implementation

1) Filter for weak mode: If BS value is less than 2 (BS = 1 or 2), we apply normal mode filter for low BS in the deblocking filter in H.264 [2]:

$$P0' = P0 + \Delta_0 \quad (1)$$

$$Q0' = Q0 - \Delta_0 \quad (2)$$

$$\text{where } \Delta_0 = (4(Q0 - P0) + (P1 - Q1) + 4) \gg 3. \quad (3)$$

2) Filter for strong mode: We describe strong mode filtering for luminance. This algorithm is a modification to the algorithm proposed by Ramkishor [5]. The filtering process is shown in Fig. 4 (a).

First, we calculate the size of discontinuity,  $d$ . In this mode, four pixels around the block boundary are filtered. Thus, we can predict that a smoothing line is the connection of P2 and Q2. In order to match pixel values with the predicted smoothing line, we can calculate pixel values as

$$P0' = P0 + (d \gg 1) - \Delta \quad (4)$$

$$P1' = P1 + (d \gg 2) + (\Delta \gg 1) \quad (5)$$

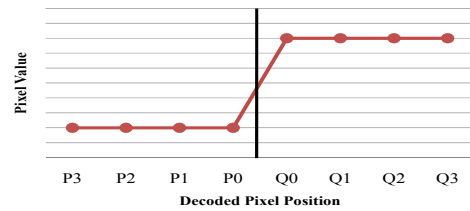
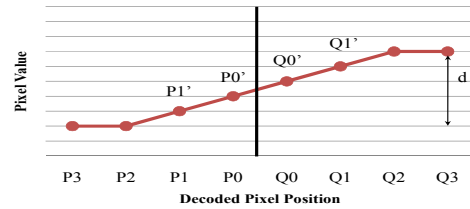
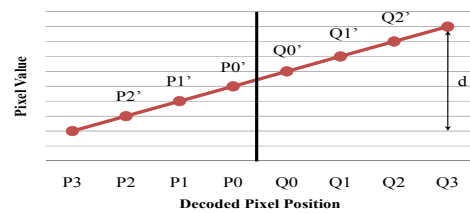


Fig. 3. Block boundary after decoding



(a) Filtering for edges with BS equal to 3



(b) Filtering for edges with BS equal to 4

Fig. 4. Block boundary after deblocking filtering

$$Q0' = Q0 - (d \gg 1) + \Delta \quad (6)$$

$$Q1' = Q1 - (d \gg 2) - (\Delta \gg 1) \quad (7)$$

The amount of modification that will be applied to each of the edge samples is obtained as

$$\Delta = d / 5 \quad (8)$$

The algorithm is applied to all blocks.

3) Filter for special mode: In strong mode filter, only four pixels around the block boundary are filtered. This filter operation avoids blurring the regions with high spatial details, but restricts the filter effect for regions with strong blocking artifact. The strength of filtering can be improved if the filter length becomes long. In case of BS equal to 4, the smoothing line is changed to the connection of P3 and Q3. In addition to the result of strong mode filter, we modify two more pixels as

$$P2' = P2 + (d \gg 3) + (\Delta \gg 2) \quad (9)$$

$$Q2' = Q2 - (d \gg 3) + (\Delta \gg 2) \quad (10)$$

and the  $\Delta$  value is changed as

$$\Delta = d / 7 \quad (11)$$

One dimensional view of block boundaries after deblocking filtering is shown in Fig. 4(b). This kind of filtering requires less amount of computation compared to N-tap low pass filtering in the conventional deblocking filter. The proposed algorithm requires very simple control mechanism for applying filtering in comparison to other well-known algorithms.

## 4. EXPERIMENTAL RESULTS

The proposed deblocking algorithm is implemented on JM 11.0. We have tested four QCIF (176×144) video sequences (Foreman, Stefan, Coastguard, and Hall Monitor). For each sequence, 100 frames are encoded. The frame rate is 30 fps. CABAC is adopted as the entropy coding method. Experiments were conducted for four quantization parameters: QP = 28, 32, 36, and 40.

### 4.1 Objective Measure

Table 1 shows the results for different test sequences. Larger value means better reconstruction. In most sequences, the proposed algorithm provides a negligible loss of PSNR, but same or better objective quality at low bitrate.

| Sequence     | QP | No Processing | JM 11.0 | Proposed Method |
|--------------|----|---------------|---------|-----------------|
| Foreman      | 28 | 36.36         | 36.5    | 36.52           |
|              | 32 | 33.57         | 33.8    | 33.82           |
|              | 36 | 30.91         | 31.19   | 31.19           |
|              | 40 | 28.20         | 28.38   | 28.41           |
| Stefan       | 28 | 34.40         | 34.51   | 34.5            |
|              | 32 | 30.87         | 30.95   | 30.97           |
|              | 36 | 27.76         | 27.81   | 27.81           |
|              | 40 | 24.87         | 24.86   | 24.87           |
| Coastguard   | 28 | 34.14         | 34.2    | 34.2            |
|              | 32 | 31.12         | 31.17   | 31.19           |
|              | 36 | 28.61         | 28.67   | 28.67           |
|              | 40 | 26.40         | 26.43   | 26.45           |
| Hall Monitor | 28 | 37.22         | 37.47   | 37.46           |
|              | 32 | 34.32         | 34.63   | 34.61           |
|              | 36 | 31.46         | 31.76   | 31.66           |
|              | 40 | 28.69         | 28.86   | 28.86           |

Table 1. PSNR values for different test sequences

### 4.2 Video Quality Measure (VQM)

Since PSNR measure does not reflect the human visual system properly, perceptual quality measure is required to evaluate quality of decoded frames [6]. A reliable video quality metric (VQM) is needed to measure the proposed method. VQM reflects human visual perception and helps for measuring visual quality of low bitrate videos. We used MWDVQM. This method is based on the Watson's digital video quality metric [7][8]. It computes the visibility of artifact expressed in the DCT domain via DCT coefficients. The detail explanation of MWDVQM is described in [3]. We have tested first frame of each sequence when QP is equal to 40. Table 2 shows simulation results. In Fig. 5, MWDVQM result of Foreman is represented when QP is equal to 40. In MWDVQM, lower values represent better quality. In case of Foreman sequence, the proposed

algorithm shows better results than JM 11.0; however, even JM 11.0 gives lower video quality than unfiltered results. For other sequences, the proposed method shows better results than JM 11.0.

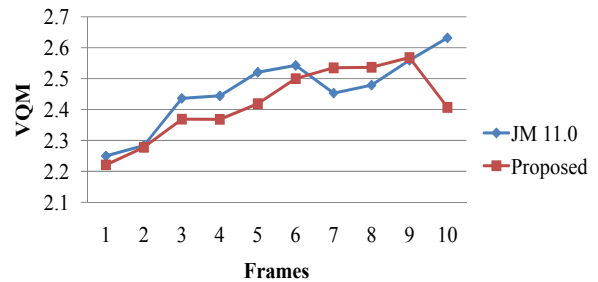


Fig. 5. Video quality comparison for Foreman sequence

| Sequence     | No Processing | JM 11.0 | Proposed Method |
|--------------|---------------|---------|-----------------|
| Foreman      | 2.2482        | 2.2500  | 2.2213          |
| Stefan       | 3.3523        | 3.3423  | 3.3314          |
| Coastguard   | 3.0350        | 3.0089  | 2.9778          |
| Hall Monitor | 2.2457        | 2.2460  | 2.2063          |

Table 2. Video quality comparison for different test sequences

### 4.3 Subjective Measure

The improvement in subjective quality is also an important feature to evaluate the deblocking algorithm. Figure 6 shows results of JM 11.0 and the proposed algorithm applied to Foreman sequence. The experiment was conducted for QP equal to 40. Figure 7 represents magnified views of Fig. 6. The original frame and deblocked frames are given. Even few amplitude edges are blurred, the proposed approach significantly outperforms the JM 11.0 deblocking filter.

### 4.4 Complexity Analysis

In Table 3, we compare average computations of the JM 11.0 deblocking filter and the proposed deblocking filter in the aspect of additions, multiplications, and shift operations. We compute the number of operations needed to filter one macroblock. The data were obtained by applying filters to four test sequences in QCIF format. It is easy to see that the proposed deblocking filter needs small amount of the operations than that of JM 11.0. As a result, an amount of complexity is significantly reduced than the JM 11.0 deblocking filter.

## 5. CONCLUSION

This paper proposes a simple and efficient deblocking algorithm for reduction of computational complexity and improvement of subjective quality. The proposed algorithm includes two stages: mode decision and filter implementation. We reorganize filtering modes and apply newly designed filters. Three computationally simple and efficient deblocking filters are presented. The performance of the algorithm is measured using objective, subjective



(a) Original frame (b) Decoded frame without deblocking filter (c) JM 11.0 (d) Proposed algorithm  
 Fig. 6. Perceptual quality comparison of deblocking algorithms for Foreman sequence (QP = 40)



(a) Original frame (b) Decoded frame without deblocking filter (c) JM 11.0 (d) Proposed algorithm  
 Fig. 7. Magnified views of the 49<sup>th</sup> frame of Foreman sequence

| Algorithm          | Operations       | Foreman           | Stefan           | Coastguard        | Hall Monitor      |
|--------------------|------------------|-------------------|------------------|-------------------|-------------------|
| JM 11.0            | Additions        | 464.5678          | 659.5753         | 682.9405          | 214.0697          |
|                    | Multiplications  | 81.68152          | 116.1827         | 120.1718          | 37.54485          |
|                    | Shift Operations | 135.2829          | 193.0346         | 199.3029          | 61.91697          |
|                    | Total Complexity | <b>681.5322</b>   | <b>968.7926</b>  | <b>1002.415</b>   | <b>313.5315</b>   |
| Proposed Algorithm | Additions        | 436.076465        | 638.376061       | 649.046364        | 191.197778        |
|                    | Multiplications  | 44.2706061        | 107.996061       | 107.158485        | 28.74             |
|                    | Shift Operations | 128.060303        | 187.576869       | 190.627374        | 56.0470707        |
|                    | Total Complexity | <b>608.407374</b> | <b>933.94899</b> | <b>946.832222</b> | <b>275.984848</b> |

Table 3. Computational complexity comparison for different video sequences

measure, and computational complexity comparison. The algorithm had a comparable objective image quality to H.264 deblocking algorithm. Subjective image quality is improved and complexity of proposed algorithm is lower than deblocking filters in H.264. Specifically, the number of multiplications for filtering is significantly reduced. We believe that our algorithm can be used in many applications of H.264/AVC, especially at low bitrate.

**ACKNOWLEDGEMENTS**

This work was supported in part by ITRC through RBRC at GIST.

**REFERENCES**

[1] Joint Video Team of ITU-T and ISO/IEC JTC 1, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Doc. JVT-G050, March 2003.

[2] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive Deblocking Filter," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 614-619, July 2003.

[3] F. Xiao, "DCT-based Video Quality Evaluation," MSU Graphics and Media Lab (Video Group), Winter 2000.

[4] E.G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley & Sons, 2003.

[5] K. Ramkishor and P. Karandikar, "A Simple and Efficient Deblocking Algorithm for Low Bit-Rate Video Coding," IEEE International Symposium on Consumer Electronics, Hongkong, China, Dec. 2000.

[6] A.B. Watson, Digital Images and Human Vision, MIT Press, 1993.

[7] A.B. Watson, "Toward a perceptual video quality metric," Human Vision, Visual Processing, and Digital Display, Proc. of SPIE, vol. 3299, pp. 139-147, July 1998.

[8] A.B. Watson, J. Hu, J.F. McGowan III, "DVQ: A digital video quality metric based on human vision," Journal of Electronic Imaging, vol. 10, pp. 20-29.