# In-loop Deblocking Filter for H.264/AVC Video

Gulistan Raja[1]and Muhammad Javed Mirza[2]

*Abstract*—**Presently block based transform coding using discrete cosine transform is the most popular approach for video compression. However, when the image is partitioned into blocks, it results is blocking artifacts due to lack of inter-correlation between blocks. These artifacts has much more annoying effect at low bit rates. This paper compares the performance of the in-loop deblocking filter of emerging H.264/AVC standard at low bit rates. Different sequences of range of typical content for low & high latency applications were used for comparison. Our test results show that H.264's in-loop deblocking filter can significantly reduce the blocking artifacts.**

## I. INTRODUCTION

MOST of the video compression standards like ITU-H.263[1,2], MPEG-4[3], H.264[4] use JPEG related block based transform coding technique to exploit spatial redundancy. The basic approach is to divide the whole image into 8 x 8 blocks, transform each block using discrete cosine transform, quantized and entropy coded. The quantization step divides transformed coefficients by quantization table by which most of the DCT coefficients in each block falls into dead zone. As a result, there is only one DC & few coefficients are present at low bit rates. The net effect is loss correlation between adjacent blocks and discontinuities on edges of blocks [5-6]. As a consequence, reconstructed images suffer from visually annoying effects known as blocking effects or blocking artifacts. Another source of blocking artifacts in video is motion compensated prediction. Motion compensated blocks are generated by copying interpolated pixel data from different locations of possibly different reference frames. This results in discontinuities on the edges of copied blocks due to fact that there is almost never a perfect match for this data. Additionally, in copying process, existing edge discontinuities in reference frames are carried into the interior of the block to be compensated, which results in visually disturbing artifacts. There are two main techniques deployed to counter the blocking artifacts: post filters and in-loop deblocking filters. Table 1 list the deblocking filters employed by various video compression standards [7-8]. In the post-filter technique, filter is applied after the decoder and makes use of decoded parameters. It operates on display buffer outside the coding loop. The use of post filter is optional in most standards as it is not a normative part of standards. The in-loop filters operate within coding loop. The filter is applied to the reconstructed frame both in encoder and decoder. Filtered frames are used as reference frames for motion compensation of subsequent coded frame. Applying the filter within coding loop can improve the quality of reconstructed frame, which results in improvement in the accuracy of motion-compensated prediction for the next encoded frame since the quality of the prediction reference is improved. H.264/AVC [4] employs mandatory adaptive in-loop deblocking filter for the reduction of blocking artifacts. The rest of the paper is organized as follows. Section II describes the working of H.264/AVC in-loop deblocking filter. The test methodology is described in section III. The results are explained in section IV while section V concludes the paper.

Table 1: Deblocking filters for various standards

| ***Standard*** | ***Deblocking Filter*** |
|---|---|
| H.261 | Optional in-loop filter |
| MPEG-1 | No filter |
| MPEG-2 | No Filter, post-filter processing often used |
| H.263 | No filter |
| MPEG-4 | Optional in-loop filter, post-filter processing suggested |
| H.264 | Mandatory in-loop filter, post- filter processing may also be used |

## II. H.264/AVC IN-LOOP DEBLOCKING FILTER

H.264 employs an adaptive in-loop deblocking filter after the inverse transform in the encoder and decoder respectively. The filter is applied to the each decided macroblock to reduce the blocking artifacts without reducing the sharpness of the picture. The net effect is in improvement of the subjective quality of compressed video. The output of filter is used for motion compensated prediction for further frames. The deblocking filter process is invoked for the luminance and chrominance components separately. Filtering is applied to vertical or horizontal edges of the block except for the edges on the slice boundaries. The order of the filtering at a macroblock level is shown in figure1. Initially, 4 vertical edges of the luminance component i.e., VLE1, VLE2, VLE3 and VLE4 are filtered. Then, horizontal edges of the luminance component i.e., HLE1, HLE2, HLE3, and HLE4 are filtered. Finally, vertical edges of chrominance component, VCE1, VCE2 and horizontal edges of chrominance component HCE1, HCE2 are filtered respectively. It is also possible for the filter to alter the filter strength or to disable the filter.

[1] Gulistan Raja is with the Electrical Engineering Department, University of Engineering & Technology, Taxila, Pakistan, (corresponding author phone: +92-51-9047522; fax: +92-51-9047420; e-mail: raja @uettaxila.edu.pk)

[2] Muhammad Javed Mirza is with Advanced Engineering Research Organization, Wah Cantt. Pakistan.

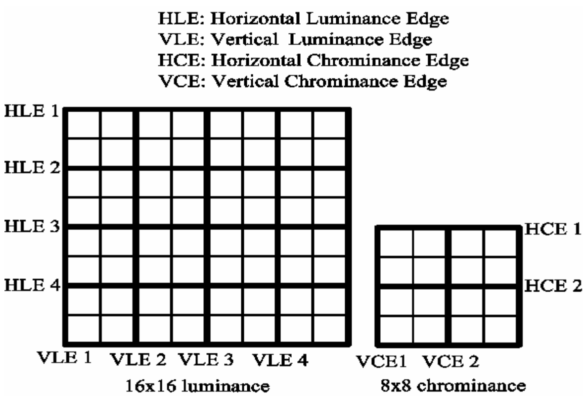

Figure 1. Filtering order at macroblock level

The filtering operation affects three samples on either side of the boundary. The four samples on vertical edge or horizontal edge in adjacent blocks are p0, p1, p2, p3 and q0, q1, q2, q3 respectively are shown in figure 2.

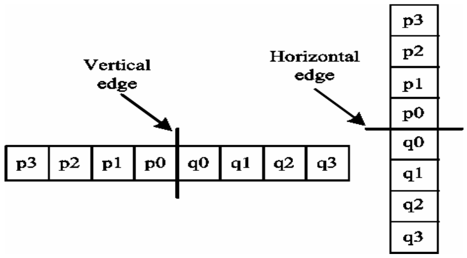

Figure 2. Adjacent samples to vertical and horizontal edges

The operation of deblocking filter can be divided into three main steps, i.e., filter strength computation, filter decision and filter implementation respectively.

### A. *Filter Strength Computation*

The filter strength i.e., the amount of filtering is computed with the help of parmameter boundary strength (bS). The boundary strength (bS) of the filter depends on the current quantizer, macroblock type, motion vector, gradient of the image samples across the boundary and other parameters as shown in figure 3. The boundary strength is derived for each edge between neighbouring 4 x 4 luminance blocks and for each edge, bS parameter is assigned an integer value for 0 to 4. The rules for selecting integer value for bS parameter boundary strength (bS) is illustrated in flow chart of figure 3. The bS values for filtering of chrominance block edges are not calculated independently and same values calculated for luminance edges are used. Application of these rules results in strong filtering in the areas where there is significant blocking distortion, such as boundary of intra coded macroblock or a boundary between blocks that contain coded coefficients.

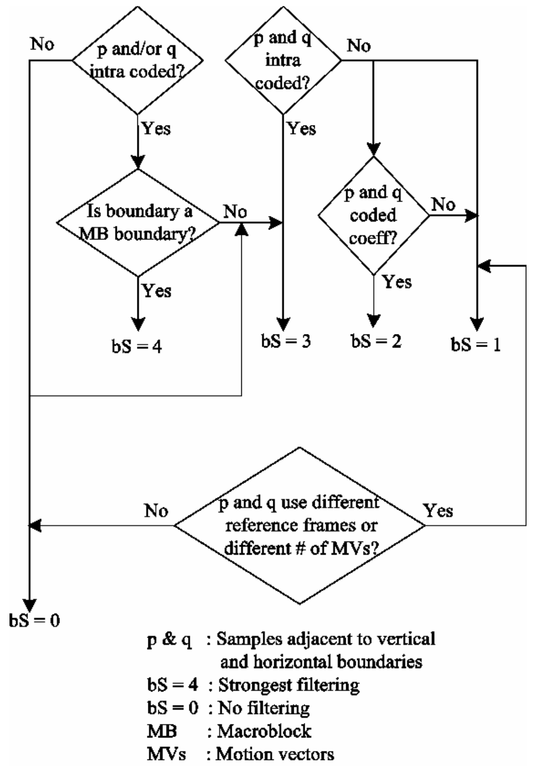

Figure 3. Boundary strength (bS) computation flowchart

### B. *Filtering Decision*

The filtering decision does not depend only on non-zero boundary strength, i.e., we cannot start filtering on the basis of non-zero boundary strength only. Deblocking filtering may not be needed, even in the case of non-zero boundary strength. This is especially true when we have real sharp transitions across the edge. Applying filter to such edges will result in blurry image. When pixels do not change much across the block edge in very smooth regions, blocking artifacts are most noticeable. Therefore, another condition in addition to non-zero boundary strength is required for filtering decision. As a consequence, set of samples p2, p1, p0 and q0, q1, q2 are filtered only, if they have met following two conditions.

- bS should be greater than zero
- $abs\ (p0-pq0) < \alpha \ \&\ abs\ (p1-p0) < \beta \ \&\ abs\ (q1-q0) \leq \beta$

Where α and β are the thresholds defined in the standard [4], they increase with the average quantizer QP of the two blocks p and q. When QP is small, the small transition across the boundary is likely due to image features rather than of blocking effects that should be preserved and so the thresholds α and β are low. When QP is large, blocking distortion is likely to be significant and α and β are higher so more boundary samples are filtered. The filter can be switched off, when there is a real significant change across the boundary of an original image, which is not due to blocking distortion.

### C. *Filter Implementation*

The luminance deblocking filtering is performed on four 16-sample edges and on two 8-sample edges for chrominance components in horizontal and vertical directions respectively. Following rules apply for filter implementation [4,9].

- Pixel values above and to the left of the current MB that may have already been modified by filter on previous MBs shall be used as input to the filter on the current MB and may be further modified during the filtering of current MB.
- Pixel values modified during filtering of vertical edges are used as input for filtering of horizontal edges for the same MB.
- Pixel values modified during the filtering of previous edges are used input for the filtering of the next edge in both horizontal and vertical directions.

The procedure for calculating filtered pixel samples is as follows. When integer values of boundary strength is 1 to 3, the steps required for computing filtered samples is:

- A 4-tap filter is applied with inputs p1, p0, q0, q1, producing filtered outputs p’0 and q’0.
- If abs (p2-p0) is less than threshold $\beta$, another 4-tap filter is applied with inputs p2, p1, p0, q0, producing filtered output p’1 for luminance component only.
- If abs (q2-q0) is less than the threshold $\beta$, a four tap filter is applied with inputs q2, q1, q0, p0, producing filtered q’1 for luminance component.

When integer value of boundary strength equals to 4, following procedure is used to get filtered output:

- If abs (p2-p0) is less than $\beta$ and abs(p0-q0) is less than $\alpha/4$ and current block is luminance block than p’0 is produced by 5-tap filtering of p2, p1, p0, q0, q1 and p’1 by 4-tap filtering of p2, p1, p0, q0 and p’2 is produced by5-tap filtering of p3,p2, p1, p1, q0 respectively.
- Otherwise p’0 is produced by 3-tap filtering of p1, p0 and q0.
- If abs(q2-q0) is less then $\beta$ & abs(p0-q0) is less then $\alpha/4$ & current block is luminance block than q’0 is produced by 5-tap filtering of q2,q1,q0,p0,p1 and q’1 is produced by 4-tap filtering of q2, q1, q0, p0 and q’2 by 5-tap filtering of q3, q2,q1,q0,p0 respectively.
- Else q’0 is produced by 3-tap filtering of q1, q0, p1.

The whole procedure of generating filtered sample values is illustrated in figure 4.

## III. TEST METHODOLOGY

H.264 Joint model reference software version encoder [10] is used for tests. We have used QCIF (176 x 144) and CIF (352 x 288) video sequences. The set of sequences represent a range of typical video content from low and high latency applications. The QCIF sequences used for experimentation are MISS AMERICA, CARPHONE, TENNIS and FOREMAN while CIF sequences used are HALL, COASTGUARD, MOBILE & CALENDAR and TEMPETE respectively. We have used 50 frames of sequences for QCIF and CIF encoding. QCIF sequences were encoded at 15 fps and CIF sequences were encoded at 30 fps frame rate respectively. Each sequence was coded a five different bit rates. The coding performance are compared on output bit rate and PSNR (peak signal-to-noise ratio) of the encoded video sequences. Only the luminance component is taken into consideration since human visual system is less sensitive to color than to luminance.

## IV. SIMULATION RESULTS

H.264 encoder was configured for quarter pixel motion vector resolution, five frames for inter motion, context-based adaptive binary coding (CABAC) for symbol coding, rate distortion optimized mode decision. Also Hadamard transform and Inter search range of 16x16, 16x8, 8x16, 8x8, 4x8 , 8x4, 4x4 were used. Both encoders were

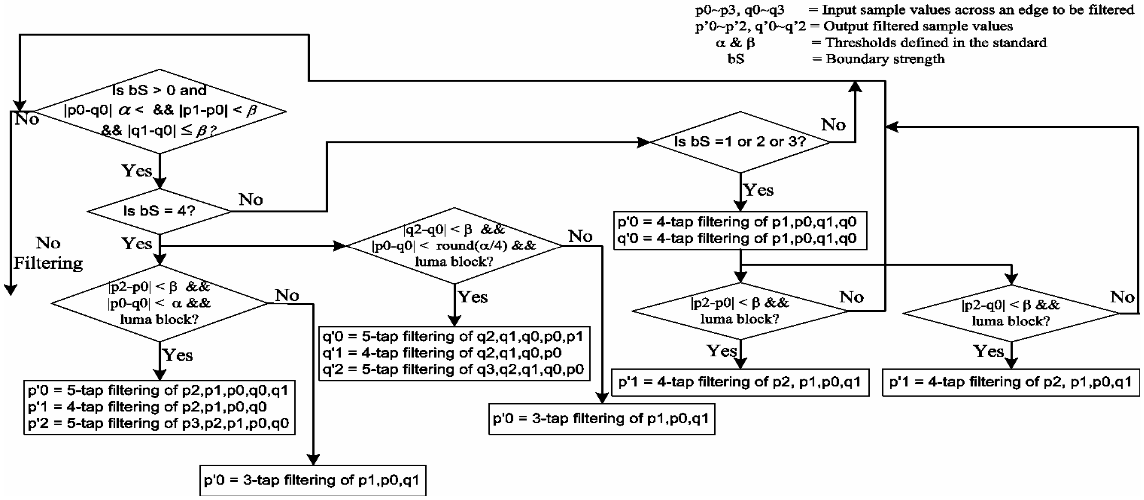

Figure4. Filter Implementation

configured to have five frames for inter motion search. The PSNR is compared by comparing coding performance with and without deblocking filter mode. Table 2 shows the luminance PSNR for various QCIF sequences while Table 3 shows CIF sequences.

Table2: Average luminance PSNR at different bit rates for QCIF sequences with- and without deblocking filter

| Sequence | Bit rate (Kbps) | Luminance PSNR (dB) | |
|---|---|---|---|
| | | No filter | With filter |
| Miss America | 41.40 | 43.35 | 43.37 |
| | 61.57 | 44.24 | 44.26 |
| | 82.15 | 44.75 | 44.78 |
| | 102.31 | 45.11 | 45.15 |
| | 122.54 | 45.46 | 45.49 |
| Car phone | 41.40 | 36.92 | 36.97 |
| | 61.87 | 38.75 | 38.83 |
| | 82.27 | 40.16 | 40.19 |
| | 102.56 | 41.05 | 41.08 |
| | 122.70 | 41.75 | 41.79 |
| Tennis | 41.12 | 32.47 | 32.48 |
| | 61.58 | 34.15 | 34.18 |
| | 81.84 | 35.32 | 35.42 |
| | 102.20 | 36.25 | 36.28 |
| | 122.54 | 36.96 | 36.99 |
| Foreman | 41.47 | 35.19 | 35.20 |
| | 61.92 | 36.88 | 36.92 |
| | 82.6 | 37.98 | 38.03 |
| | 102.88 | 38.87 | 38.88 |
| | 123.28 | 39.61 | 39.64 |

Table3: Average luminance PSNR at different bit rates for CIF sequences with- and without deblocking filter

| Sequence | Bit rate (Kbps) | Luminance PSNR (dB) | |
|---|---|---|---|
| | | No filter | With filter |
| Hall | 205.08 | 37.49 | 37.51 |
| | 407.95 | 39.05 | 39.07 |
| | 610.36 | 39.88 | 39.91 |
| | 811.90 | 40.45 | 40.46 |
| | 1012.29 | 40.89 | 40.91 |
| Coast guard | 207.75 | 28.89 | 28.93 |
| | 412.15 | 30.97 | 31.01 |
| | 615.50 | 32.47 | 32.50 |
| | 820.18 | 33.59 | 33.62 |
| | 1023.70 | 34.59 | 34.60 |
| Mobile & Calendar | 209.49 | 26.94 | 26.97 |
| | 414.99 | 29.50 | 29.52 |
| | 615.03 | 30.99 | 31.02 |
| | 820.89 | 32.13 | 32.15 |
| | 1022.62 | 40.89 | 40.91 |
| Tempete | 207.06 | 29.45 | 29.49 |
| | 412.20 | 31.93 | 31.96 |
| | 616.77 | 33.48 | 33.51 |
| | 819.44 | 34.65 | 34.68 |
| | 1023.35 | 35.62 | 35.66 |

Rate PSNR graph with and without in-loop filter for QCIF car phone sequence and CIF coastguard sequence at various bit rates are shown in figure 5 and figure 6 respectively. Although these differences are not very significant but their perceptual quality is quite significant as shown in figure 7. The main reason is that the blocking artifacts are structural disturbance, and are sometimes ‘‘buried’’ in the massively accumulated across the-board pixel-wise error. Therefore their significance in perceptual visual quality assessment is not reflected correctly in the conventional PSNR measure [11]

## V. CONCLUSION

In this paper, we have presented the performance of adaptive in-loop deblocking filter for emerging H.264/AVC standard. Our test analysis shows that adaptive filtering employed by emerging JVT H.264/AVC standard removes the blocking artifacts significantly. However, the in-loop filtering takes One-third of computational resources of the decoder according to an analysis of run-time profiles of decoder sub-functions [12].

## VI. REFERENCES

[1] “Video coding for low bit rate communication,” Draft ITU-T Recommendation H.263, Dec. 5, 1995.

[2] H.263+ Video Group, “Draft text of H.263+,” ITU SG15 LBC Experts Group ITU-LBC-97-094 Draft A, Feb. 25–28, 1997

[3] MPEG Video Group, “MPEG-4 video verification model version 8.0,” ISO/IEC JTC1/SC29/WG11 N1796, July 1997.

[4] 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),” Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, May 2003.

[5] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, “A deblocking filter with two separate modes in block-based video coding,” IEEE Trans. Circuits Syst. Video Technology, vol. 9, pp. 156–160, Feb. 1999.

[6] Y.-L. Lee and H. W. Park, “Loop filtering and post-filtering for low-bit rates moving picture coding,” Signal Processing: Image Communication., vol.16, pp. 871–890, 2001.

[7] Iain E.G. Richardson: “Video Codec Design,” Publisher: John Wiley & Sons Ltd, England, 2003, ISBN: 0-471-48553-5.

[8] Gulistan Raja and Muhammad Javed Mirza, “Performance Comparison of Advanced Video Coding H.264 Standard with Baseline H.263 and H.263+ Standards,” in Proc. 4th IEEE International Symposium on Communications & Information Technologies, Oct 26-29, 2004, Sapporo, Japan, pp. 743-746.

[9] Atul Puri, Xuemin Chen and Ajay Luthra, “Video coding using the H.264/MPEG-4 AVC compression standard” Signal Processing: Image Communication., vol.19, pp. 793–849, 2004.

[10] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Joint Model Reference Software Version 9.2.

[11] F. Pan, X. Lin, S. Rahardja, W. Lin, E. Ong, S. Yao, Z. Lu, and X. Yang, ” A locally adaptive algorithm for measuring blocking artifacts in images and videos,” Signal Processing: Image Communication, 19, 2004, pp. 499–506.

[12] Michael Horowitz, Anthony Joch, Faouzi Kossentini, and Antti Hallapuro,” H.264/AVC Baseline Profile Decoder Complexity Analysis,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, July 2003, pp. 704-716.

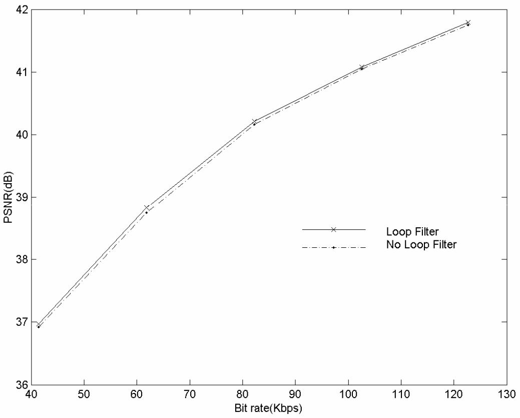

Figure 5. Rate-PSNR of QCIF Car phone sequence at various bit rates with & without in-loop filter

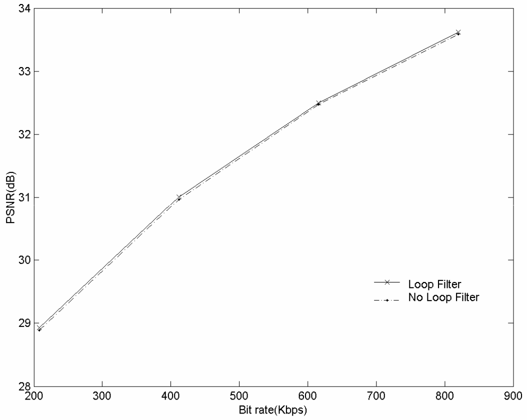

Figure 6. Rate-PSNR of CIF Coastguard sequence at various bit rates with & without in-loop filter

(a) (b)

Figure 7. Frame 34 of QCIF FOREMAN sequence encoded at 40Kbps
(a) Without filter (b) With loop filter