A deblocking filter with two separate modes in block-based video coding

Sung Deuk Kim, Jaeyoun Yi, and Jong Beom Ra
Dept. of Electrical Engineering
Korea Advanced Institute of Science and Technology
373-1, Kusongdong, Yusonggu, Taejon, Republic of Korea
Email: sdkim@issserver.kaist.ac.kr, jyyi@issserver.kaist.ac.kr, jbra@ee.kaist.ac.kr

ABSTRACT

This paper presents a method to remove blocking artifacts in low bit-rate block-based video coding. The proposed algorithm has two separate filtering modes, which are selected by pixel behavior around the block boundary. In each mode, proper one-dimensional filtering operations are performed across the block boundary along horizontal and vertical directions, respectively. In the first mode corresponding flat regions, a strong filter is applied inside the block as well as on the block boundary, because the flat regions are more sensitive to the human visual system (HVS) and the artifacts propagated from the previous frame due to motion compensation are distributed inside the block. In the second mode corresponding to other regions, a sophisticated smoothing filter, which is based on the frequency information around block boundaries, is used to reduce blocking artifacts adaptively without introducing undesired blur. Even though the proposed deblocking filter is quite simple, it improves both subjective and objective image quality for various image features.

Keywords: blocking artifacts, blocking noise, post-processing, deblocking filter.

1. INTRODUCTION

Traditional block-based video coders such as H.261, MPEG-1, and MPEG-2, suffer from annoying blocking artifacts when they are applied in low bit-rate coding because inter-block correlation is lost by block-based prediction, transform, and quantization. In order to overcome the blocking artifact problem, various non block-based coding schemes have been proposed. Among them, lapped orthogonal transform\(^1\) and embedded zerotree wavelet coding\(^2\) have been proposed as a non block-based texture coding method, and warping prediction\(^3\) and overlapped block motion compensation\(^4\) (OBMC) have been proposed as a non block-based prediction method. Though these activities have drawn a lot of interest, recent international standards regarding low bit-rate video coding tend to adopt the block-based coding scheme as the baseline method based on the overall consideration of performance, complexity, compatibility, market requirements, and so on. Therefore, as a post-processing method, deblocking filtering is regarded important due to improvement of visual quality in low bit-rate video coding. Even though the OBMC technique is adopted for deblocking in the recent low bit-rate video coder\(^5\), H.263, it can not prevent the blocking artifacts caused by block-based texture coding and the artifacts propagated from the previous reconstructed frame. Therefore, additional post-processing operations are also required for better image quality.

Many deblocking schemes have been proposed in still image coding such as JPEG under the assumption that blocking artifacts are always located at block boundaries. In video coding, however, blocking artifacts of the previous frame can be propagated to the current frame and can be located at any position in a block due to motion compensated prediction. Therefore, simple block boundary smoothing is not good enough to remove blocking artifacts appearing in video coding, so a more complicated scheme is needed. Iterative methods based on projection on convex set\(^6,7\) may be a candidate algorithm. However, it is not adequate for real time video coding due to its complexity. In contrast, smoothing skills using edge
information may be easier to adopt to video coding. Unlike still image, however, a video sequence consists of a set of still images having various image features. Hence these methods relying on edge detection performance suffer from various image features. This is because it is hard to find a proper threshold value for each frame to make a good edge map in real time and inaccurate edge detection may incur undesirable blur or regard blocking artifacts as edges.

In order to find an efficient deblocking filter, we investigate smoothing features in a video sequence in terms of the characteristics of HVS. According to these features across the block boundary, a deblocking filter with two separate modes is proposed. Each filter is a one-dimensional filter appropriate in real time application. For the mode decision independent of artifact position, we examine the existence of the offset in the region rather than the existence of the edge around the block boundary. In a very smooth region with small dc-like offset, strong smoothing is applied inside the block as well as on the block boundary. In the other region, a sophisticated smoothing operation is applied to reduce blocking artifacts without introducing undesirable blur by using the frequency information around the block boundary. It has been shown that the proposed deblocking filter improves not only subjective image quality but also objective image quality for various image features.

2. PROPOSED DEBLOCKING FILTER

A major operation of the deblocking filter is smoothing. But smoothing may introduce different effects according to local image characteristics across block boundaries. When we consider a smoothing operation to remove blocking artifacts in video coding, we find three interesting observations. First, the HVS is more sensitive to blocking artifacts in flat regions than in complex regions. Therefore, a strong smoothing filter is required on those regions. In complex regions, however, smoothing of a few pixels around the block boundary is enough to achieve a desired deblocking effect. Secondly, smoothing operations tend to introduce more undesirable blur in complex regions than in flat regions. Hence, adaptive smoothing to preserve image details is mandatory in complex regions. Thirdly, because of motion compensation, blocking artifacts are propagated and the propagated artifacts are more visible in flat regions than in complex regions. Therefore, smoothing in flat regions must cover the inside of a block as well as block boundaries.

From the observations above, it is found that the use of two separate filters depending on local image characteristics is preferable for effective deblocking. The filter for flat regions should provide a strong smoothing effect inside a block as well as on block boundaries. On the other hand, the filter for complex regions is required to work only on block boundaries. In the following discussions, filtering modes for flat regions and for complex regions will be called as the DC offset mode and the default mode, respectively. The filtering process consists of three major functional blocks, i.e., mode decision, filtering for the DC offset mode, and filtering for the default mode; and is based on one-dimensional filtering along the boundaries of a 8 x 8 block (see Fig. 1).

2.1. Mode decision

To select a proper mode between the DC offset mode and the default mode, local image characteristics in the region are to be examined. In the proposed scheme, we examine the flatness of the region by using the following measurement. When we consider a block boundary pixel array $\mathbf{V} = [v_1, v_2, \cdots, v_8]$,.

$$F(\mathbf{V}) = \sum_{i=0}^{8} \phi(v_i - v_{i+1}),$$

where

$$\phi(\Delta) = \begin{cases} 1, & |\Delta| \leq T_1 \\ 0, & \text{otherwise} \end{cases}$$

In Eq. (1), $T_1$ is set to a small value so that $F(\mathbf{V})$ may reflect the flatness of the local image across a block boundary. If $\mathbf{V}$ is dc-like, $F(\mathbf{V})$ has a big value larger than a certain threshold $T_2$. In that case, $\mathbf{V}$ is assigned to the DC offset mode, and strong smoothing is applied; otherwise, $\mathbf{V}$ is assigned to the default mode, and accurate and adaptive filtering is applied. Noticeable blocking artifacts may result from the concatenation of two flat regions with a small offset. In this case, the DC offset mode is selected because $F(\mathbf{V})$ still has a big value due to the two flat regions. It should be noticed that the value of $F(\mathbf{V})$ does not depend on the location of blocking artifacts, since it is obtained from difference values between neighboring pixels. This
2.2 Filtering in the DC offset mode

In this mode, we apply a 9-tab smoothing filter inside the block as well as on a block boundary. To prevent real edges in the filtering region from smoothing, however, filtering is not performed when the difference between the maximum value and minimum value of \( v \) is larger than a certain value, \( 2Q_P \). Here \( Q_P \) is the quantization parameter of the macroblock to which pixel \( v_3 \) belongs. This is because the offset related with blocking artifacts is usually a small value and is highly related with the quantization parameter. The filtering process of the 9-tab smoothing filter is as follows.

\[
v_n' = \frac{1}{16} \sum_{k=-4}^{4} b_k \cdot p_{s+nk}, \quad 1 \leq n \leq 8,
\]

where

\[
p_m = \begin{cases} p_0, & m < 1 \\ v_m, & 1 \leq m \leq 8 \\ p_9, & m > 8 \end{cases}, \quad p_0 = \begin{cases} v_0, & |v_1 - v_0| < Q_P \\ v_1, & \text{otherwise} \end{cases}, \quad p_9 = \begin{cases} v_9, & |v_9 - v_8| < Q_P \\ v_8, & \text{otherwise} \end{cases},
\]

and \( \{b_k : -4 \leq k \leq 4 \} = \{1,1,2,2,4,2,2,1,1\} \).

In Eq. (2), \( p_0 \) and \( p_9 \) are used as padded pixel values for 9-tab smoothing filter. In actual implementation, 9-tab filtering can be simplified by using shift and addition operations only.

2.3 Filtering in the default mode

As was mentioned, the default mode is applied on complex regions. In this mode, we limit the pixels to smooth to the two block boundary pixels, \( v_4 \) and \( v_5 \), and use the 4-point DCT as a frequency analysis tool to get the feature information of the pixel array. As shown in Fig. 1, if a 4-point pixel array \( S_1 \) is located across the block boundary, 4-point DCT basis vectors of \( S_1 \) have symmetric and anti-symmetric properties around the center of 4 points, or the block boundary (refer Fig. 3). We define \( a_{0,1}, a_{1,1}, a_{2,1}, \) and \( a_{3,1} \) as the 4-point DCT coefficients of \( S_1 \). Then it is noticed that the high frequency anti-symmetric component \( a_{3,1} \) is a major factor affecting the blocking artifact. This means that the proper adjustment of \( a_{3,1} \) is directly related to the reduction of block discontinuity in the spatial domain. Thus, in this mode, the magnitude of \( a_{3,1} \) is to be scaled down by using a scaling factor, \( \text{MIN}(|a_{3,0}|, |a_{3,1}|, |a_{3,2}|) / |a_{3,1}| \), whose value is between 0 and 1. The flatter neighboring regions are, the smaller the scaling factor is. By doing this, the block boundary is smoothened in smooth regions and is not affected in complex regions so that undesirable blurring can be prevented. If the magnitude of \( a_{3,1} \) is greater than a certain value (which is related with the quantization parameter), however, the filter is not applied to preserve the image details. The detailed filtering procedure in the default mode can be described as follows.

\[
a_{3,k} = \frac{c_1 \cdot v_{2k+1} - c_2 \cdot v_{2k+1} - c_2 \cdot v_{2k+3} - c_1 \cdot v_{2k+4}}{c_3}, \quad \text{where} \quad 0 \leq k \leq 2,
\]

and

\[
v'_n = v_n - d,
\]

\[
v'_5 = v_5 + d,
\]

where

\[
d = \text{CLIP}(\frac{c_2 \cdot (a'_{3,1} - a_{3,1})}{c_3}, 0, \frac{(v_4 - v_5)}{2}),
\]

\[
a'_{3,k} = \begin{cases} a_{3,k}, & \text{MIN}(|a_{3,k}|, |a_{3,1}|, |a_{3,2}|) \neq 0 \\ 0, & |a_{3,1}| = 0 \end{cases}
\]

Here, \( \text{CLIP}(x, p, q) \) clips \( x \) to a value between \( p \) and \( q \). And \( [c_1, -c_2, c_2, -c_1] / c_3 \) denotes a DCT kernel corresponding to the highest frequency component \( a_{3,k} \). If we describe 4-point DCT/IDCT of \( S_1 \) as a matrix form,
\[
\begin{bmatrix}
  a_{0,1} \\
  a_{1,1} \\
  a_{2,1} \\
  a_{3,1}
\end{bmatrix} =
\begin{bmatrix}
  k_0 & k_0 & k_0 & k_0 \\
  k_1 & k_1 & -k_1 & -k_1 \\
  k_2 & -k_2 & -k_2 & k_2 \\
  k_3 & -k_1 & k_1 & -k_3
\end{bmatrix}
\begin{bmatrix}
  v_3 \\
  v_4 \\
  v_5 \\
  v_6
\end{bmatrix}
\]  

(6)

\[
\begin{bmatrix}
  v_3 \\
  v_4 \\
  v_5 \\
  v_6
\end{bmatrix} =
\begin{bmatrix}
  k_0 & k_1 & k_2 & k_3 \\
  k_0 & k_1 & -k_2 & -k_1 \\
  k_0 & -k_1 & k_2 & k_1 \\
  k_0 & -k_1 & -k_2 & k_1
\end{bmatrix}
\begin{bmatrix}
  a_{0,1} \\
  a_{1,1} \\
  a_{2,1} \\
  a_{3,1}
\end{bmatrix}
\]  

(7)

where

\[
k_0 = 0.5,
\]

\[
k_1 = \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} = \frac{c_3}{c_1} = 0.6533,
\]

\[
k_2 = \frac{1}{\sqrt{2}} \cos \frac{3\pi}{4} = 0.5,
\]

and \(k_3 = \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} = \frac{c_2}{c_3} = 0.2706\).

It is noted that, using Eqs. (4) and (7), new values of \(v_4\) and \(v_5\) due to the change of \(a_{3,1}\) can be easily obtained without performing a full 4-point IDCT. If \(a_{3,1}\) increases by \(\varepsilon\), we can find that \(v_4' = v_4 - k_1 \cdot \varepsilon\) and \(v_5' = v_5 + k_1 \cdot \varepsilon\). We should also note that the scaling of \(a_{3,1}\) can be easily achieved without a dividing operation because the scaled value \(a_{3,1}'\) is given as \(\text{SIGN}(a_{3,1}) \times \text{MIN}(|a_{3,0}|, |a_{3,1}|, |a_{3,2}|)\). The clipping operation in Fig. 2 is used to make sure that the magnitude of the gradient at the boundary is reduced without change in direction. Then the values of \(v_4'\) and \(v_5'\) always reside between 0 and 255.

### 3. EXPERIMENTAL RESULTS

Simulation is performed by using the MPEG-4 VM 5.0 coder for low bit-rate DCT-based video compression. The advanced prediction mode with 8 x 8 block motion vectors and the OBMC mode are turned on, and a fixed quantization parameter is used. H.263 quantization method is adopted, and the motion search range is [-16.0, 15.5]. Various MPEG-4 test sequences are used for the bit rates of 10 Kbps, 24Kbps, 48 Kbps, 112 Kbps, and 1Mbps. Each test sequence has 300 frames, and only the first frame is coded as the intra frame. The components of the DCT kernel corresponding to \(a_{3,k}\), \(c_1\), \(c_2\), and \(c_3\), are approximated to 2, 5, and 8, respectively so that the filtering operation may require only integer multiplication and shift operations. And the threshold values of \(T_1 = 2\) and \(T_2 = 6\) are used. The proposed deblocking filter is applied for all the block boundaries along the horizontal edges first, and then along the vertical edges. If a pixel value is changed by the previous filtering operation, the updated pixel value is used for next filtering.

Fig. 4 shows deblocking results for the Foreman sequence. Even though the OBMC technique is used in the video coder, blocking artifacts produced in low bit-rate coding are still objectionable. In addition, we can see the drift of the location of blocking artifacts especially in the flat region. As shown in Fig. 4, the proposed deblocking filter reduces blocking artifacts substantially, and still preserves edge details quite well. PSNR is also improved for all the test sequences with 0.43 dB maximum in the Hall monitor sequence (see Table 1). These PSNR improvements demonstrate that the proposed filtering scheme is robust with respect to various image characteristics.
4. CONCLUSIONS

A new deblocking filter has been proposed to remove the blocking artifacts in reconstructed video frames. In order to minimize the computational complexity, the filter is based on one-dimensional filtering with two separate modes. The selection of a proper mode and a corresponding filtering process provide the improvements of the PSNR as well as the subjective quality by reducing blocking artifacts without sacrificing image details. Because of its simple architecture, the proposed deblocking filter can be easily implemented to real time applications.

5. REFERENCES

Figure 1: 8 x 8 block boundaries.

Figure 2: Basic structure of the proposed deblocking scheme.
Wave number

0  |  Symmetric
1  |  Anti-symmetric
2  |  Symmetric
3  |  Anti-symmetric

Block boundary

Figure 3: DCT basis vectors for 4 points of $S_1$.

Table 1: Performance evaluation of the proposed deblocking filter in terms of PSNR.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Sequence</th>
<th>QP</th>
<th>Bits</th>
<th>PSNR_Y [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Before</td>
<td>After</td>
</tr>
<tr>
<td>10Kbps</td>
<td>Hall monitor</td>
<td>17</td>
<td>96583</td>
<td>30.04</td>
</tr>
<tr>
<td>QCIF</td>
<td>Container ship</td>
<td>17</td>
<td>93556</td>
<td>29.21</td>
</tr>
<tr>
<td>7.5Hz</td>
<td>Mother &amp; daughter</td>
<td>15</td>
<td>95579</td>
<td>32.32</td>
</tr>
<tr>
<td>24Kbps</td>
<td>Hall monitor</td>
<td>9</td>
<td>236220</td>
<td>33.85</td>
</tr>
<tr>
<td>QCIF</td>
<td>Container ship</td>
<td>10</td>
<td>217480</td>
<td>32.36</td>
</tr>
<tr>
<td>10Hz</td>
<td>Mother &amp; daughter</td>
<td>8</td>
<td>231791</td>
<td>35.20</td>
</tr>
<tr>
<td>48Kbps</td>
<td>Foreman</td>
<td>13</td>
<td>478108</td>
<td>30.91</td>
</tr>
<tr>
<td>QCIF</td>
<td>Coast guard</td>
<td>14</td>
<td>446028</td>
<td>29.01</td>
</tr>
<tr>
<td>10Hz</td>
<td>Silent voice</td>
<td>7</td>
<td>484656</td>
<td>34.30</td>
</tr>
<tr>
<td>48Kbps</td>
<td>News</td>
<td>18</td>
<td>472973</td>
<td>31.20</td>
</tr>
<tr>
<td>CIF</td>
<td>Mother &amp; daughter</td>
<td>10</td>
<td>468027</td>
<td>36.06</td>
</tr>
<tr>
<td>7.5Hz</td>
<td>Hall monitor</td>
<td>12</td>
<td>458086</td>
<td>33.59</td>
</tr>
<tr>
<td>112Kbps</td>
<td>News</td>
<td>11</td>
<td>1139868</td>
<td>34.00</td>
</tr>
<tr>
<td>CIF</td>
<td>Foreman</td>
<td>30</td>
<td>1184538</td>
<td>28.25</td>
</tr>
<tr>
<td>15Hz</td>
<td>Coast guard</td>
<td>29</td>
<td>1172406</td>
<td>26.36</td>
</tr>
<tr>
<td>1Mbps</td>
<td>Stefan</td>
<td>13</td>
<td>9796735</td>
<td>29.00</td>
</tr>
<tr>
<td>SIF, 30Hz</td>
<td>Mobile &amp; Calendar</td>
<td>14</td>
<td>10259224</td>
<td>26.25</td>
</tr>
</tbody>
</table>
Figure 4: Deblocking results for Foreman sequence (CIF, 112kbps, 15Hz); (a), (c), and (e) are the 0th, 80th, and 160th reconstructed images, respectively; (b), (d), and (f) are deblocking results of (a), (c), and (e), respectively.