ADAPTIVE DERINGING AND MOSQUITO NOISE REDUCER

Salim Chebbo *, Philippe Durieux #, Béatrice Pesquet-Popescu *

# NXP Semiconductors, France
* Signal and Image Proc. Department, Telecom-ParisTech, France

ABSTRACT

In this paper, we propose an efficient algorithm to reduce the ringing and temporal noise artifacts in block based video coding. This algorithm operates adaptively on an 8x8 block basis. Temporally non-static blocks are spatially filtered, whereas temporally static blocks are filtered in both spatial and temporal directions. Experimental results show that the presented method considerably improves the subjective and objective quality of video sequences.

Index Terms— Post-Processing, deringing, mosquito noise, temporal noise.

1. INTRODUCTION

Video data are heavy forms of information and require a large bandwidth for delivery, which is considered as the greatest technical constraint. However, these data contain important redundancy in the form of high spatial and temporal correlation. Thus many coding techniques for removing such redundancy were developed, most of them utilizing block-based discrete cosine transformation (DCT), e.g. MPEG-1/2/4 and H.264. The quantization stage of these coding standards is the main contributor to bit rate reduction but also to impairment generation, such as blocking, ringing and mosquito noise.

The blocking artifact is one major issue of the block based coding technique; it appears as intensity discontinuities in the flat areas of the decoded images, and is a result of the coarse quantization of the frequency coefficients assembled in non-overlapping blocks. Numerous deblocking algorithms have been proposed for example in [1][2][3]. However, being more frequently addressed, deblocking algorithms are not the main focus of this paper.

The ringing artifacts are more noticeable along sharp edges as a rippling or ghost effects. In fact, in the DCT frequency domain, a sharp edge contains coefficients at all frequencies. Any change in the relative amplitudes of any of the coefficients will result in ripples around this sharp edge. Many post-processing methods have been proposed to reduce ringing artifacts. In [4], an edge map guided deringing filter is proposed; the classification procedure is based on the local pixels variance. In [5] a deringing system is proposed, based on two layers: classification and center weighted filter. The classification layer consists of a global and local classification of blocks and pixels, using the Line Spread Function (LSF). These two methods [4][5] use pixel variance based methods, which are consuming in terms of computational complexity. The MPEG-4 deringing filter [6] uses a simple classification method and an average filter to detect and smooth the ringing artifacts. However, this method is not very accurate and introduces blurring effects.

Mosquito noise is a time dependent video compression impairment, it can be noticed at the background uniform areas as a fluctuation of the luminance or chrominance levels, and as a moving ringing at textured areas in which the high frequency spatial details in video images having crisp edges are aliased intermittently. In the literature, many papers address the temporal noise reduction for image sequences [7][8]. However, fewer are the works treating the temporal noise in MPEG or ITU video sequences. In [9] a mosquito noise reducer is proposed; it is based on a motion detector to locate the motionless areas in the image, which are then filtered using a weighted (motion-controlled) median filter. The filtering is conducted in the DCT frequency domain, hence a DCT transform is required, as well as an inverse DCT. Furthermore, any change in any DCT coefficient for a given block will affect the entire block, and modify too many pixels, leading to other temporal artifacts. Thus, this filtering scheme is not accurate enough in removing mosquito noise. In [11], a recursive filtering scheme is presented. It applies an adaptive epsilon-filter recursively to each video frame in the spatial domain. This method tends to reduce the spatial fluctuation around a given pixel, contributing in the reduction of mosquito noise in the temporal domain. However, spatial filtering in not good enough in removing mosquito noise especially in uniform and background areas.

In this paper, we present new deringing and mosquito noise filters, which significantly reduce rippling and flickering artifacts. The deringing algorithm uses a simple block classification method to detect edge blocks, and a specific median filter to remove the detected ringing artifacts. The mosquito noise algorithm utilizes a motion
detector to locate the motionless areas, which are filtered with a temporal median filter applied directly on pixels value.

In the remainder of this paper, the proposed deringing and mosquito noise algorithms are described in Section 2. In Section 3, several simulation results are presented, and some conclusions are drawn in Section 4.

2. THE PROPOSED ALGORITHM

In our approach, we propose a spatial filter to reduce the ringing artifacts, as well a temporal algorithm to correct the mosquito noise and the temporal flickering between consecutive frames. These filters can operate separately or combined. In the later case, the deringing filter has to be applied first, and the spatially filtered pixels are used as input for the temporal filter.

2.1. Deringing filter

As already mentioned, ringing artifacts occur along edges; thus, in order to reduce the processing time, we only filter the blocks containing edges. However, attention should be given when filtering these blocks in order to preserve the details and avoid any blurring effect.

In order to locate blocks containing edges, a simple classification process is applied using the range of pixels variation inside the block, and comparing this range to a fixed threshold $T_1$. If the range is smaller than $T_1$, the block is considered as flat and no filtering is applied. Otherwise, the filtering is applied on each pixel inside the block. Thus, for a given pixel $C_5$, we consider a local filter mask within a 3x3 window around the current pixel (see Figure 3). Then a weight is assigned to each pixel position inside the filtering window. This weight can only take two values: 0 or 1, except for the center pixel $C_5$, as following:

$$ w(C_j)_{i=5} = \delta_i \quad (1) $$

where $\delta_i$ are set according to the following relation:

$$ \delta_i = \begin{cases} 1, & |C_5 - C_i| \leq TH \\ 0, & \text{otherwise} \end{cases} \quad (2) $$

The threshold $TH$ depends on the block range and the current quantization step $QP$. In fact, the decoding error resulting from the comparison between decoded frames and the original raw YUV frames (calculated on pixel basis difference), increases with the block range as shown in Fig. 4. This figure represents the absolute values of the decoding error of the first frame of the mobile sequence coded in Intra mode at $QP=10$. The average values of these errors, are generally smaller than $2^*QP$.

Therefore, the value of the threshold $TH$ is smaller than $2^*QP$ and is set empirically as a function of the block range and of the quantization step, as follow:

$$ TH = \text{clip} \left[ C + 2.\text{QP} \cdot \log_{10}(\frac{\text{range}}{4\text{QP}}) + 1, \ 2.\text{QP} \right] \quad (3) $$

where $C$ is a fixed constant equal to 5.

Using these weighting assignment rules, only the neighborhood pixels within a certain confidence interval around the target pixel are used by the median filter; therefore the filtering process adapts according to the local pixel variations. The filtering process uses a median filter, which is very effective in removing outliers, and shows better results than the conventional average filters for removing ringing artifacts.

2.2. Mosquito noise filter

As mentioned above, mosquito noise can appear at any block in the image, regardless if it is a stable or a moving block. In the latest case, mosquito noise is more annoying in textured blocks, containing spatial ringing artifacts. Therefore, deringing these blocks results in reducing the temporal flickering and the mosquito noise.
Figure 4. Decoding error variation with respect to the block range.

The mosquito noise filtering process uses three consecutive frames to temporarily filter the intermediate frame. It applies a median temporal filter on the three spatially co-located pixels in the consecutive frames. To avoid blurring moving areas, only static ones are filtered.

These areas are located by the motion detector (Figure 5), the motion detector proposed in [9] was preferred in our implementation. It demonstrates a good compromise between complexity and efficiency compared with other tested motion detectors. In our context, a small modification was done on the low pass filter (LPF) in order to decrease its complexity. This motion detector uses the difference between two consecutive frames to characterize motion and noise. This difference is filtered with the LPF to suppress the noise and the flickering, potentially interpreted as motion information. Finally, the absolute value of the LPF output is computed, in order to obtain the motion activity $M$ for each pixel. To decide if a given pixel is stable or not, a 3x3 window around this pixel is considered, and the global motion activity $GM$ inside this window is examined.

$$GM = \sum_{\text{pel} \in \text{3x3 window}} (M_{\text{pel}} > T_{\text{motion}})$$

where $T_{\text{motion}}$ is a fixed threshold, empirically set to 6.

If $GM$ is smaller then 2, then the given pixel is considered as stationary between the two frames in question.

The motion detector is applied between the current and previous frames, and then between the current and next frames, to form two motion maps. Only globally stationary pixels on these two motion map are considered as motionless. In that case temporal filtering is applied on these pixels using a median filter. A motionless pixel $C_{(i,j)}$ in the current frame is filtered with co-located pixels $P_{(i,j)}$ and $N_{(i,j)}$, shown in Figure 6.

3. SIMULATION RESULTS

The MPEG-4 VM coder is used to encode the test sequences, i.e. “Mobile”, “News” and “Hall”. These sequences contain 300 frames each; only the first frame is coded in intra mode, while the following ones are coded as P frames.

In Figure 7, the temporal filter is evaluated and compared to the MNR [9]. In this figure, R represents the ratio between the accumulated absolute pixel differences of two consecutives decompressed and filtered frames, of the mobile sequence coded at different bitrates. The higher is the ratio; the lower is the temporal variation of the filtered images. As shown in this figure, the ratio is always above 1 indicating that the temporal variation of the decoded sequence is greater than the temporal variation of the filtered one. Furthermore the ratio belonging to the proposed method is higher than the ratio of the MNR, thus a better flickering reduction.
Table 1. PSNR with the MPEG-4 and the proposed deringing filter.

<table>
<thead>
<tr>
<th>Bitrate</th>
<th>PSNR [dB] (MPEG-4 Filter)</th>
<th>PSNR [dB] (proposed filter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile (CIF, 30fps)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Mbs</td>
<td>26.85</td>
<td>26.90</td>
</tr>
<tr>
<td>2 Mbs</td>
<td>30.13</td>
<td>30.28</td>
</tr>
<tr>
<td>3 Mbs</td>
<td>32.41</td>
<td>32.67</td>
</tr>
<tr>
<td>NEWS (CIF, 30fps)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100 Kbs</td>
<td>33.97</td>
<td>34.16</td>
</tr>
<tr>
<td>150 Kbs</td>
<td>34.95</td>
<td>35.13</td>
</tr>
<tr>
<td>250 Kbs</td>
<td>36.01</td>
<td>36.28</td>
</tr>
<tr>
<td>Hall (CIF, 30fps)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150 Kbs</td>
<td>33.12</td>
<td>33.20</td>
</tr>
<tr>
<td>300 Kbs</td>
<td>36.24</td>
<td>36.43</td>
</tr>
<tr>
<td>500 Kbs</td>
<td>39.01</td>
<td>39.60</td>
</tr>
</tbody>
</table>

To evaluate the deringing algorithm, we used the MPEG-4 normative deringing filter [6] as a reference method for comparisons, and the PSNR values as a quality measure. Table 1 shows the overall PSNR gain of the proposed method and the MPEG-4 deringing filter. Figure 8 shows the relative bitrates gain of our deringing filter, with respect to the MPEG-4 deringing filter. This gain is calculated using the Bjontegaard metric described in [10]. According to Table 1 and Figure 8, we can remark that our algorithm is more efficient than the MPEG-4 deringing filter in terms of PSNR gain, especially at high bitrates, where the MPEG-4 filter over-smoothes the images, and induces blurring effects, without effectively removing the ringing artifacts. Subjective results are shown in Figure 9. In this figure, an area of the first frame of the “Mobile” sequence illustrates the result of the two algorithms, this intra frame is coded at QP = 20. While the MPEG-4 deringing filter is unable to completely remove the ringing artifacts, our algorithm can remove these artifacts without any collateral blurring effect.
4. CONCLUSION

In this paper, spatial and temporal filters for removing ringing and mosquito noise have been presented. The deringing algorithm uses a simple classification method to differentiate flat and edge blocks, which are filtered using a particular weighted median filter. The temporal filter utilizes a motion detector to locate motionless areas in the image, and then a median filter is applied on these areas. Simulation results showed that our algorithm can remove spatial and temporal oscillations, while preserving important image details.

REFERENCES


