A MULTI-FRAME POST-PROCESSING APPROACH TO IMPROVED DECODING OF H.264/AVC VIDEO

Xin Huang, Huiying Li, and Søren Forchhammer

COM Department, Technical University of Denmark, Building 343, Lyngby, DK-2800
Email:{xin,hli,sf}@com.dtu.dk

ABSTRACT

Video compression techniques may yield visually annoying artifacts for limited bitrate coding. In order to improve video quality, a multi-frame based motion compensated filtering algorithm is reported based on combining multiple pictures to form a single super-resolution picture and decimation to the desired format. The algorithm is applied to H.264/AVC decoded sequences and the processing involves a quality estimation based on picture type and local quantization value. Compared with directly decoding, the peak signal to noise ratio (PSNR) of the sequence obtained by the proposed algorithm is improved, and annoying ringing artifacts are effectively suppressed.

Index Terms— Artifacts reduction, motion compensated filtering, H.264/AVC

1. INTRODUCTION

H.264/AVC is the latest video compression standard. Due to its highly efficient performance, it will be used in future video storage and distribution applications. An in-loop de-blocking filter has already been addressed in H.264/AVC, therefore the most annoying artifact is ringing. Many postprocessing methods [1] have been developed based on the MPEG2 and H.263 standards. These methods can remove artifacts but also have a risk of over-smoothing details and sharpness, especially for sequences at medium coding bitrate. H.264/AVC has higher compression efficiency but it also loses many details. In order to remove ringing artifacts, enhance picture resolution, avoid over-smoothing details and preserve the sharpness after decoding, we modify and improve our previous work on MPEG2 [2] for application to H.264/AVC [3] decoded sequences.

The basic idea of the scheme is to apply an adaptive filter along motion trajectories utilizing an estimated quality of the pixel on each trajectory. The process can be divided into quality evaluation, motion compensated upsampling and de-ringing integrated decimation. First, the assumed quality of each pixel in the decoded sequence is estimated based on picture type and quantization step. In the second step, a super-resolution version (quadruple resolution default) of each directly decoded picture is constructed through temporal and spatial upsampling. Finally, a quality based decimation filter is designed to improve video quality and remove ringing artifacts. The motivation for a separate upsampling is an attempt to reduce single frame aliasing and trying to improve sharpness. The aim of this work mainly focuses on artifacts removal and video quality improvement, but by decreasing the decimation degree, higher resolution pictures can be also obtained.

The rest of the paper is organized as follows: In Section 2, a quality metric is designed to estimate each pixel's relative quality in the decoded sequence. A motion compensated upsampling algorithm to construct super-resolution pictures is described in Section 3. The de-ringing integrated decimation filter is described in Section 4. Test results are presented in Section 5.

2. QUALITY METRIC

The coded video sequence is mainly degraded by coarse quantization and inaccurate motion compensation. Macroblocks with different quantization parameter (QP) and prediction types (I, P or B) may have different distortion. Based on different picture types, we define a quality parameter q to reflect the mean squared error (MSE) for pixels in I, P and B pictures. With QP values and picture types available at the decoder, the quality parameter is calculated by $q = \sqrt{12 \times MSE}$, where MSE is determined by picture type and Q_{step} based on curves as shown in Fig. 1. The curves are obtained by measuring the MSE of the luminance components of H.264/AVC decoded sequences. QP determines the quantizer step size, Q_{step}. The results indicate that intra coded pictures (I) provide the best quality, and unidirectional prediction pictures (P) have better quality than bidirectional prediction pictures (B). We only use these training data to describe relative comparisons between different coding modes, it is not an absolute measure. All the settings and testing in later experiments are based on these curves. With this quality parameter, it is feasible to combine pixels with better assumed quality from neighboring pictures into the current picture, and prevent poor quality pixels from degrading better quality pixels.

The MSE caused by the quantization depends on the distribution of transform coefficients. This distribution is hard to estimate accurately due to varying image content. Some studies [4] have proposed to model transform coefficients with the Laplacian distribution, as opposed to the model in [2]. The distortion in pixel domain can be modeled as shown in Fig. 1 in comparison with the measured values.

3. MOTION COMPENSATED UPSAMPLING

Motion Compensated (MC) upsampling tries to form a superresolution picture (default has (V=4) times the resolution vertically and (H=4) times the resolution horizontally) by using the information from current picture and the N_f previous and subsequent pictures. Compared with a directly decoded picture, a MC upsampled higher resolution picture contains more information, which is helpful to remove artifacts and avoid over-smoothing details. MC upsampling starts with sub-pixel accuracy Motion Estimation (ME) to align pixels in the current picture with pixels in reference pictures. Pixels from reference pictures with fractional motion vector are assigned to the corresponding locations in the higher resolution pictures. Pixels from reference pictures with integer motion vectors are combined with decoded pixels in the current picture using a linear filter.

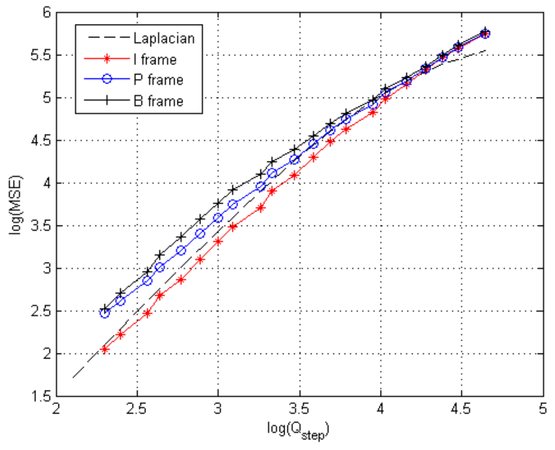


Fig. 1. MSE vs. Q_{step} measured on mobcal(CIF). Rate control is disabled, different QP values are chosen for the different points.

3.1. Motion Compensated Upsampling

In order to obtain reliable and homogeneous motion pixels x_r from reference pictures, a hierarchical block-based ME is utilized. The initial searching block size is set to be 16 $\times$ 16, followed by 4 sub-blocks (8 $\times$ 8). This final block size is our compromise between larger blocks for robustness and smaller blocks for accuracy. The motion vectors are obtained by searching in reference pictures for the best matching 8 $\times$ 8 block. They are denoted by $(m{+}\Delta m, n{+}\Delta n)$, where (m, n) is the integer part and $(\Delta m, \Delta n)$ is the fractional part of each motion vector. The fractional part is calculated by refining best matching block to sub-pixel accuracy using interpolation. The interpolated sub-pixels are generated by a six tap filter and then a linear filter as in H.264/AVC [3].

However, block-based motion estimation is not sufficient to guarantee that the best match pixels are in accordance with the true motion. It might introduce errors e.g., at occlusions in the motion compensation process. In order to reduce the risk of errors, we use a rejection criteria to evaluate for each pixel x_r whether it should be placed in the super-resolution picture. As in [2], the evaluation is based on intra-prediction [5]:

$$\hat{x}_{intra} = \begin{cases} min(a,b) & \text{if } c \geq max(a,b) \\ max(a,b) & \text{if } c \leq min(a,b) \\ a+b-c & \text{otherwise} \end{cases} \qquad (1)$$

where a, b and c denote the pixel at the left, top and top-left of pixel x_c respectively. We compare the intra-predicted pixels and best match pixels based on sum of absolute difference (SAD). The pixels x_r with larger SAD over an 8 $\times$ 8 block will be rejected.

Let (m_r, n_r) denote the absolute coordinates of the best matching pixel, x_r, with integer motion vectors in a reference picture. Let $(\Delta m, \Delta n)$ denote the relative displacement of interpolated pixels having minimum SAD within an 8 $\times$ 8 block. Its corresponding best match x_r with integer motion vectors is now perceived as an upsampled pixel at position $\big((m_r - m - \Delta m)V, (n_r - n - \Delta n)H\big)$. If more than one reference pixel map to the same position of the current super-resolution picture, the pixel is assigned to be the reference pixel with the best estimated quality (Fig. 1). If these reference pixels have equal quality parameter, the super-resolution pixel is assigned to be their weighted average.

Reference pixels with integer motion vectors ($\Delta m = 0, \Delta n = 0$) may also achieve the minimum SAD. These reference pixels are combined with the directly decoded pixels in the current picture on the same trajectory by using a linear filter. The linear filter is only applied on the condition that the reference pixels have better estimated quality parameters. Let x_c be a pixel in the current decoded picture and x_r a trajectory pixel from a reference picture with integer motion vector. We combine their values to obtain an estimated pixel by:

$$\hat{x} = h_r x_r + h_c x_c \qquad (2)$$

To minimize the expected MSE, the coefficients h_r and h_c could be estimated in a training session using original data by solving the Wiener-Hopf equations:

$$\begin{pmatrix} E\{X_r X_r\} & E\{X_r X_c\} \\ E\{X_c X_r\} & E\{X_c X_c\} \end{pmatrix} \begin{pmatrix} h_r \\ h_c \end{pmatrix} = \begin{pmatrix} E\{X X_r\} \\ E\{X X_c\} \end{pmatrix} \qquad (3)$$

where X_r and X_c represent stochastic variables of pixel values in reference picture and current picture respectively. X represents a stochastic variable of original pixel values at the same position in original resolution picture. In order to preserve the mean value, coefficients of this filter should be computed under the constraint $h_r + h_c = 1$. Given enough training data, the second-order mean values in (3) could be conditioned on quality of x_r and x_c. To reduce the training h_r and h_c are modeled as in [2]:

$$h_r = 1 - (1-\alpha)^{(q_c/q_r)^{\beta}} \qquad (4)$$

$$h_c = 1 - h_r \qquad (5)$$

This filter is fitted to optimal values of h_r (Fig. 2). The parameter α specifies the *a priori* weight that x_r should carry. The parameter β specifies how much the difference in qualities of x_r and x_c should influence the estimated pixel value. Equation (4) is monotonically increasing in the ratio q_c/q_r from 0 to 1 and it has the property that for $0 \leq \alpha \leq 1$, $\beta \geq 0$, $q_r, q_c \geq 0$ and $0 \leq h_r \leq 1$. Once this filter is applied to the pixels of the current and the reference picture, the estimated pixels in the super-resolution picture are assigned a new quality parameter value:

$$\hat{q} = h_r q_r + h_c q_c \qquad (6)$$

3.2. Interpolated Upsampling

After MC upsampling, an unfinished superresolution picture is formed. In order to complete the current super-resolution picture with irregular samples, we modify the cubic interpolation process with an irregular sample detection. Cubic spatial interpolation is based on rectangular lattice samples, which can supply true continuity among each segment and produce less jaggy edges. If there are no irregular samples in the nearest 4×4 pixel region, a normal cubic interpolation is implemented. Otherwise, a modified version is used:

$$x_{intp}(m',n') = \textstyle\sum_i \sum_j x_{re}(i,j) K_1 \beta^3(|m'-i|)\beta^3(|n'-j|)$$
$$+ \textstyle\sum_a \sum_b x_{ir}(a,b) K_2 \beta^3(|m'-a|)\beta^3(|n'-b|) \qquad (7)$$

where K_1 and K_2 are normalizing coefficients, $x_{re}(i,j)$ and $x_{ir}(a,b)$ represent samples at regular and irregular positions, respectively. $\beta^3(z)$ is a typical cubic convolution kernel [6]:

$$\beta^3(z) = \begin{cases} \frac{3}{2}|z|^3 - \frac{5}{2}|z|^2 + 1 & \text{if } 0 \leq |z| \leq 1 \\ -\frac{1}{2}|z|^3 + \frac{5}{2}|z|^2 - 4|z| + 2 & \text{if } 1 \leq |z| \leq 2 \\ 0 & \text{if } 2 \leq |z| \end{cases} \qquad (8)$$

4. DECIMATION

A super-resolution picture for each directly decoded picture is formed after upsampling. In order to reduce ringing artifacts and get the desired picture resolution, we propose a de-ringing integrated downsampling scheme applying a quality based spatial filter. Since ringing artifacts mainly appear in the vicinity of sharp edges, different types of decimation filters are operated in no-edge areas and edge areas, respectively. Canny's method is used for edge detection. In order to reduce the risk of blurring edges in the decimation process, both of the decimation filters are operated in a small 9 × 9 window.

4.1. No-edge Area Decimation

For the no-edge area, a two-dimensional spatial linear filter combined with adaptive quality weights is applied in the vicinity of each sample position (m_0, n_0) to obtain a lower resolution picture.

$$p_l(m_0', n_0') = \sum_{m,n} g(m, n, m_0, n_0) p_h(m, n) = \sum_{m,n} K g_v(|m - m_0|) g_h(|n - n_0|) w(m, n) p_h(m, n) \quad (9)$$

where $p_l(m_0', n_0')$ represents a downsampled pixel in the lower resolution picture, $p_h(m, n)$ represent the pixels which are adjacent to sample pixel $p_h(m_0, n_0)$ in the super-resolution picture. K is a normalizing factor ($\sum_g = 1$). g_v and g_h are 1-D symmetric filters in the vertical and horizontal direction, respectively. $w(m, n)$ is a weight function for each pixel based on its corresponding quality parameter described below. The 1-D symmetric filters g_v and g_h reflecting the spatial distance are defined by [2]:

$$g_2 = (\ldots, 0, a, 1, a, 0, \ldots) \quad (10)$$
$$g_4 = g_2 * g_2 = (\ldots, a^2, 2a, 1 + 2a^2, 2a, a^2, \ldots) \quad (11)$$
$$g_v = g_h = g_4 * g_4 \quad (12)$$

Furthermore, the value of a should be adaptive depending on local characteristics (smooth or texture). Therefore, we calculate a standard deviation σ of each downsampling sample $p_h(m_0, n_0)$ within a 9 × 9 window to obtain an adaptive control value:

$$a = \begin{cases} 1, & \text{if } \sigma \le 10 \\ 0.5, & \text{otherwise} \end{cases} \quad (13)$$

$w(m, n)$ is a weight function reflecting the qualities of different kinds of pixels. It depends on whether $p_h(m, n)$ and $p_h(m_0, n_0)$ are compensated upsampling pixels (p_{cu}) or interpolated upsampling pixels (p_{iu}). If both of them are compensated upsampling pixels, their quality parameters are used to determine the weight of $p_h(m, n)$. If one of them is obtained by interpolation, a constant weight value is assigned [2]:

$$w(m, n) = \begin{cases} \frac{w_0}{\gamma} \gamma^{\, q(m,n)/q(m_0,n_0)}, & \\ \quad p_h(m, n), p_h(m_0, n_0) \in p_{cu} & \\ 1, & p_h(m, n) \in p_{iu}, p_h(m_0, n_0) \in p_{cu} \\ w_0, & p_h(m, n) \in p_{cu}, p_h(m_0, n_0) \in p_{iu} \end{cases} \quad (14)$$

where the parameter w_0 (set to 6) specifies the *a priori* worth of a compensated upsampling (p_{cu}) pixel compared to an interpolated pixel (p_{iu}). The parameter γ (set to 0.3) is a global parameter reflecting the influence introduced by quality ratio.

4.2. Edge Area Decimation

For the edge areas, de-ringing integrated decimation filters are separately applied on each side of the edge boundary. Only those pixels, which are inside the decimation window and on the same side of the sample pixel $p_h(m_0, n_0)$, are used for this de-ringing filter [7]. Therefore, we define pixel sets $F^{(m_0,n_0)}$ as all the pixels used for the weighted de-ringing filter. The downsampled pixel value $p_l(m_0', n_0')$ is obtained by:

$$p_l(m_0', n_0') = \frac{\sum_{p_h(m,n) \in F^{(m_0,n_0)}} W_{(m,n)}^{m_0,n_0} p_h(m, n)}{\sum_{p_h(m,n) \in F^{(m_0,n_0)}} W_{(m,n)}^{m_0,n_0}} \quad (15)$$

where the weight factor $W_{(m,n)}^{m_0,n_0}$ is the product of local position distance factor $w_d(m, n)$, pixel difference factor $w_l(m, n)$ and quality factor $w(m, n)$. $w_d(m, n)$ and $w_l(m, n)$ are defined as:

$$w_d(m, n) = \begin{cases} \frac{1}{2 \times dis((m,n),(m_0,n_0))}, & \text{if } (m, n) \neq (m_0, n_0) \\ 1, & \text{otherwise} \end{cases} \quad (16)$$

$$w_l(m, n) = e^{-\frac{p_h(m,n) - p_h(m_0,n_0)}{Th}} \quad (17)$$

5. EXPERIMENTAL RESULTS

We used the H.264/AVC reference software JM9.3 [3] for experiments. Several *CIF* sequences (4:2:0) are chosen. They were encoded with different bitrates by enabling rate control. The GOP structure is defined as $(IBBP)_{12}$. In-loop de-blocking filter is on and single encoding reference frame is used. The parameter N_f is set to 5, α and β of the filter (4) are estimated using many frames of different sequences based on Equation (3), (See Fig. 2), the curves yield $\alpha = 0.15$ and $\beta = 0.7$.

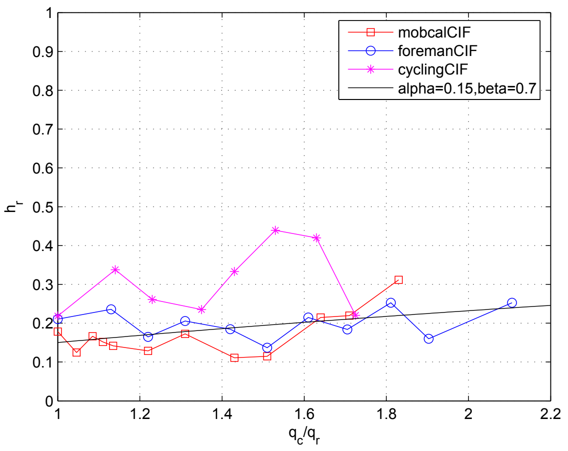
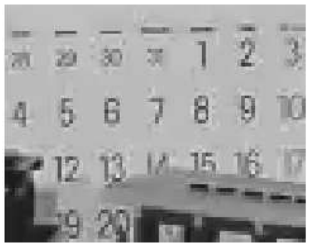
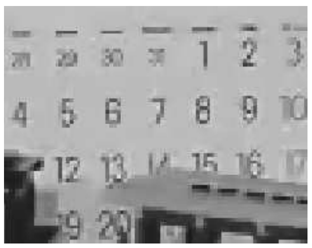
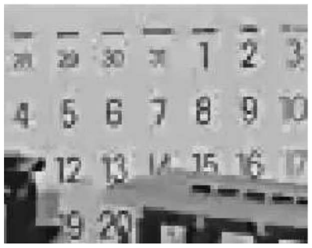


Fig. 2. Filter coefficient h_r as a function of q_c/q_r

Based on these settings, we implemented our algorithm on different directly decoded sequences. Fig. 3 is an example frame with our motion compensated filtering for *mobcal*. The average PSNR performances for the sequences *mobcal* and *foreman* are depicted in Figs. 4 and 5, respectively. From these figures we can clearly see that our algorithm is able to improve the average PSNR performance up to 0.3dB. The more interesting thing is that our algorithm can give improvement for the sequences at medium or relative high bitrate. It can be explained as: the magnitude of the improvements mainly depends on the relative quality of decoded picture compared to its surrounding pictures. Fig. 6 illustrates the PSNR improvement for each individual picture, it is noted that the algorithm improves all the pictures regardless of their directly decoded quality.


(a) Directly decoded frame


(b) With MC filter


(c) Sharpening decoded frame

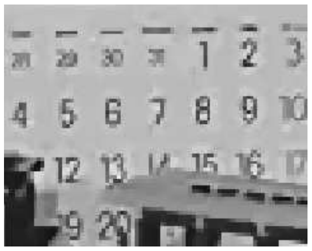

(d) Sharpening MC frame

Fig. 3. Visual comparison, mobcal(CIF) at 498kbit/s, frame 25

6. CONCLUSION

This paper presents a multi-frame approach to improve decoding quality of H.264/AVC sequences. From the experimental results, the average PSNR of the whole sequence is robustly improved especially for sequences at medium or relatively high bitrate. For individual pictures, all the pictures' quality is improved regardless of their directly decoded quality. Visually, ringing artifacts are reduced, sharp details and edge are well preserved.

7. REFERENCES

[1] Y.Nie, H.S.Kong, A.Vetro, and K. Barner, "Fast adaptive fuzzy post-filtering for coding artifacts removal in interlaced video," *Proc. ICASSP*, vol. 2, pp. 993–996, Mar. 2005.

[2] B.Martins and S.Forchhammer, "A unified approach to restoration, deinterlacing and resolution enhancement in decoding mpeg-2 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 803–811, Sept. 2002.

[3] MPEG AVC/H.264 video reference software JM9.3, *Available: http://iphome.hhi.de/suehring/tml/download/.*

[4] S.Smoot and L.Rowe, "Study of dct coefficient distributions," *Proc. SPIE*, pp. 303–311, Jan. 1996.

[5] JPEG-LS IS 14495-1, "Lossless and near-lossless coding of continuous tone still images," *ISO/IEC International Standard*, 1998.

[6] E.Meijering and M.Unser, "A note on cubic convolution interpolation," *IEEE Trans. Image Process.*, vol. 12, pp. 477–479, Apr. 2003.

[7] H.Li and S.Forchhammer, "Spatial postprocessing filtering for MPEG compressed video," in preparation.

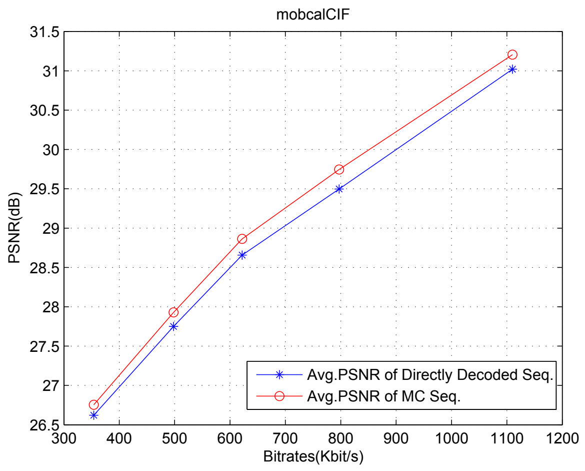


Fig. 4. PSNR performance of MC filter for mobcal(CIF)

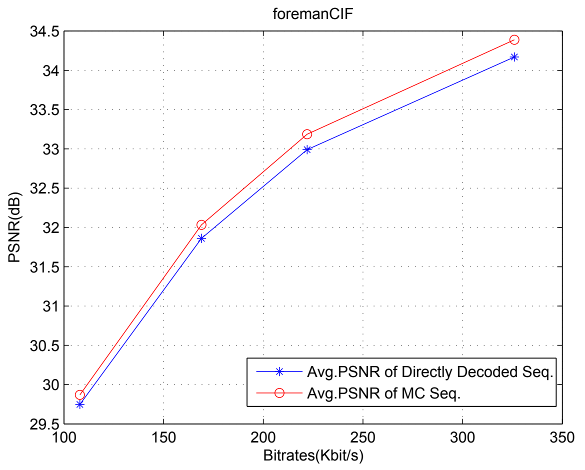


Fig. 5. PSNR performance of MC filter for foreman(CIF)

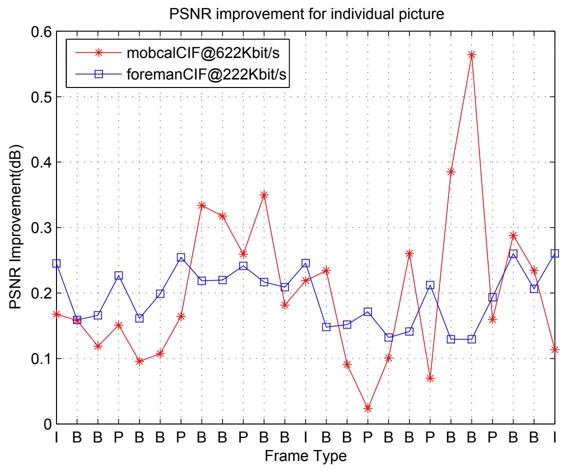


Fig. 6. PSNR improvement for individual pictures