A LOW-COMPLEXITY POSTPROCESSING ALGORITHM FOR BLOCK BASED VIDEO CODING

Ravikanth Maddikonda and Tushar K. Adhikary

Emuzed India Pvt. Ltd.

ABSTRACT

This paper presents a low-complexity algorithm to remove quantization effects in low bit rate video coding, such as blocking artifacts and ringing noise. The postprocessing algorithms presented in literature and MPEG-4 standard are of high computational complexity and apply low pass filtering twice to remove the two quantization effects mentioned above. The proposed algorithm uses the information contained in the compressed bit stream for processing. In this method, the deblocking and deringing operations are combined effectively and filtering is done only once to remove both the noises. This results in reduction of computational complexity. The efficiency of the proposed algorithm is compared with MPEG-4 postprocessing algorithm in terms of computational complexity and peak signal to noise ratio for several video sequences.

1. INTRODUCTION

The high volume audio-visual data associated with typical multimedia services call for efficient data compression schemes to facilitate transmission and storage applications. Most video coding standards like MPEG-4 [1] and H.263 [2] use block based Discrete Cosine Transform (DCT). An 8x8 block DCT packs the information into few coefficients by utilizing the spatial correlation of images. This block based coding introduces blocking artifacts at block boundaries as the blocks are coded independently. The blocking artifacts are mainly resulted from the quantization of DC and low frequency AC coefficients. The other artifacts that are introduced due to quantization of high frequency coefficients are ringing effect around the object edges (Gibbs's phenomenon) and corner outliers.

Many postprocessing algorithms like two dimensional signal adaptive filtering [3], iterative image recovery using theory of projection onto convex sets [4], spatio-temporal adaptive theory [5], Markov random fields [6], algorithm using spatial frequency and temporal information [7], etc. are proposed in literature. The MPEG-4 standard also suggests a postprocessing algorithm. The main draw back of these algorithms is high computational complexity. With increasing attraction on applications like video streaming and video on mobile handsets, where computational capacity and available power are limited, there is high demand for low complexity algorithms.

In this paper, a low-complexity video postprocessing algorithm is proposed which uses the spatial frequency and temporal information present in the coded bit stream. This algorithm removes the blocking artifacts at block boundaries and ringing noise at the object edges, which are usually more annoying among all the quantization effects in low bit rate video applications. The deblocking and deringing operations are combined effectively to decrease the computational complexity. The proposed algorithm needs approximately 15% of the computational complexity of MPEG-4 postprocessing algorithm.

This paper is organized in the following manner. Section 2 describes the proposed algorithm. Section 3 presents the simulation results and comparison of the proposed algorithm with MPEG-4 postprocessing algorithm in terms of computational complexity and peak signal to noise ratio (PSNR). Conclusions are given in section 4.

2. PROPOSED ALGORITHM

In MPEG-4 algorithm, pixel wise filtering decision is a computationally intensive step. In the proposed algorithm, the filtering decision is partially made on a 8x8 block basis using the DCT coefficients and motion vectors available from the bit stream.

2.1. Characterization of blocks

Each 8x8 block in a frame is divided into one of the two categories, based on the intensity changes. Blocks with low intensity variation are called smooth blocks and those with high intensity variation are called as complex blocks. Intensity variation is measured as explained below.

2.1.1 Intra coded block

Absolute sum of AC coefficients of an intra coded block is used as a measure for intensity variation of the block.

$$S = \Sigma^{7}_{u=0} \Sigma^{7}_{v=0} | F_{u,v} | \qquad (u,v) \neq (0,0) \qquad (1)$$

$F_{u,v}$ represents an AC coefficient. To check the effectiveness of the measure, scaled value of S is plotted on a block basis in Fig.1. The bright blocks matches with the blocks having high intensity variation.

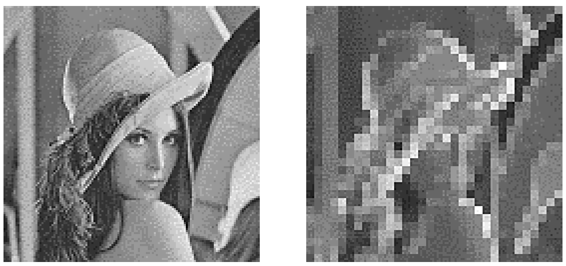

Fig.1: (a) Lena (b) Intensity variations

To decide whether a block is smooth or complex, its intensity variation measure (S) is compared with a threshold, T_1. If S is less than T_1 then current block is smooth. Otherwise current block is complex.

2.1.2 Inter coded block

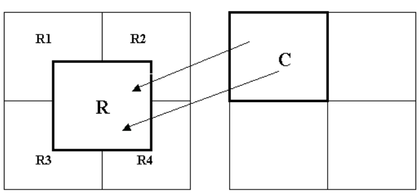

Fig.2: Reference Frame Predicted Frame

For characterization of an inter coded block, the motion vector information is used. In Fig.2, the block C is predicted as the region R in the reference frame. The overlapping blocks R1, R2, R3 and R4 with the region R are considered for decision making if overlapping region is at least two pixels width. If at least one of the overlapped blocks in the reference frame is a complex block, the block C is characterized as a complex block. If all overlapped blocks in reference frame are smooth blocks, then the block C is characterized using the criteria for the intra coded blocks.

2.2. Processing

It has been observed that blocking artifacts in smooth regions are usually more annoying compared to artifacts in complex regions. Hence, a strong filtering operation is required in smooth regions. Adjustment of pixels along the block boundary is done in the case of complex regions. This prevents excessive smoothing of edges and texture data of the video sequence.

In the case of smooth blocks there are no edges and hence only deblocking operation is performed. The block boundaries between two smooth blocks are processed using a one-dimensional filter perpendicular to the block boundary. In Fig.3, blocks A and B are smooth. P_0 to P_7 are pixel values on either side of the vertical block boundary. To prevent smoothing of real edges present along the block boundary, the processing is done only when absolute value of the difference between P_3 and P_4 is less than twice the quantization parameter (QP) of the block B. Pixels from P_0 to P_7 are processed using a 9 tap low pass filter. This operation is performed on all the pixels along the block boundary. Similar processing is done in the case of horizontal block boundary. In this case, the QP of the block that stays below is considered.

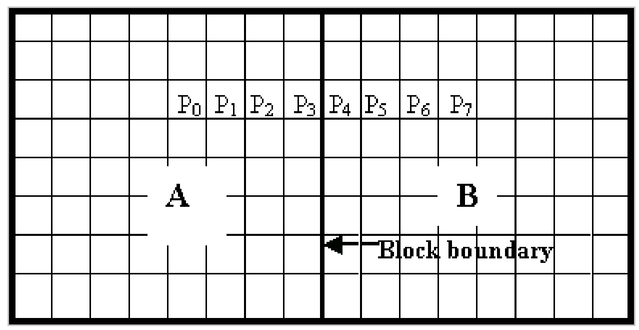

Fig.3: Processing of a vertical block boundary

The deringing operation is done on complex blocks. This deringing operation inherently removes the blocking artifacts.

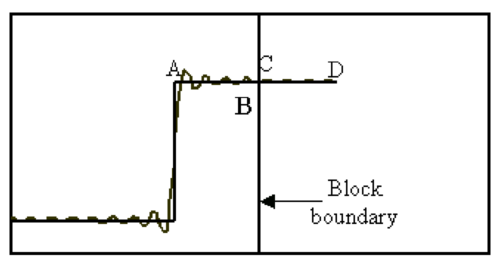

Fig.4: Ringing Noise

Ringing of edges is due to quantization of high frequency coefficients. In the Fig.4, ringing of a one-dimensional signal is shown. This can be assumed as a row of pixels. Most of the previous algorithms including MPEG-4 apply deringing operations, which are entirely block based. From Fig.4, one can observe that better results can be yielded by using pixel data from C to D, which is present on the other side of the block boundary, for processing of pixel data from A to B. The proposed algorithm makes use of this observation and uses data outside the block boundary, whenever desired.

A row of pixels is shown in Fig.5. One-dimensional horizontal gradient operator is applied on the row of pixels from P_2 to P_9 in order to find the edge pixels. If the absolute value of the difference between two consecutive pixels is greater than or equal to QP, then both the pixels are marked as edge pixels.

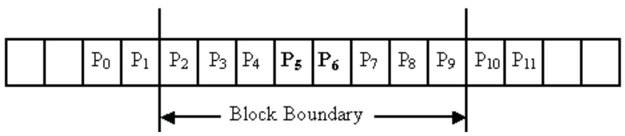

Fig.5: Array of pixels

For example, in Fig.5, pixels P_5 and P_6 are edge pixels. Now 1-D processing is done on pixels (excluding the edge pixels) as follows.

Step1: For processing the pixels P_2 to P_4, pixels outside the block boundary (P_0 & P_1) are used if the following constraints are satisfied.

(a) $| P_1 – P_2 | < (0.5QP)$

(b) P_1 lies in a smooth block.

Step 2: If the above conditions are satisfied, first the pixels P_2 to P_4 are to be processed. Processing is done on pixels starting from block boundary to the edge pixel. P_2 is processed using the pixels on the left side of it.

$$P_2' = (P_0 + P_1 + 2P_2) / 4.$$

Similarly,

$$P_3' = (P1 + P_2' + 2P_3) / 4,$$

$$P_4' = (P_2' + P3' + 2P_4) / 4.$$

Step 3: If the constraints (a) and (b) are not satisfied, filtering is done using the pixels within the block boundary only. Three tap filter with coefficients (1/4)[1 2 1] is used in processing. In the above example P_3 and P_4 are processed as follows:

$$P_3' = (P_2 + 2P_3 + P_4) / 4,$$

$$P_4' = (P_3' + 2P_4 + P_5) / 4.$$

Step 4: When the pixels between two edges in a row are to be processed, processing is done as explained in step 3.

Step 5: When no edge is found in the row of pixels, boundary pixels P1 and P2 are processed as following.

$d = (P_1 – P_2)$

if (| d | < 2QP) {

$P_1' = P_1 – d/4$

$P_2' = P_2 + d/4$ }.

Processing of pixels P_7, P_8 & P_9 is done in similar manner as explained above. Here, processing is done from block boundary to the edge i.e., P_9 to P_7 using the pixels on the right side of the pixel being processed.

Each block characterized as a complex block is processed in the above manner. First each row and then each column of the block is processed. Both the luminance and chrominance data of the frame are processed using the above procedure.

3. SIMULATION RESULTS

Simulations are performed using MPEG-4 simple profile video codec with fixed QP. Motion vector search range is [-16, 15.5] with four motion vectors per macroblock. Various QCIF test sequences with 15Hz frame rate coded at bit rates approximately 10, 24 and 48 Kbps are used. Each sequence is of 300 frames.

Simulations are carried out using the thresholds $T_1 = 10$ and $T_2 = 5$. Nine tap filter (1/16)[1 1 2 2 4 2 2 1 1] is used for filtering of block boundaries between smooth blocks. For subjective measure of quality, postprocessed frames using proposed algorithm and MPEG-4 algorithm are presented in Fig.6. Proposed algorithm reduces blocking artifacts and ringing noise substantially and the result of proposed algorithm is comparable with that of MPEG-4 algorithm.

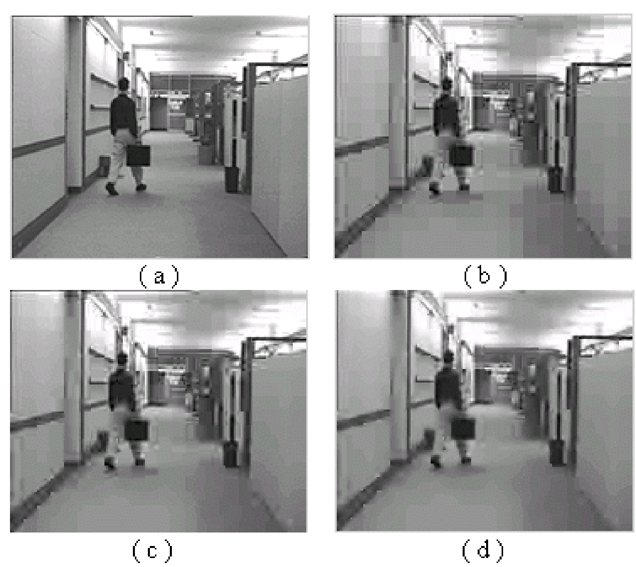

Fig.6: Postprocessing results of Hall Monitor (QCIF, 15Hz, QP = 25) 45^{th} frame (a) Original frame (b) Without postprocessing (c) Proposed filtering (d) MPEG-4 filtering

Table I: PSNR comparison for various sequences

Bit-rate condition	Fixed QP	Bit-rate (Kbps)	Sequence	Without postprocessing		MPEG-4 postprocessing		Proposed postprocessing	
				Intra PSNR (Y)	Avg. PSNR (Y)	Intra PSNR (Y)	Avg. PSNR (Y)	Intra PSNR (Y)	Avg. PSNR (Y)
10 Kbps	20	10.4	Container	29.20	28.88	29.44	28.91	29.43	28.97
	30	11.6	News	26.63	26.57	27.12	26.67	26.98	26.65
	14	9.6	Akiyo	33.04	32.83	33.69	33.03	33.37	32.96
24 Kbps	10	22.8	Container	33.76	32.80	33.92	32.88	33.92	32.89
	20	19.4	Students	28.40	28.30	28.78	28.32	28.66	28.35
	13	24.8	News	31.87	31.29	32.35	31.55	32.09	31.40
	9	25.6	Hall	34.78	34.27	35.22	34.56	35.06	34.42
48 Kbps	7	50.4	News	36.12	35.28	36.59	35.57	36.30	35.38

Table II: Comparison of computational complexity

Algorithm / Sequence		Hall QP = 9	Container QP = 10	News QP = 30	Akiyo QP = 14
MPEG-4	Control Instr.	1515057	1532696	1511539	1526000
	Additions	4005914	4077575	3902638	3989998
	Multiplications	284540	283567	281108	288473
	Total Complexity	**5805511**	**5893838**	**5695285**	**5804471**
Proposed algorithm	Control Instr.	188456	214473	172353	196651
	Additions	634798	623005	630634	613661
	Multiplications	66730	55124	62440	62488
	Total Complexity	**889984**	**892602**	**865427**	**872800**

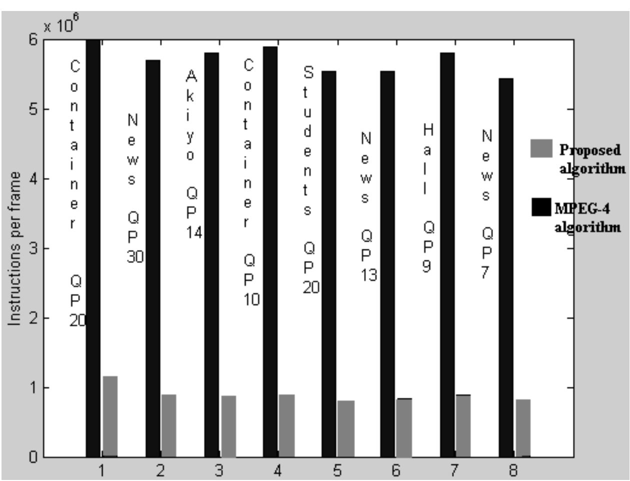

Fig.7: Comparison of total number of instructions per frame for MPEG-4 and the proposed algorithms

Peak signal to noise ratio (PSNR) is used as an objective measurement for comparing the results. Table I gives the PSNR measurements of luminance for several sequences at various bit rates. Table II gives the computation complexity statistics of the proposed algorithm and compares with MPEG-4 postprocessing algorithm. Computational complexity is measured in terms of number of additions, multiplications and control instructions required per frame averaged over 300 frames. On average the proposed algorithm requires only 15% of the computational complexity of the MPEG-4 postprocessing algorithm. Fig.7 presents the bar graph of computational complexity.

4. CONCLUSIONS

The proposed postprocessing algorithm effectively combines the deblocking and deringing operations and uses the information contained in the coded bit stream to reduce complexity. The proposed algorithm requires very low computational complexity and gives comparable performance to MPEG-4 postprocessing algorithm.

5. REFERENCES

[1]. “Information Technology – Generic Coding of Audio-Visual Objects – Part 2:Visual,” MPEG-4 standard, ISO/IEC/ JTC 1/SC29/WG 11 N 2688, March 1999.

[2]. “Video Coding for Low Bit Rate Communication,” H.263 Standard, ITU-T Recommendation H.263, February 1998.

[3]. Y. L. Lee, H. C. Kim, and H. W. Park, “Blocking Effect Reduction of JPEG Images by Signal Adaptive Filtering,” *IEEE Trans. on Image Processing*, Vol. 7, pp. 229-234, Feb. 1998.

[4]. Y. Yang, N. Galatsanos, and A. Katsaggelos, “Projection-based Spatially Adaptive Reconstruction of Block Transform Compressed Images,” *IEEE Trans on Image Processing*, Vol. 4, pp. 896-908, July 1995.

[5]. T. S. Liu and N. S. Jayant, “Adaptive Postprocessing Algorithm for Low Bit-Rate Video Signals,” *IEEE Trans. on Image Processing,* Vol. 4, pp. 1032-1035, July 1995.

[6]. Thomas Meier, King N. Ngan, and Gregory Crebbin, “Reduction of Blocking Artifacts in Image and Video Coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 3, pp. 490-500, April 1999

[7]. H Hyun Wook Park and Yung Lyul Lee, “A Postprocessing Method for Reducing Quantization Effects in Low Bit-Rate Moving Picture Coding,” *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, pp. 161-171, Feb. 1999.