Journal of Computer Science 8 (9): 1447-1454, 2012
ISSN 1549-3636

Performance, Analysis of Filtering Schedule Using Deblocking Filter for the Reduction of Block Artifacts from MPEQ Compressed Document Images

[1]M. Anto Bennet and [2]I. Jacob Raglend
[1]Department of Electronics and Communication Engineering,
Nandha Engineering College, Erode-638052, India
[2]Depatment of Electrical and Electronics Engineering,
Noorul Islam University, Nagercoil, India

Abstract: Problem statement: The visual effects of blocking artifacts can be reduce by using deblocking filter. Also with out smoothing the natural edges,the perceived quality of video sequence can be enchanced. This study propose a method to remove blocking artifacts in low bit-rate block based video coding. **Approach:** 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. **Results and Conclusion:** Deblocking filter improves the PSNR of about 0.1 dB for video encoded using MPEG-4 and H.264 without using its own in-loop deblocking filter. It has proven to be good in the reduction of the very annoying blocking artifacts caused by video compression.

Key words: H.264 compression, deblocking filters, video coding, discrete cosine transform, filtering, Peak Signal to Noise Ratio (PSNR)

INTRODUCTION

The main topics in the modern research of the field of multimedia are video compression and video coding. As the amount of information is huge on videos, video compression plays a vital role to transmit videos. By considering height, width, number of channels (usually three), color depth (usually minimum 8 bits) and sequence length (expressed in number of frames) are the parameters required for the calculation of amount of information contained in raw video. N bits = n Frames * H * W * n Channels * color Depth.

Many different Encoding strategies were proposed in literature. The main achievement is to obtain the representation of the sequence which is as tiny as possible. Block-based processing technique was used by the most of the video coding standards of the part. But, this kind of processing can cause visible blocking artifacts in the encoded video which are annoying for the user.Thus their effect should be mitigated as much as possible.The main source of blocking artifacts is the block-based integer Discrete Cosine Transform (DCT) in intra- and inter-frame prediction error coding.The second source of blocking artifacts is motion compensated prediction.The interpolated pixel data present in different reference frames can be used to generate motion compensated blocks.Discontinuities on the edge of copied block occur as there is almost never a perfect fit fot this data.Video compression leads to blocking which is the annoying visible artifacts. This problem is reduced by the small 4x4 transform size used in H.264/MPEG-4 AVC. Deblocking filter is an advantageous tool to maximize coding performance (List *et al*., 2003).

Existing system: The compression artifacts arise in JPEG, MPEG and H.264 by quantization of DCT coefficients. The quantization of low frequency coefficients results in blocky noise and the quantization of high frequency coefficients results in mosquito noise.

Corresponding Author: M. Anto Bennet, Department of Electronics and Communication Engineering,
Nandha Engineering College, Erode-638052, India

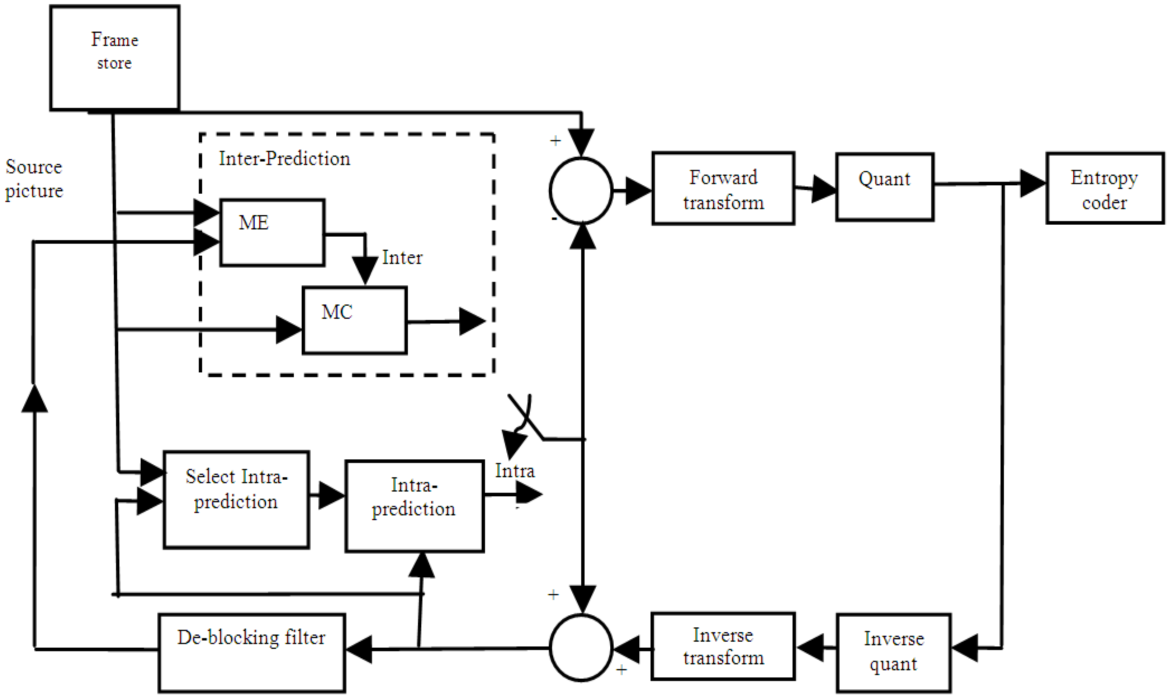


Fig. 1: H.264 encoding scheme

In data transmission if channel bandwidth is narrow, then data rate will be low and quantization level is dropped. As a result, the compression artifacts are increased. There are many methods which reduces the compression artifacts at the decoder. The Deblocking Edge Filter (DEF) method (Aujol *et al*., 2005; List *et al*., 2003; Chambolle, 2004) was reduces the blocky noise by using the noise removal technique called Projection On to Convex Sets (POCS), which is based on an iterative filtering (Dolar *et al*., 2009). An approach using Wavelet transform was proposed (Goto *et al*., 2008) for effective blocky noise reduction.

Another effective approach is Total Variation (TV) regularization (Kaup, 1998; Kim *et al*., 1998; Orchard *et al*., 1997). Which reduces the noise (Robertson and Stevenson, 2005). By utilizing this method, it is possible to reduce blocky noise. Alter *et al*. (2005) In this method based on a projected TV regularization especially targeting on DCT noise removal (Zakhoe, 1992). In this method, the total variation is reduced under DCT coefficient quantization constraint. This method will reduce a compression distortion, but minuteness of image is lost due to detetion of texture components. Thus the reduction of blocky noise and mosquito noise became insufficient at low bit rates (Rudin *et al*., 1992). The goal of a deblocking filter is to reduce blockiness and also preserving the sharpness of the content of the picture. To attain this, the large absolute difference between samples near a block edge is measured, which should be reduced (Choi and Kim, 2000). If magnitude of that difference is large then it cannot be described by coarseness of the quantization used in encoding in which the edge are more likely to reflect the actual behaviour of the source picture and should not be smoothed over.

Deblocking filter is implemented in the encoding loop shown in Fig. 1. In literature, two main approaches for deblocking can be found (List *et al*., 2003).

The first approach is post processing filter in which the deblocking operation is applied at each frame of the video after encoding/decoding procedure. The second approach is loop filtering in which filtering operation is carried out in encoding loop,which has the advantage of using filtered frames as reference frames leading to a higher quality prediction in motion compensation.

At the same time,the disadvantage is the application of identical filtering for the purpose to stay in synchronization with encoder. The deblocking effect can be improved by usage of post processing deblocking filter and loop filter. Filtering is not required in case of real edges of the video as application of filter will do blurring and it may result in difficulty in distinguishing real edges. so, an additional condition other than non-zero boundary strength (bs) is needed for effective use of deblocking filter. Block edge samples (p2, p1, p0, q0, q1, q2) are filtered only if they meet the following conditions Eq. 1 and 2:

$$bs>0 \tag{1}$$

$$|p0-q0| < \alpha \ \&\& \ |p1-p0| < \beta \ \&\& \ |q1-q0| \leq \beta \tag{2}$$

where, α and β are the thresholds defined in the standards Two types of filters are used: strong filter (5-tap filtering) and normal filter (4-tap filtering).

Filters are applied according to following:

```
if ((abs (p0-q0)< α && abs(p1-p0) < β && abs(q1-q0) ≤ β) && bS ==4)
apply strong filter;
else if ((ab s(p0-q0)< α && abs(p1-p0) < β && abs(q1-q0) ≤ β) && 0 < bS<4)
apply normal filter;
else
no filter;
```

MATERIALS AND METHODS

The processing of frames in the sequence is independently on each other.2-D filter (working both in horizontal and vertical directions) can be applied on each pixel. The decision map determines the process of filtering for the specific pixel .All the block-based video codecs (and the related blocking-artifacts), are covered by this method such that it filters 4×4 boundary of the frame. The algorithm can be applied on MPEG and on H.264 coded sequences. At first 8×8 block edges are scanned and the 4×4 block edges are processed subsequently.

F is the activity factor for the six-pixel vector P = {p0, p1, p2, p3, p4, p5}. G (QP) is a threshold and it is function of QP: stronger is the quantization and higher should be the value of G. F(p) represents the number of detected edges inside the vector P. T2 represents a fixed threshold max and min are the maximum and minimum valves of P0 and According to F(p) the vector P can divide the processing of the algorithm in three filtering mode types:

- Filtering decision step
- First filtering pass
- Second filtering pass

Decision modes: The two values XY is called filtering mode which assigns each pixel of the frame. X represents horizontal filtering mode and Y represents vertical filtering mode. X and Y take the values from the set {N, D, S}, where N means no filtering, D means default filtering and finally S means strong filtering.

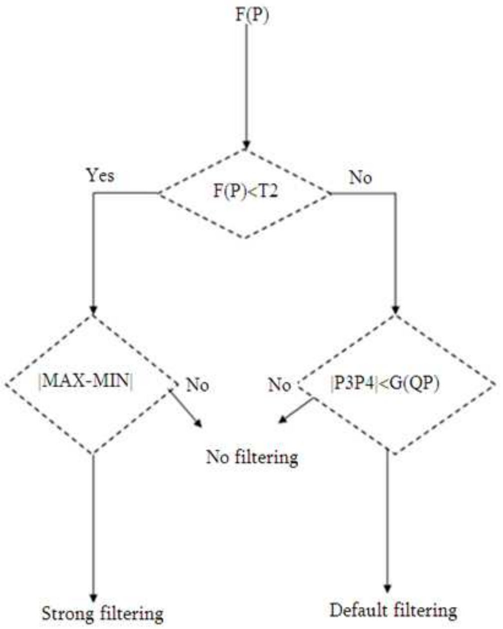


Fig. 2: Vector filtering classification

The outputs of this first step are two filtering mode decision matrices ,one for horizontal and other for the vertical direction (for each pixel the couple of value XY is defined). Filtering modes are calculated based on the variation of vertical and horizontal six-pixel vectors in its each 4x4 block boundary. First the activity of the six-pixel vector must be checked and if it is high, it means that there are variations in the set of pixels and they are to be filtered using strong filtering mode, otherwise bydefault filtering mode.

In second step, the final decision is made by estimating the activity of the pixel set is caused due to blocking artifacts or natural sharpness of the image itself. Statistics of the vector (based on neighbour pixel values) are computed if the pixel in the vector are candidates for strong or default filtering mode.If the value of the pixel has difference among them it cannot be explained due to the blocking artifacts effects. The decision map does not has filtering mode and the pixel value is altered. Otherwise, the decision will be strong or default filtering mode depending upon the first decision step in Fig. 2. Each set is characterized by their filtering modes estimated to its pixels. Pixels located around the horizontal block boundary of G2, have the filtering modes: {NN, ND, NS, SN, SD, SS}, as the horizontal default filter is not possible for G2 pixels.

Similarly the possible filtering modes of other sets are:

G1 : {NN, NS, ND, DN, DD, DS, SN, SD, SS}
G3 : {NN, NS, DN, DS, SN, SS}
G4 : {NN, NS, SN, SS}

Filtering: For the quality we introduce the motion of Filtering Window (FW) in Fig. 3. To designate a 6×6 pixel box centred at the intersection of four 4×4 pixel blocks. In Fig. 4a FW is first placed at the upper left corner of MB and shifted based on scanning order.In Fig. 4b 8×8 block edges are filtered by the remaining 4×4 block edges. This process considers the blocking artifacts in video coded with 8×8 block DCT, like MPEG-2.

To avoid multi-filtering, 16 pixels in FW are filtered by 2-D filter.To reduce the complexity of nonseparable filters , MB is processed in two passes. In Fig. 5b, the dark yellow pixels represent formerly filtered pixel from the upper and left MBs, while the light yellow pixels are the filtered pixels after running the first pass on current MB. The white regions represent the remaining unfiltered pixels. These pixels are filtered later in a second pass, with a simplified set of filters as defined below.

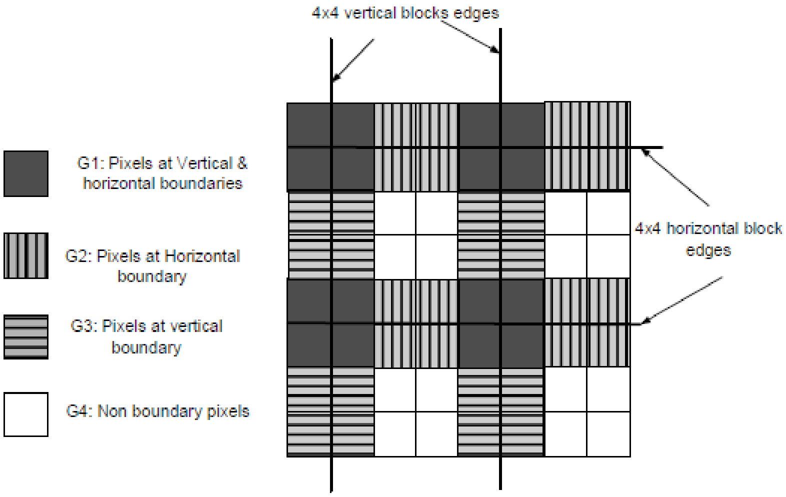
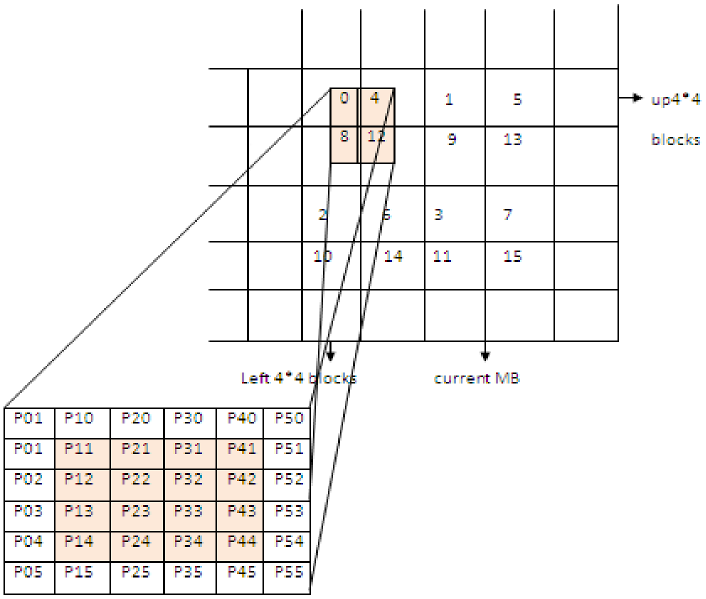
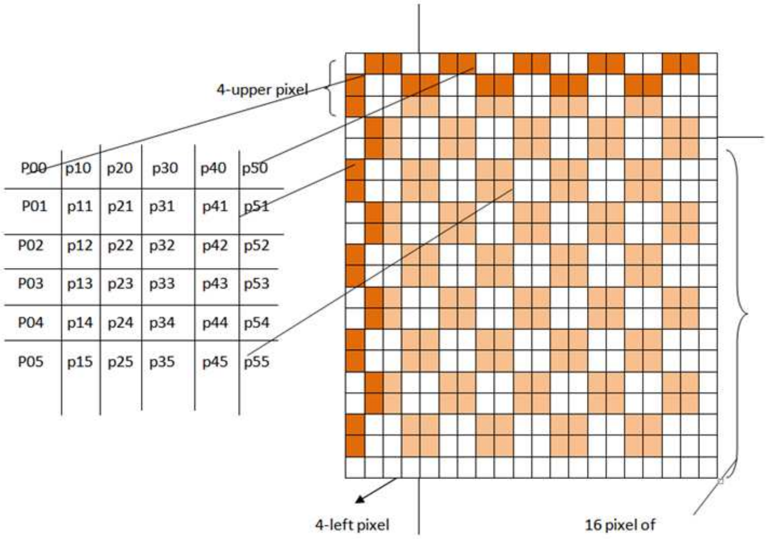

4x4 vertical blocks edges
G1: Pixels at Vertical & horizontal boundaries
G2: Pixels at Horizontal boundary
G3: Pixels at vertical boundary
G4: Non boundary pixels
4x4 horizontal block edges

Fig. 3: Pixel groups according to their filtering mode



P01	P10	P20	P30	P40	P50
P01	P11	P21	P31	P41	P51
P02	P12	P22	P32	P42	P52
P03	P13	P23	P33	P43	P53
P04	P14	P24	P34	P44	P54
P05	P15	P25	P35	P45	P55

Fig. 4: (a) Filtering window (b) Filtering window position Order throughout the MB

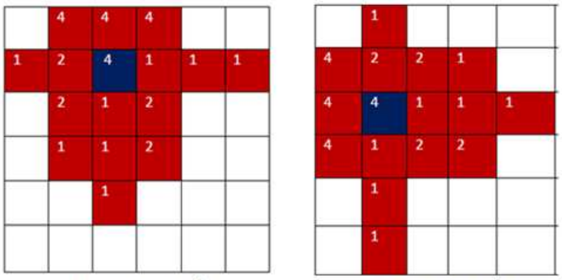


P00	p10	p20	p30	p40	p50
P01	p11	p21	p31	p41	p51
P02	p12	p22	p32	p42	p52
P03	p13	p23	p33	p43	p53
P04	p14	p24	p34	p44	p54
P05	p15	p25	p35	p45	p55

(a) Filtering window

(b) 16 pixel of the current MB

Fig. 5: (a) Pixel filtered in step 1 inside the filtering window (b) inside the MB



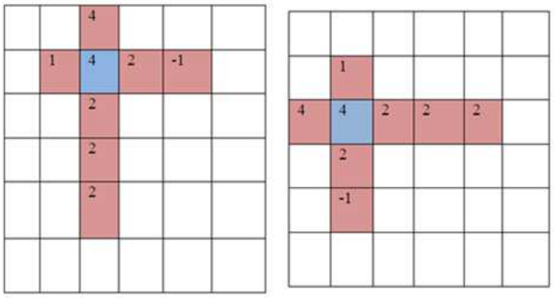
	4	4	4		
1	2	4	1	1	1
	2	1	2		
	1	1	2		
		1			

P22 in SS mode

	1			
4	2	2	1	
4	4	1	1	1
4	1	2	2	
	1			
	1			

P12 in SS mode

Fig. 6: Weighting coefficients for SS Mode



		4			
	1	4	2	-1	
		2			
		2			
		2			

P21 in SS mode

	1				
4	4	2	2	2	
	2				
	-1				

P12 in SS mode

Fig. 7: Weighting coefficients for DS/SD Modes

First filtering pass: In first pass eight pixels FW(p12, p13, p21, p24, p31, p34, p42, p43) are filtered. in Fig. 5a. Filtering modes for G2 and G3 pixels are:

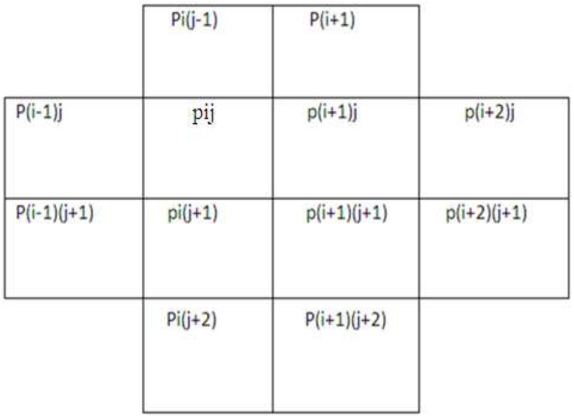
{NN, ND, NS, SN, SD, SS} +
{NN, NS, DN, DS, SN, SS}

For filtering modes with an N (no filter) in any direction (ND, NS, DN, SN), only one dimensional filters are required. For instance, ND and DN modes apply a 1-D default filter on the target pixel in vertical and horizontal direction respectively. The ND mode can be assigned to the pixels belonging to G2, (p12, p13, p42, p43,). In this case, the filter is applied vertically on the target pixel.

As in the DN filtering mode of the pixels belonging to G3 (p21, p24, p31, p34), the filtered pixel values are computed symmetrically to the ND filtering.

In the cases like where the filtering mode belongs to {DS, SD, SS}, a 2-D filtering is applied on the desired pixel. The introduced 2-D filters are the simplified versions from the combination of the horizontal and vertical 1-D filters.In order to preserve a small amount of computations, the weighted matrix of the 2-D filter is simplified. some coefficients which are having a small weight are cut and others are rounded, while preserving similar filter characteristics. Figure 6 and 7 show the simplified 2-D filters used to process p21 and p12 in SS, DS and SD modes. Other G2 and G3 pixels are filtered in the same way.

Second filtering pass: At the end of the first pass, pixels belonging to G2 and G3 are filtered throughout the MB. During this second pass, we filter the remaining pixels which are belonging to G1 and G4, by applying the appropriate filter.



	Pi(j-1)	P(i+1)	
P(i-1)j	pij	p(i+1)j	p(i+2)j
P(i-1)(j+1)	pi(j+1)	p(i+1)(j+1)	p(i+2)(j+1)
	Pi(j+2)	P(i+1)(j+2)	

Fig. 8: Unfiltered pixel positions after the first pass

This filter is done according to the pre assigned filtering mode and also by using the updated pixels from the first pass. In Fig. 8, pi(j-1),p(i+1),p(i+2)j,p(i+2)(j+1),p(i+1)(j+2),P(j+2),p(i-1)(j+1), p(i-1)j pixels represent pixels filtered during the first pass, while white pixels are updated according to their assigned filtering mode as follows:

DN:
Pij=(p(i-1)j+5pij+3p(i+1)j-p(i+2)j)>>3NS:
Pij=2pi(j-1)+pi(j+2)+pij>>2
SD:
Pij=(6p(i-j)+4p(i+2)j+4pij+2pij(j+1)+pi(j-1)-pi(j+2)>>4
DD:
Pij=(8pij+4p(i+1)j+4pi(j+1)+p(i-1)j+pi(j-1)-pi(j+2)-p(i+2j)>>4
SS:
Pij=(2pi(j-1)+2p(i-1)j+2pij+pi(j+2)+p(i+2)j)>>3

For symmetric filtering modes, the filtered values of pij are simply computed in a symmetric manner. quality. Once the decision map is ready, the filtering can be started. A 6×6 filtering window is considered and centered at the intersection of four 4×4 pixel blocks. The filtering window is first filter 8×8 block edges and then the 4×4 block. For each filtering pixels shown in Fig. 8 are filtered ,once the first filtering pass is completed the second filtering pass is applied to remaining pixels.

Experimental results: The sequences have been encoded starting from a raw YUV file. A frame of the sequence "Foreman" from the uncompressed YUV sequence in Fig. 9. A frame of the sequence "Foreman" from theMPEG-4 compressed sequence in Fig. 10.

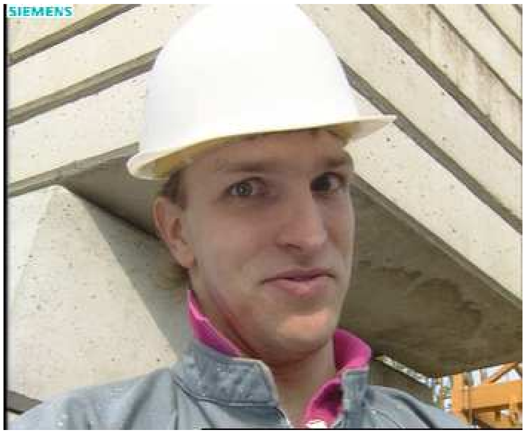


Fig. 9: A frame of the sequence Foreman from the uncompressed YUV sequence

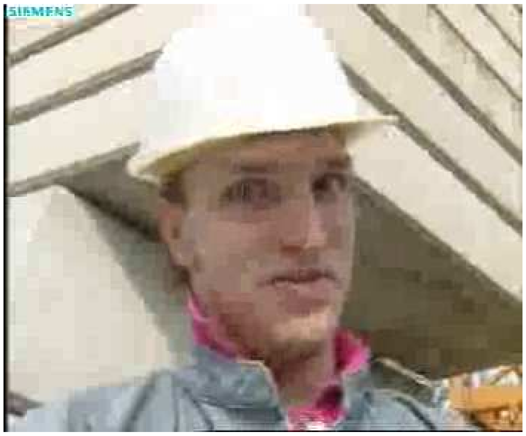


Fig. 10: A frame of the sequence Foreman from the MPEG-4 compressed sequence

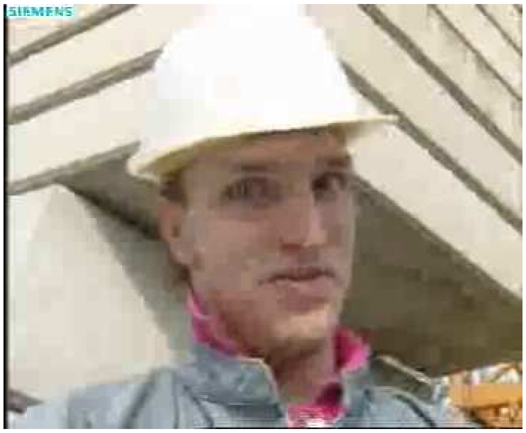


Fig. 11: The same frame, after applying the deblocking filter

Table 1: Performance of deblocking filter

Video sequence	Bitrate	PSNR compressed	PSNR deblocked	PSNR gain	Encoder
Foreman	168 kbps	27.68	27.83	0.15	MPEG-4
News	179 kbps	31.84	31.91	0.07	MPEG-4
Coastguard	211 kbps	25.62	25.69	0.07	MPEG-4
Foreman	63 kbps	26.90	27.06	0.16	H.264 (without in-loop deblocking filter)
News	47 kbps	29.73	29.87	0.14	H.264 (without in-loop deblocking filter)
Coastguard	144 kbps	27.25	27.35	0.10	H.264 (without in-loop deblocking filter)
Foreman	63 kbps	27.57	27.60	0.03	H.264 (with in-loop deblocking filter)
News	38 kbps	28.83	28.86	0.03	H.264 (with in-loop deblocking filter)

The same frame, after applying the deblocking filter in Fig. 11 Working of the deblocking filter has been tested in three different scenarios:

- Videos are encoded using MPEG-4 codec
- Videos have been encoded with H.264 codec, with the in-loop deblocking filter turned off
- Videos have been encoded with the H.264 codec, with the in-loop deblocking filter turned on

The encoding has been tuned in order to generate clear blocking artifacts. In H.264 encoding block is inevident and in remaining it is visible.

The bit rate is directly connected to amount of compression used (the lower the bit rate, the worse the video and, generally speaking, the more visible the encoding artifacts). The PSNR (Peak Signal-to-Noise Ratio) is an objective measure of video quality. The PSNR for two images can be computed as follows:

$$MSE=1/m\ n \sum\sum [i(i ,j)-k(i, j)]2$$

MSE is called Mean Square Error. The PSNR is defined as:

$$PSNR=10.Log\ 10(MAX21/MSE)$$

where, MAX_I is the maximum value of the image, which is for example 255 for 8-bit images.

RESULTS

The PSNR value reported here is the average PSNR value calculated for each frame. To compute this value, the compressed sequences (after encoding) and the deblocked sequences (after our filter) have been tested against the uncompressed YUV sequence.

DISCUSSION

PSNR does not take in to account of the HVS (Human Visual System model) and is not reliable measure of the objective quality of processed image to original image not only the small improvement in PSNR value,the deblocking filter has proved to be reduced of annoying blocking artifacts by video compression..It can be clearly seen that the blocks are smoothed out human eye perceives a better quality of the deblocked frame.

CONCLUSION

From the results reported in the Table 1, it is proved that deblocking filter improves thePSNR to 0.1 dB for video encoder using MPEG-4 and H.264 without using its own in-loop deblocking filter. Video sequences encoded by H.264 with its own deblocking filter is enabled and the improvement is lower (0.02-0.03 dB). This is because most of the blocking artifacts is already removed by H.264 in-loop deblocking filter and the video sequences cannot be further improved.

REFERENCES

Alter, F., S. Durand and J. Froment, 2005. Adapted total variation for artifact free decompression of JPEG images. J. Math. Imag. Vis., 23: 199-211. DOI: 10.1007/s10851-005-6467-9

Aujol, J.F., G. Aubert, L. Blanc-Feraud and A. Chambolle, 2005. Image decomposition into a bounded variation component and an oscillating component. J. Math. Imag. Vis., 22: 71-88.

Chambolle, A., 2004. An algorithm for total variation minimization and applications. J. Math. Imag. Vision, 20: 89-97. DOI: 10.1023/B:JMIV.0000011325.36760.1e

Choi, H. and T. Kim, 2000. Blocking-artifact reduction in block-coded images using wavelet-based subband decomposition. IEEE Trans. Circ. Syst. Video Technol., 10: 801-805. DOI: 10.1109/76.856457

Dolar, C., M.M. Richter and H Schroder, 2009. Total variation regularization filtering for video signal processing. Proceedings of the IEEE 3rd International Symposium on Consumer Electronics, May 25-28, IEEE Xplore Press, Kyoto, pp: 1-5. DOI: 10.1109/ISCE.2009.5156858

Goto, T., T. Yamazaki and M. Sakurai, 2008. A study on improving image quality of highly compressed moving pictures. Proceedings of the International Symposium on Information Theory and its Applications, Dec. 7-10, IEEE Xplore Press, Auckland, pp: 1-6. DOI: 10.1109/ISITA.2008.4895629

Kaup, A., 1998. Reduction of ringing noise in transform image coding using simple adaptive filter. IEEE Elect. Lett., 34: 2110-2110. DOI: 10.1049/el:19981519

Kim, H.C., Y.L. Lee and H.W. Park, 1998. Blocking effect reduction of JPEG images by signal adaptive filtering. IEEE Trans. Image Process., 7: 229-237. DOI: 10.1109/83.661000

List, P., A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, 2003. Adaptive deblocking filter. IEEE Trans. Circ. Syst. Video Technol., 13: 614-619. DOI: 10.1109/TCSVT.2003.815175

Orchard, M.T, Z. Xiong and Y. Zhang, 1997. A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations. IEEE Trans. Circ. Syst. Video Technol., 7: 433-437. DOI: 10.1109/76.564123

Robertson, M.A. and R.L. Stevenson, 2005. DCT quantization noise in compressed images. IEEE Trans. Circ. Syst. Video Technol., 15: 27-38. DOI: 10.1109/TCSVT.2004.839995(410) 1

Rudin, L.I., S. Osher and E. Fatemi, 1992. Nonlinear total variation based noise removal algorithms. Phys. D., 60: 259-268. DOI: 10.1016/0167-2789(92)90242-F

Zakhoe., A.,1992. Iterative Procedures for reduction of blocking effects in transform image coding. IEEE Trans. Circ. Syst. Video Technol., 2: 91-95. DOI: 10.1109/76.134377