Reducing the blocking Effect in Image and Video Coding by Three Modes of Adaptive Filtering or Interpolating[1]

Issam Abboud[2]

Abstract

Most image and video coding techniques are based on the method of Discrete Cosine Transform (DCT) after partitioning the image into square blocks. This method is called block DCT (BDCT). It is considered as one of the best methods of image and video compression. However, in cases of high compression ratios, the BDCT method results in the block boundaries being visible in the reconstructed image (after decoding). This distortion is called the blocking effect or blocking artifacts.

In this research we illustrate the blocking effect and its causes. Then, we propose two methods for reducing the blocking effect using a 3-mode adaptive filtering and a 3-mode adaptive interpolation. We realize the two proposed methods by two suitable algorithms, seeking to effectively and relatively simply reduce the blocking artifacts without causing degradation of the other characteristics of the image quality (taking into account some specific requirements of video coding). We present the results of applying both the algorithms. The results show the effectiveness of the two proposed methods.

[1] For the paper in Arabic see pages (43-44).

[2] Dept of Electronic and Communications Engineering- Faculty of Mechanical and Electrical Engineering- Damascus University.

1-Introduction

In most image and video coding techniques, a still image or each frame of a video sequence, is partitioned into square blocks; then each block is encoded independently using the Discrete Cosine Transform (DCT). This method is called block direct cosine transform (BDCT) [1, 2]. The size of each block in most applications is 8 x 8 pixels, but it may be 4 x 4 or 16 x 16 pixels in some cases [3, 4 ,23]. BDCT is employed in most JPEG and MPEG standards, and the video coding standards of ITU (H.261, H.263) [3]. The performance of BDCT approaches that of the optimal Karhumen-Loeve transform (KLT) [1, 2, 5]. However, in cases of high compression ratios, such as low-bit-rate applications in video communications, BDCT results in unpleasant visual distortions in the reconstructed image (after decoding). The most obvious distortion caused by BDCT is the blocking effect (blocking artifact), which appears as discontinuities along the block boundaries. Fig.1 illustrates the blocking effect for a part of the well-known "Lena" image (512 x 512 pixels), decoded at 0.15 bpp (bits per pixel) [6]. The small square blocks in Fig.1 are seen all over the image because of the high compression ratio. In general, the blocking effect is observable as vertical and horizontal false edges periodically appearing in the image, especially in the “smooth” regions [2].

Fig.1- The blocking artifacts in “Lena” Image.

2-The causes of blocking artifacts

The blocking effect in a BDCT-coded still image such as a JPEG image, or a frame of a video sequence in a MPEG sequence, is due to the independent quantization of DCT coefficients in neighboring blocks [2]. In the BDCT quantization procedure, the encoder divides the DCT coefficients by the corresponding dividing factors in the quantization table and encodes the resulting coefficient value [2, 3, 7, 22]. The decoder de-quantize each coefficient be setting the value to the midpoint of the corresponding quantization interval. This method is optimal in the minimum mean- square- error (MSE) sense, for the case of a uniform DCT coefficients distribution. However, the accuracy loss of DCT coefficients in the process of independent quantization of each block, results in the blocking artifacts, which may be noticeable particularly in the “smooth” or slowly varying parts of an image [8]. For example, noticeable blocking artifacts may result from the concatenation of two “flat” regions with a small offset (of the luminance value) [9].

Fig.2 shows the comparison of pixel values (luminance or brightness values) for a part of a row (scan line) of the original 512 x 512 Lena image, and the same image, compressed by JPEG at 0.25 bpp [5]. The pixel values in Fig.2 are shown for pixels at the positions (437, 425) to (437, 472).

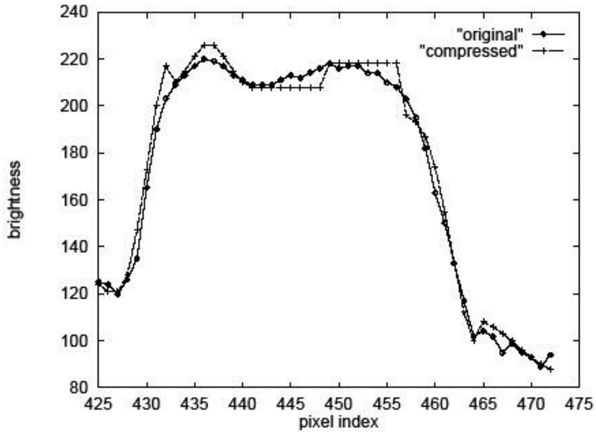


Fig 2- A comparison of pixel values of original and compressed images.

Because of the independent quantization of each block, a smooth change of luminance (brightness) across a border can result in a step in the decoded image, if neighboring samples fall into different quantization intervals [10]. The blocking effect observed around the pixel index 449 in Fig.2 corresponds to Fig.3 [5].

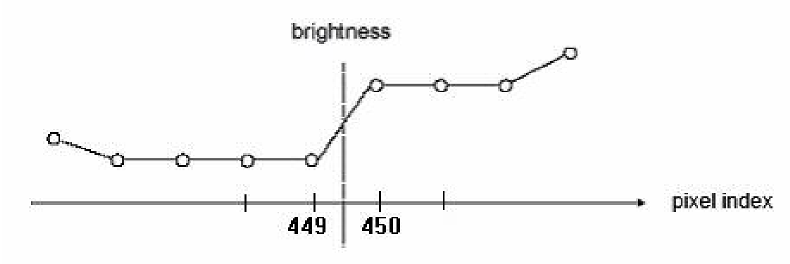


Fig.3- Typical shape of a scan line at the block boundary.

The characteristics of the human visual system (HVS) contribute to a worsened perception of the luminance “jump” between the blocks. Noises with a regular geometric pattern having a strong correlation, such as the blocking artifacts, are about ten times more perceptible than uncorrelated ones [8]. HVS is naturally sensitive to edges, especially vertical and horizontal ones, particularly in uniform or smooth zones [10]. This may be explained by the simple perceptual masking assumption that the blocking effect is a high frequency noise which is less visible in highly detailed areas but very visible in the smooth regions [11].
The problem of the blocking artifacts is complicated by its nonlinear nature and by the strong dependence on the behavior of the HVS, which is also nonlinear [10].

3-The objective evaluation of the compressed image quality
In most image compression applications, the quality of the reconstructed image is objectively evaluated using PSNR (peak signal-to-noise ratio). It may be computed as follows [3, 12, 23].

We assume that a source image f(i,j) has the size nxn pixels and reconstructed by decoding its encoded version F(i,j). The pixel value (luminance value) ranges between 0 (black) and 255 (white), according to the quantization by 8 bits. The MSE of the reconstructed image is computed by:

$$MSE = \frac{\sum [f(i,j) - F(i,j)]^2}{NM} \quad (1)$$

The summation is over all pixels.
The root mean squared error is:

$$RMSE = \sqrt{MSE} \quad (2)$$

After computing RMSE, we compute PSNR in decibels (db) by:

$$PSNR = 20\log_{10}\left(\frac{255}{RMSE}\right) \quad , \text{dB} \quad (3)$$

Typical PSNR values range between 20 and 40 dB and they are usually reported to 2 decimal points (e.g. 25.47 dB). We may compare the values of PSNR for 2 different reconstructed images to decide which one of these images has better overall quality. However, PSNR measure does not equate with human subjective perception [12, 13]. PSNR may not allow always for an accurate judgment on the perceptual image quality. In some cases, after enhancing the quality of the compressed image, the value of PSNR may become lower, though the overall image quality may become better [3, 13]. Therefore, the MPEG committee used an informal threshold of 0.5 dB as the minimum change of PSNR that may indicate minimum significant visible improvement of the overall image quality [12, 14].

Several improved distortion measures have been proposed to evaluate the quality of the compressed image more accurately than by the use of PSNR [3, 15]. However, PSNR is preferred in most image and video compression applications, because it is the most common and the easiest to compute. We may judge if PSNR measure is accurate enough, and acceptable, by comparing it with the subjective evaluation of the image quality. This is especially important for estimating the effectiveness of methods used or proposed for reducing the blocking effect, considering its nonlinear nature, and the related characteristics of the HVS, as we previously mentioned.

4-The proposed Blocking Effect Reduction Methods

Many researches have been carried out for effectively reducing the blocking artifacts, without degradation of other characteristics of the compressed image quality [16, 17, 23, 24]. In general, the various proposed and used blocking effect reduction methods differ in their effectiveness, speed and computational complexity. Some of these methods may be preferred, for specific applications, whereas they may not be suitable for other applications.

We propose two relatively simple and effective methods, using adaptive filtering and adaptive interpolation. The two proposed methods may be considered fast enough because they require relatively simple computations. This is important for real time applications, according to the requirements of video coding [9].

4-1 -The adaptive filtering approach

The proposed adaptive filtering approach is partly similar to the approaches of [9, 11, 18]. However, we propose simpler filtering operations according to the desired compromise between the effectiveness of blocking artifact reduction and computational complexity.

We try to reduce the blocking effect using adaptive low-pass spatial filtering of the compressed image, taking into consideration that low-pass spatial filtering may degrade the sharpness of image details and edges (An edge in the image is usually defined as the boundary between two regions with relatively distinct luminance values [16]). Therefore, we adapt the filtering operations to the local characteristics of the image. We take into consideration the following three observations [9]. First, HVS is more sensitive to blocking artifacts in flat or smooth regions than in complex regions, containing image details and edges. Therefore, we use relatively strong filtering on those regions. Second, in less smooth regions we reduce the strength of filtering and in complex areas we use a weak filtering to try to preserve the sharpness of image details and edges. Third, in video coding, because of motion compensation, blocking artifacts are propagated, and the propagated artifacts are more visible in flat or relatively smooth regions than complex regions. Hence filtering in smooth regions should be applied inside the blocks as well as on the block boundaries [9].

We take into consideration the observations above by using three-mode adaptive low-pass spatial filtering or interpolating.

4-1-A- The three modes of the proposed adaptive filtering

Instead of the relatively complicated filtering approaches, used in [9, 11, 18], we apply the following function, applied in [24] and used in [19] for other applications:

$$h(n) = \begin{cases} a, & n = 0 \\ \frac{1}{4}, & n = \pm 1 \\ \frac{1}{4} - \frac{a}{2}, & n = \pm 2 \end{cases} \qquad (4)$$

The function h(n) represents the "impulse response" used for filtering. It corresponds to the equivalent weighting functions h(x), shown in Fig.4, according to [19] for three values of a(with x denoting the spatial coordinate). We let h(n) = h(x) for x=n.

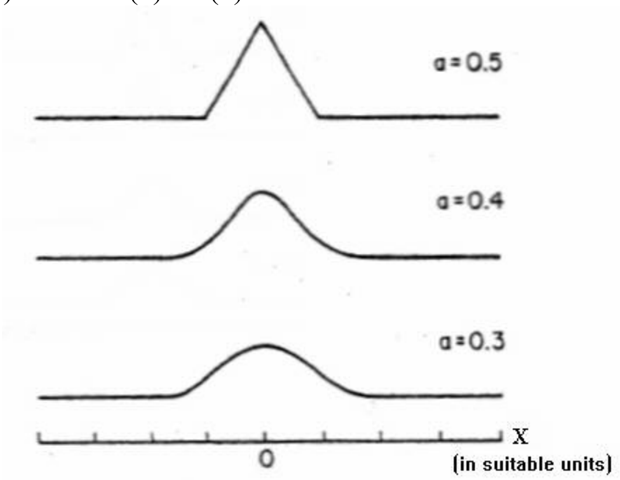


Fig.4- The equivalent weighting functions h(x).

The equivalent weighting function in Fig.4 is Gaussian-like when a=0.4. When a= 0.5 the shape is triangular; when a=0.3 it is flatter and broader than a Gaussian [19].

We first apply the function h(n) horizontally to filter the vertical boundaries, according to a discrete spatial convolution, expressed by the following equation [20]:

$$V_x(x,y) = h(-2).v(x-2,y) + h(-1).v(x-1,y) + h(0).v(x,y) + h(1).v(x+1,y) + h(2).v(x+2,y) \quad (5)$$

with v(x,y) denoting the value of the pixel (x,y).
Then, we apply h(n) vertically to filter the horizontal boundaries, using the equation:

$$V_y(x,y) = h(-2).v(x,y-2) + h(-1).v(x,y-1) + h(0).v(x,y) + h(1).v(x,y+1) + h(2).v(x,y+2) \quad (6)$$

We employ adaptive filtering, using the function h(n) for different values of a , according to the following classification which is partly similar to the approaches of classification or mode selection in [9,11,24].

We classify the image regions according to the positions of the pixels in every block, as shown in Fig.5, by computing the following function, used in [11,24]:

$$count = \Phi(v0-v1) + \Phi(v1-v2) + \Phi(v2-v3) + \Phi(v4-v5) + \Phi(v5-v6) + \Phi(v6-v7) \quad (7)$$

Where Φ(Δ)=1 if |Δ| is less than a threshold th1, and Φ(Δ) is set to zero, otherwise. The value of th1 is selected 3. The value of Φ(Δ) for Δ=v4-v3 was not included in equation(7), because it corresponds to the boundary pixels v3 and v4.

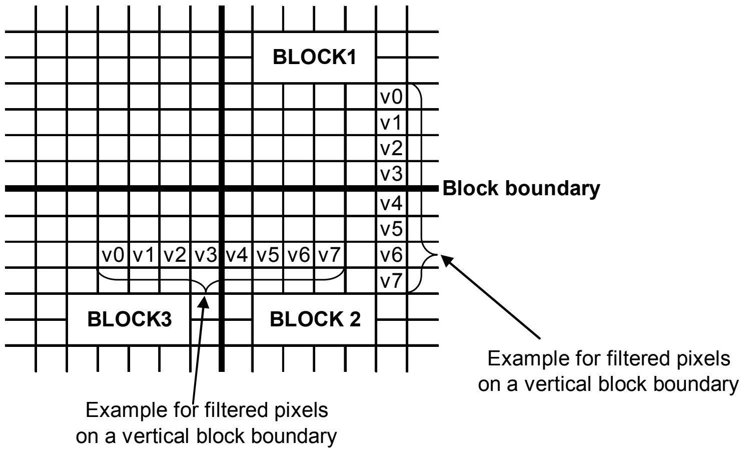


Fig. 5- Examples of positions of pixels.

The function "*count*" reflects the "flatness" of the local image area across a block boundary [9, 11]. We will use the following classification, which is a modified version of the classification, employed in [9] or [11].

1. If the value of *count* equals 5 or 6, the boundary is in the flat or very smooth region.
2. If the value of *count* equals 2, 3 or 4, the boundary is considered nearly smooth.
3. The boundary is considered "complex" [9], if *count* equals 0 or 1.

In the flat region mode, we filter the pixels v1 to v6 in Fig.5. We apply the strongest filtering for the boundary pixels v3 and v4, using the function h(n) with a=0.3. We employ a moderate filtering for the pixels v2 and v5, using h(n) with a=0.4. We apply a weak filtering for the pixels v1 and v6, employing h(n) with a=0.5.

In the nearly smooth area mode, we filter the pixels v2 to v5. We use h(n) with a=0.4 for the boundary pixels v3 and v4 and h(n) with a=0.5 for v2 and v5.

We apply a weak filtering in the complex area mode, using the function h(n) with a=0.5 for the boundary pixels v3 and v4.

4-1-B- The proposed adaptive filtering algorithm

We apply the proposed filtering approach according to the following algorithm:

Along every vertical block boundary

1. Compute *count* according to equation (7) and Fig.5.
2. If count equals 5 or 6 (i.e. If the region is flat), then apply equations (5) and (4) to:
Compute v3 and v4, using h(n) with a=0.3,
Compute v2 and v5, using h(n) with a=0.4,
Compute v1 and v6, using h(n) with a=0.5.
3. If *count* equals 2,3 or 4 (i.e. If the region is nearly smooth), then apply equations (5) and (4) to:
Compute v3 and v4, using h(n) with a =0,4,
Compute v2 and v5, using h(n) with a=0.5.
4. If *count* equals 0 or 1 (i.e. If the region is complex), then apply equations (5) and (4) to:
Compute v3 and v4, using h(n) with a=0.5.
5. Perform the previous computations for all of the vertical block boundaries.

After applying the previous steps along all vertical block boundaries, similar steps are applied along all horizontal block boundaries, using equation (6) instead of (5). If pixel value is changed by the previous filtering operation, the updated value is used for the next filtering (according to a partly similar approach, applied in [9]).

4-2-The adaptive interpolating method

We propose to use the following interpolating functions in some applications, instead of equation (4) in the previous adaptive filtering approach.

a) If *count* equals 5 or 6 (i.e. If the region is flat), we employ the interpolating function:

$$H(n) = [1,1,1,1,1] / 5 \qquad (8)$$

We apply this function to compute each of the pixels v1 to v6 in Fig.5, according to equation (5) or (6).

The simple interpolation employing equation (8) is equivalent to replacing the value of a pixel with the average value of 5 pixels (the filtered pixel and 4 neighboring pixels). The averaging according to equation (8) may be considered as an extreme case of low-pass filtering [8].

b) If *count* equals 2,3 or 4 (i.e. If the region is nearly smooth), we apply interpolation (averaging), using the function:

$$H(n) = [1,1,1] / 3 \qquad (9)$$

We apply this function to compute each of the pixels v1 to v5 in Fig.5 according to equation (5) or (6).

c) If *count* equals 0 or 1 (i.e., if the region is complex), we apply equation (9) only for computing the values of the boundary pixels v3 and v4, according to equation (5) or (6).

5-The results of applying the proposed methods

We implemented both the algorithms and applied first the filtering algorithm on the known "Lena" image, shown in Fig.6 and on the image “Nature”, shown in Fig.7. We repeated the execution of the algorithm on both the images several times, for different numbers of the DCT coefficients, used for the reconstruction of the images (after decoding). Changing the numbers of the DCT coefficients used for the reconstruction may be considered equivalent to changing the quantization accuracy from the view point of changing the compression ratio and, also, from the view point of the visual appearance of the blocking effect. We present the resulted PSNR values in Table 1 for “Lena” image and in Table 2 for the “Nature” image. Fig.8 and Fig.9 show the visual effectiveness of the proposed adaptive filtering algorithm in reducing the blocking artifacts for “Lena” and “Nature” images (for the case 3 x 3 DCT coefficients), with a slight degradation of the image sharpness.

Then, we applied a software program realizing the three-mode interpolation method for both "Lena" and “Nature” images. We present the resulted PSNR values in Table 3 and Table 4. By the visual comparison of "Lena" images after applying the proposed filtering and interpolating methods, we found that the two approaches provide nearly the same degree of “smoothing” of the block boundaries. This may be partly explained by the approximate similarity of the step responses of the averaging and Gaussian filters [8, p.353]. However, we may expect that in some cases the interpolation approach degrades other characteristics of the reconstructed image more than the proposed method of adaptive filtering applying the function (4) with gradual decreasing of the strength of filtering. We see from Tables 3 and 4 that PSNR values for the interpolation method are in more cases worse than the corresponding values in Tables 1 and 2 for the adaptive filtering method.

The gain in Tables 1-4 represents the PSNR improvement. If it is positive, or if it is less negative than a threshold of 0.5 dB, we consider that the algorithm is effective (acceptable for implementation). As mentioned earlier, PSNR may not allow always for an accurate judgment on the overall image quality. However, a slight PSNR improvement such as 0.1 dB may be considered important in some cases [21].

The effectiveness of the two proposed filtering and interpolating methods (for reduction of the blocking effect with an increase, or a slight reduction, of PSNR value) is clearer in cases of small numbers of the DCT coefficients, i.e. in cases corresponding to high compression ratios.

The PSNR values obtained using the two proposed adaptive methods are better in most cases than the values obtained using a simpler two-mode adaptive approach in [24] and much better than the values obtained using a simple un-adaptive interpolating approach in [23].

In general, the results obtained using the two proposed adaptive algorithms show good effectiveness of the two algorithms, taking into consideration the relative simplicity of both the algorithms (e.g. comparing with the filtering approach of [9]). This simplicity is very important for real-time applications of video coding [11]. The two proposed methods may be applied in video decoders, and may be also incorporated in the motion estimation loop in video encoders in some video coding applications, as compared with the filtering approach in [9].

Fig. 6-The “Lena” image.

Fig.7-The “Nature” image

	PSNR (dB)			
No. Of coefficients	**DCT**	**DCT+The Proposed Method**	**Gain**	**Acceptable?**
1x1	**22.81**	**23.46**	**0.65**	**Yes**
2x2	**26.02**	**26.34**	**0.32**	**Yes**
3x3	**27.77**	**27.85**	**0.08**	**Yes**
4x4	**28.75**	**28.64**	**-0.11**	**Yes**
5x5	**29.26**	**29.03**	**-0.23**	**Yes**
6x6	**29.51**	**29.2**	**-0.31**	**Yes**
7x7	**29.63**	**29.28**	**-0.35**	**Yes**
8x8	**29.69**	**29.32**	**-0.37**	**Yes**

Table 1- PSNR results for "Lena" image with 3 x 3 DCT coefficients (Adaptive Filtering)

	PSNR (dB)			
No. Of coefficients	**DCT**	**DCT+The Proposed Method**	**Gain**	**Acceptable?**
1x1	**20.51**	**20.7**	**0.19**	**Yes**
2x2	**21.99**	**22.08**	**0.09**	**Yes**
3x3	**23.12**	**23.09**	**-0.03**	**Yes**
4x4	**24.15**	**23.95**	**-0.2**	**Yes**
5x5	**25.12**	**24.7**	**-0.42**	**Yes**
6x6	**25.98**	**25.28**	**-0.7**	**No**
7x7	**26.64**	**25.71**	**-0.93**	**No**
8x8	**27.11**	**26.05**	**-1.06**	**No**

Table2 - PSNR results for "Nature" image with 3 x 3 DCT coefficients (Adaptive Filtering)

	PSNR (dB)			
No. Of coefficients	**DCT**	**DCT+The Proposed Method**	**Gain**	**Acceptable?**
1x1	**22.81**	**23.58**	**0.77**	**Yes**
2x2	**26.02**	**26.31**	**0.29**	**Yes**
3x3	**27.77**	**27.79**	**0.02**	**Yes**
4x4	**28.75**	**28.55**	**-0.2**	**Yes**
5x5	**29.26**	**28.93**	**-0.33**	**Yes**
6x6	**29.51**	**29.09**	**-0.42**	**Yes**
7x7	**29.63**	**29.16**	**-0.47**	**Yes**
8x8	**29.69**	**29.2**	**-0.49**	**Yes**

Table 3- PSNR results for “Lena” image with 3 x 3 DCT coefficients (Adaptive Interpolation)

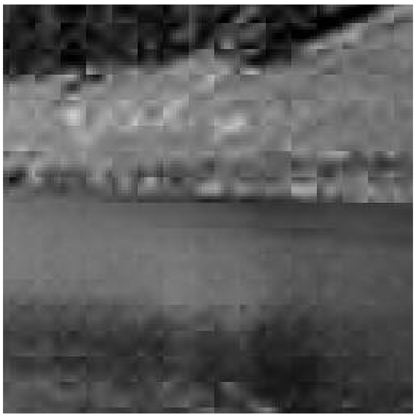
	PSNR (dB)			
No. Of coefficients	**DCT**	**DCT+The Proposed Method**	**Gain**	**Acceptable?**
1x1	**20.51**	**20.73**	**0.22**	**Yes**
2x2	**21.99**	**22.06**	**0.07**	**Yes**
3x3	**23.12**	**23.03**	**-0.09**	**Yes**
4x4	**24.15**	**23.84**	**-0.31**	**Yes**
5x5	**25.12**	**24.5**	**-0.62**	**No**
6x6	**25.98**	**24.98**	**-1**	**No**
7x7	**26.64**	**25.33**	**-1.31**	**No**
8x8	**27.11**	**25.63**	**-1.48**	**No**

Table4 - PSNR results for “Nature” image with 3 x 3 DCT coefficients (Adaptive Interpolation)

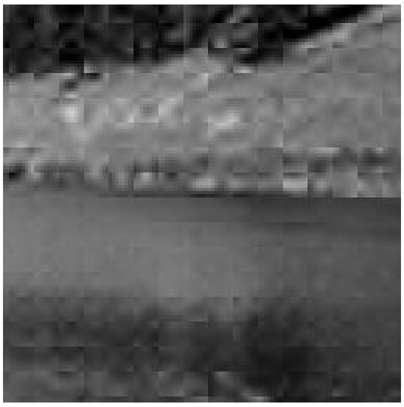
(a)

(b)

Fig.8- A part of "Lena" image with 3x3 DCT coefficients
a) Before filtering
b) After filtering



(a)



(b)

Fig.9- A part of "Nature" image with 3x3 DCT coefficients
c) Before filtering
d) After filtering

6-Conclusions

In this research we illustrated the blocking effect in BDCT image and video coding. This effect is considered as the most annoying distortion in the reconstructed images in applications with high compression ratios. We proposed two adaptive three-mode filtering and interpolating methods for reducing the blocking effect or artifacts. We took into consideration the fact that, in video coding, because of motion compensation, blocking artifacts are propagated, and the propagated artifacts are more visible in smooth regions than in complex regions. Hence, we applied filtering or interpolating filtering in smooth regions inside the blocks as well as on the block boundaries. We presented the results of applying the two algorithms, realizing both methods. The obtained results show the good effectiveness of the two proposed methods, taking into consideration their relative simplicity. This simplicity is particularly important in real-time applications of video coding. The two methods may be applied in video decoders, and may be also incorporated in the motion estimation loop in video encoders in some video coding applications.

References

1- Y.K. Lai, J. Li, C.C.J. Kuo, " Removal of Blocking Artifacts of DCT Transform by Classified Space-Frequency Filtering"(1995). Available from:
Http://research.microsoft.com/~jinl/old/publication/asilomar95_blocking.pdf.
2- Y.K. Lai, J. Li, C.C.J. Kuo," Image Enhancement for Low Bit-rate JPEG and MPEG Coding via Postprocessing"(1996). Available from: http://research.microsoft.com/users/jinl/paper_1996/vcip96.PDF.
3- Iain E G Richardson, "Video CODEC Design", John Wiley & Sons, 2002.
4- Stefan Winkler, "Color image quality on the Internet", in Proc. SPIE/IS&T Internet Imaging, vol. 5304, pp.118-131, 2004. Available from: http://stefan.winkler.net/Publications/ii2004.pdf.
5- Nam Ik Cho, "Reduction of Blocking Artifacts by Cepstral Filtering", 2001. Available from:
Http://ispl.snu.ac.kr/publications_pdf/ceps.pdf.
6-Z. Xiong el al., "A Deblocking Algorithm for JPEG Compressed Image Using Overcomplete Wavelet Representations", IEEE Trans. Circuits And Systems for Video Technology, Apr. 1997.
7-M.Robin, M.Poulin, "Digital Television Fundamentals", Mcgraw-Hill, 2000.
8- B. Jeon, J. Jeong, "Blocking Artifacts Reduction in Image Compression with Block Boundary Discontinuity Criterion", IEEE Trans. Circuits And Systems for Video Technology, June, 1998.
9- S.D. Kim, J Yi, H.M. Kim, J. Beom, " A Deblocking Filter with Two Separate Modes in Block-Based Video Coding", IEEE Trans. Circuits And Systems for Video Technology, Feb., 1999.
10- R. Castagno, S. Marsi, G. Ramponi ,"A Simple Algorithm for the Reduction of Blocking Artifacts in Images and Its Implementation" ,IEEE Trans. On Consumer Electronics, Aug., 1998.
11- M.Y. Shen, J.W. Kim, C.C.J. Kuo, " Fast Compression Artifact reduction Technique Based on Nonlinear Filtering"(1995). Available from:http://netmedia.kjist.ac.kr/jongwon/papers/99iscas-post.pdf.

12-L.A. Rowe, "Image Quality Computation"(1997). Available from: http://bmrc.berkeley.edu/courseware/cs294/fall97/assignment/psnr.html.
13-C.A.Segall,P.Karunaratne,A.K.Katsaggelos, " Pre-processing of compressed digital video"(2000). Available from: ivpl.ece.northwestern.edu/Publications/Conferences/2000/asilomar00.pdf.
14- A. Paquet, "Blind measurement of blocking artifacts in JPEG compressed images" (University of British Columbia, 2000). Available from:
www.alex.paquet.com/Papers.html.
15- Z. Wang, A. C. Bovik, B. L. Evans, "Blind measurement of blocking artifacts in images ", IEEE International Conference on Image Processing, vol. 3, Sept. 2000. Available from:
Http://www.cns.nyu.edu/~zwang/files/papers/icip00.pdf.
16-Y.L. Lee ,H.C. Kim, H.W. Park, " Blocking effect reduction of JPEG images by signal adaptive filtering", IEEE Trans. Image Process., Feb., 1998.
17- D. Petrescu , " Efficient Implementation Of Video Post-Processing Algorithms On The Bops Parallel Architecture". BOPS Inc., 6340...(2000). Available from: http://mti.xidian.edu.cn/multimedia/2001/supp/icassp2001/MAIN/papers/pap1548.pdf.
18- P. List, A. Joch, J. Lainema, G. Bjntegaard, M. Karczewicz, " Adaptive Deblocking Filter", IEEE Trans. Circuits And Systems for Video Technology, Jul., 2003.
19- P.J. Burt, E.H. Adelson, " The Laplacian Pyramid as a Compact Image Code", IEEE Trans. Com., Apr., 1983.
20- S.K. Kon, J. H. Yoon, Y.M. Ro," Image Enhancement with Attenuated Blocking Artifact in Transform Domain". IEICE Trans. Inf. & Syst., Vol.E85-D, No.1 January 2002. Available from: vega.icu.ac.kr/~ivy/ivy_publication/International_Journal/OSK/OSK01.pdf.
21- K. Ramkishor, P. Karandikar, "A Simple and Efficient Deblocking Algorithm for Low Bit-Rate Video Coding", (2000). Available from: http://www.geocities.com/ramkishor/papers/Post_processing_ISCE.pdf.

22- د. عصام عبود والمهندس محمد باسل المجتهد " ضغط الصور الثابتة باستخدام التحويل التجيبي المتقطع والهرم اللابلاسي"، مجلة باسل الأسد للعلوم الهندسية، العدد 19، 2004.

23- د. عصام عبود " إنقاص التأثير الكتلي الناتج عن ضغط الصور المعتمد على تحويل التجيب المتقطع"، مقبول للنشر في مجلة باسل الأسد للعلوم الهندسية (قيد النشر).

24- د. عصام عبود " خفض التأثير الكتلي في ترميز الصورة و الترميز المرئي بترشيح متكيف "، مقبول للنشر في مجلة باسل الأسد للعلوم الهندسية(قيد النشر).

Received, 17 October, 2005.