Efficient DCT-domain Blind Measurement and Reduction of Blocking Artifacts

Shizhong Liu *Student member* and Alan C. Bovik *Fellow* IEEE

Laboratory for Image and Video Engineering
Dept. of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX 78712-1084, USA

Abstract

Blocking artifacts continue to be among the most serious defects that occur in images and video streams compressed to low bitrates using block DCT-based compression standards(*e.g.*, JPEG, MPEG and H.263). It is of interest to be able to numerically assess the degree of blocking artifact in a visual signal, for example, in order to objectively determine the efficacy of a compression method, or to discover the quality of video content being delivered by a web server. In this paper, we propose new methods for efficiently assessing, and subsequently reducing, the severity of blocking artifacts in compressed image bit-streams. The method is blind, and operates only in the DCT domain. Hence it can be applied to unknown visual signals, and it is efficient since the signal need not be compressed or decompressed. In the algorithm blocking artifacts are modeled as 2-D step functions. A fast DCT-domain algorithm extracts all parameters needed to detect the presence of, and estimate the amplitude of blocking artifacts, by exploiting several properties of the human vision system (HVS). Using the estimate of blockiness, a novel DCT-domain method is then developed which adaptively reduces detected blocking artifacts. Our experimental results show that the proposed method of measuring blocking artifacts is effective and stable across a wide variety of images. Moreover, the proposed blocking artifact reduction method exhibits satisfactory performance as compared to other post-processing techniques. The proposed technique has low computational cost hence can be used for real-time image/video quality monitoring and control, especially in applications where it is desired that the image/video data be processed directly in the DCT-domain.

Keywords

DCT-domain, blocking artifacts, image quality control, human visual perception, video post-processing.

I. Introduction

Transform coding is the heart of several industry standards for image and video compression. In particular, the discrete cosine transform (DCT) is the basis for the JPEG image coding standard [1], the MPEG video coding standard [2], and the ITU-T H.261 [3] and H.263 recommendations [4] for real-time visual communications. In a typical DCT compression scheme, the input image is divided into small blocks (typically 8×8), each block being transformed independently to convert the image elements to DCT coefficients. The DCT coefficients are then quantized using a scalar quantizer defined by the so-called quantization matrix. Since the DCT usually efficiently concentrates most of the signal energy into few coefficients, the DCT block tends to be quite sparse after quantization. Finally, all of the quantized DCT coefficients are converted to a bit-stream via variable

length encoding. At the decoder end, the received data is decoded, de-quantized, and reconstructed by the Inverse DCT (IDCT). At high or moderate bit rates, the DCT coded image yields excellent reproduction without noticeable artifacts. However, at low bit rates, the reconstructed images generally suffer from visually annoying artifacts as a result of coarse quantization. One major artifact is the blocking effect, which appears as artificial block boundaries between adjacent blocks. Fig. 1 shows two versions of the image *Lena* JPEG-coded at different bit-rates. As can be seen, there is no visible blocking artifact in the image coded at 1.0 bit per pixel (bpp) while severe blocking artifacts exist in the one coded at 0.217 bpp. Since blocking artifacts significantly degrade the visual quality of the reconstructed images, it is desirable to be able to monitor and control the visibility of blocking effects in DCT coded images.

Over the past several years, quite a bit of research has been applied to the problem of alleviating blocking effects in block DCT coded images. At the encoding end, different transform schemes have been suggested, such as the interleaved block transform [5],the combined transform [6] and so on. However, since all these transform schemes do not conform with the existing image/video coding standards such as JPEG, MPEG, it is difficult or impossible to adequately integrate them with the existing standards.

The other strategy is to reduce blocking artifacts via post-processing techniques at the decoder side. This strategy is of practical interest since it only requires the decoded image, hence is fully compatible with the coding standards. Typical post-processing techniques include block-boundary filtering techniques [4,7], iterative methods based on the theory of projections onto convex sets (POCS) [8,9], and the maximum *a posteriori* probability approach [10], etc. However, iterative POCS-based methods usually have high computational complexity, and thus are difficult to adapt to real-time video or image processing applications. The simplest approach, global low-pass filtering, has the obvious defect of blurring information-bearing detail. To reduce blocking artifacts without significantly degrading valid high-frequency information, several space-variant adaptive filtering algorithms have been proposed. For example, In [11], the visibility of artifacts at each block boundary were first computed based on several characteristics of the Human Vision System (HVS); this information was used to guide the selection of an adaptive non-linear, space-variant

smoothing operation at block boundaries.

It is known that blocking artifacts are introduced by the coarse quantization of DCT coefficients in each block. While most post-processing techniques proposed in the literature suppress blocking artifacts by changing the pixel values near the block boundary, some DCT domain post-processing algorithms have also been investigated recently, where the blocking artifacts are reduced by directly manipulating DCT coefficients instead of pixel values . In [12], Jeon and Jeong proposed a post-processing method that can achieve *minimum discontinuity* of pixel values over block boundaries by compensating the loss of a coefficient's accuracy in the transform domain. To quantify the blocking artifacts, a block discontinuity measure was defined as the sum of the squared pixel differences across the four block boundaries. They concluded that there were unique solutions of up to 28 compensating terms, when the criterion was to minimize pixel discontinuities along block boundaries. In [13], Zeng proposed a simple DCT domain method for blocking effect reduction, applying a zero-masking to the DCT coefficients of some shifted image blocks. However, the loss of edge information caused by the zero-masking scheme can be noticed. More recently, Chen *et al.* [14] introduced a DCT-domain post-filtering approach to reduce blocking artifacts, where the post-filter made use of the DCT coefficients of shifted blocks in order to obtain a close correlation between the DCT coefficients at the same frequency. The filtering adapts according to the local activity of each block, to achieve simultaneous artifact reduction and detail preservation. However, twenty-four shifted blocks are needed for each block in smooth image regions, while eight are needed in other areas, hence the computational cost is rather high [15, 16].

In this paper, we propose a novel and efficient DCT-domain algorithm for the measurement and reduction of blocking artifacts in DCT-coded images. By constituting a new shifted block across any two adjacent blocks, blocking artifacts are modeled as 2-D step functions [13] in the shifted block. Then a DCT-domain method for blind measurement of the local visibility of the blocking artifacts is defined, utilizing properties of the HVS. A DC image, consisting of all the DC components from each block, is used to detect primary edges in the image so that the edge information in the original image can be preserved. Based on the local visibility measure of the blocking artifacts at each block edge, and on

the result of edge detection, all of the block edges in the original image are divided into three categories. Different processing methods are applied to the three types of block edges in order to reduce the blocking artifacts while preserving the integrity of the original image as much as possible. Experimental results show that the proposed method of measuring blocking artifacts is robust and consistent with human perception; the proposed scheme for reducing blocking artifacts yields excellent blocking artifact suppression while maintaining good image quality. Another advantage of the proposed scheme is that it has a low computational complexity, making it suitable for real-time applications. The proposed technique can be easily integrated into standards-compliant DCT-domain image/video processing applications.

The rest of this paper is organized as follows. Section II describes in detail the modelling of blocking artifacts and blind measurement of the local visibility of blocking artifacts. In section III, the algorithm to reduce blocking artifacts is presented. Simulation and comparisons with other post-processing methods are given in Section IV. Lastly, Section V concludes the paper.

II. Blind Measurement of Blocking Artifact

An ideal blocking artifact reduction algorithm should only remove visible blocking artifacts while maintaining the original image content as much as possible. To achieve this, we have developed a HVS-based method that measures the local visibility of blocking artifacts. Several prior methods have sought to measure blocking artifacts in DCT-coded images [11, 17, 18]. In [17], a visual model was developed that delivers a numerical value quantifying the visibility of the blocking effect in compressed images. However, the model requires the original images for comparison to the reconstructed (decompressed) images. In practice, the original images will not be available. Therefore, blind block artifact measurement methods are of much greater interest, where no original image is needed. Several authors have also pursued this avenue; in [18], the blocky image is modeled as a non-blocky image interfering with a pure blocky signal. The measurement of blocking artifacts is accomplished by estimating the power of the blocky signal.

In this paper we propose a novel DCT-domain method for blind measurement of blocking artifacts, by modeling the artifacts as 2-D step functions in shifted blocks. Since blocking

effects occurring in the horizontal and vertical directions generally have no difference in principle, we shall next describe the proposed algorithm for measuring horizontal blocking artifacts.

A. Model of Blocking Artifact

Given a block-DCT-compressed image, we will model each block as a constant block distorted by i.i.d. white noise with zero mean and unknown variance. Consider two adjacent 8×8 blocks b_1 and b_2 with average values μ_1 and μ_2, respectively, where $\mu_1 \neq \mu_2$. Thus we model these two blocks as follows:

$$b_1 = \mu_1 + \varepsilon_{i,j}, \quad b_2 = \mu_2 + \delta_{i,j}$$

where $\varepsilon_{i,j}$ and $\delta_{i,j}$ are modeled as i.i.d white noise blocks with zero mean. When the corresponding DCT blocks of b_1 and b_2 are quantized using a large quantization parameter, most of the DCT coefficients become zero, which reduces the effect of the variance of $\varepsilon_{i,j}$ and $\delta_{i,j}$. As a result, a 2-D step function between b_1 and b_2 (due to $\mu_1 \neq \mu_2$) may become visible, creating a blocking artifact as shown in Fig. 1(b). Based on this observation, we form a *new shifted block* $\hat{b}$ composed of the right half of b_1 and the left half of b_2 as shown in Fig. 2. The blocking artifact between blocks b_1 and b_2 can be modeled as a 2-D step function in the block $\hat{b}$. Define a 2-D step block s in the new shifted block $\hat{b}$ as follows:

$$s(i,j) = \begin{cases} -\frac{1}{8} & i = 0, \ldots, 7; j = 0, \ldots, 3 \\ \frac{1}{8} & i = 0, \ldots, 7; j = 4, \ldots, 7. \end{cases} \tag{1}$$

Therefore

$$\hat{b}(i,j) = \beta \cdot s(i,j) + \mu + r(i,j); \quad i,j = 0, \ldots, 7, \tag{2}$$

where $|\beta|$ is the amplitude of the 2-D step function s, μ is the average value of the block $\hat{b}$, indicating the local background brightness, and r is the residual block, which describes the local activity around the block edge. The larger the value of $|\beta|$, the more serious the blocking effect is taken to be, provided that the background brightness and local activity remain unchanged. In the following, an efficient DCT-domain algorithm is implemented which computes the DCT coefficients of the block $\hat{b}$ and the parameter values in (2).

B. Efficient DCT-domain Algorithm

First, let us define two matrices q_1 and q_2 as follows:

$$q_1 = \begin{bmatrix} O & O_{4\times4} \\ I_{4\times4} & O \end{bmatrix}, \quad q_2 = \begin{bmatrix} O & I_{4\times4} \\ O_{4\times4} & O \end{bmatrix}$$

where I is an identity matrix, and O is a zero matrix. Thus, in Fig. 2, $\hat{b}$ can be written as

$$\hat{b} = b_1 q_1 + b_2 q_2. \tag{3}$$

Using the linear and distributive properties of the DCT, the DCT-domain counterpart of (3) is easily obtained:

$$\hat{B} = B_1 Q_1 + B_2 Q_2 \tag{4}$$

where $\hat{B}$, B_1, B_2, Q_1 and Q_2 are the DCT's of $\hat{b}$, b_1, b_2, q_1 and q_2, respectively. Although the matrices q_1 and q_2 are quite sparse, the corresponding matrices Q_1 and Q_2 in the DCT-domain are not sparse. This means that many multiplications may be necessary in (4) to obtain $\hat{B}$. It is observed that the sum $F_+ = Q_1 + Q_2$ and the difference $F_- = Q_1 - Q_2$ are quite sparse (See Appendix for details.). More than 50% of the elements of F_+ and F_- are zero. Therefore, significant computational savings can be achieved by using the following relation:

$$\begin{aligned} \hat{B} &= B_1 Q_1 + B_2 Q_2 \\ &= \frac{1}{2}[(B_1 + B_2)(Q_1 + Q_2) + (B_1 - B_2)(Q_1 - Q_2)] \\ &= \frac{1}{2}(B_+ F_+ + B_- F_-) \end{aligned} \tag{5}$$

where $B_+ = B_1 + B_2$ and $B_- = B_1 - B_2$. Note that the 8×8 DCT transform of the 2-D step block s defined in (1) only has four non-zero elements in the first row since s is constant in the vertical direction and anti-symmetric in the horizontal direction. Let the vector $\upsilon = [\upsilon_0, \upsilon_1, \ldots, \upsilon_7]$ be the first row of the 8×8 DCT transform of the 2-D step function. Then $\upsilon_0 = \upsilon_2 = \upsilon_4 = \upsilon_6 = 0$. By the unitary property of the DCT transform

$$\|\upsilon\|_2 = \sqrt{\sum_{i=0}^{7} \upsilon_i^2} = \sqrt{\sum_{m=0}^{7}\sum_{n=0}^{7} s^2(m,n)} = 1. \tag{6}$$

Hence the parameters in (2) can be computed as follows:

$$\mu = \hat{B}(0,0)/8 \tag{7}$$

$$\begin{aligned} \beta &= \sum_{j=0}^{7} \upsilon_j \hat{B}(0,j) \\ &= \upsilon_1 \hat{B}(0,1) + \upsilon_3 \hat{B}(0,3) + \upsilon_5 \hat{B}(0,5) + \upsilon_7 \hat{B}(0,7). \end{aligned} \tag{8}$$

Let R be the 8×8 DCT transform of the residual block r in (2). Then R can be easily obtained by the following steps:

$$\begin{aligned} R &= \hat{B} \\ R(0,0) &= 0 \\ R(0,i) &= R(0,i) - \beta \cdot \upsilon_i \quad i = 0, \ldots, 7. \end{aligned}$$

Because of the sparseness of DCT coefficients in the DCT block, the proposed DCT-domain algorithm is far more efficient than conventional IDCT-DCT methods such as [13], even if fast IDCT-DCT algorithms are employed. Now that we have shown how to extract all of the parameters needed in the blocking model, we may describe our method of measuring the blocking artifacts, making use of known properties of the HVS.

C. HVS Based Measurement of Blocking Artifacts

In most applications, the human observer is usually the end consumer of visual information. Thus, measuring blocking artifacts using relevant and accurately known properties of human perception should prove efficacious for judging the visibility of the blocking effect [11, 17, 18]. *Texture* (or activity) *masking* and *luminance* (or brightness) *masking* are two well-studied properties of the HVS which are highly relevant to the perception of blocking artifacts [19]. We consider activity masking first. In order to exploit the activity masking property of the HVS, we assume that the total activity of a given block is simply the sum of the activities due to all of its spatial frequencies, and that the DCT coefficient is the amplitude of the expected component. The masking effect also depends on the relative orientation of the masking signal [19]. Since blocking artifacts manifest in only two

particular directions (vertical and horizontal) we define two distinct oriented activities, vertical activity A_v and horizontal activity A_h, as follows:

$$A_v = \sum_{v=1}^{7} v \sum_{u=0}^{7} |R(u,v)| \tag{9}$$

$$A_h = \sum_{u=1}^{7} u \sum_{v=0}^{7} |R(u,v)|. \tag{10}$$

For vertical blocking artifacts, the masking effect due to horizontal activity will be dominant, which leads to

$$A^v_{total} = A_h + \alpha A_v. \tag{11}$$

Similarly, for the horizontal artifacts, the total activity may be written as

$$A^h_{total} = A_v + \alpha A_h \tag{12}$$

where $\alpha = 0.8$, according to [11].

We also consider brightness masking: the visibility of blocking artifacts depends on the local background luminance. Taking into account both masking properties of the HVS, the model below is employed to compute the visibility of blocking artifacts, η:

$$\eta = \frac{|\beta|}{(1 + A^h_{total})(1 + (\frac{\mu}{\mu_0})^\gamma)} \tag{13}$$

where β and μ are given in (7) and (8), respectively; A^h_{total} is the total horizontal activity in the block $\hat{b}$ and $\mu_0 = 150$, $\gamma = 2$ [17].

For vertical blocking artifacts, an identical method can be applied. The 2-D map of the visibility of blocking artifacts across the image that results from this process will be used in the next section to adaptively remove blocking artifact at each block edge.

The local blockiness measures can also be aggregated to produce a single numerical value that predicts the overall image quality as follows:

$$\Theta = \sqrt[p]{\frac{1}{N} \sum_{k=1}^{N} \eta_k^p} \tag{14}$$

where N is the total number of inter-block boundaries in the image, Θ is the global measure of the blocking artifacts in the image, and the exponent $p = 4$ [11].

III. Blocking Artifacts Reduction Algorithm

Given a DCT block edge, if its measured visibility η falls below a certain threshold τ, no further process need be applied to the block edge. However, if $\eta \geq \tau$, then the it is deemed that reduction of a blocking artifact will be necessary. The threshold τ is determined experimentally. In order to avoid blurring information-bearing edges in the image, a DC image, consisting of the DC components from each DCT block, is formed. Meaningful (non-artifact) edges are detected from this reduced resolution image. Blocks that are contiguous to the detected edges are called *edge blocks*. According to the measured local visibility of the blocking effect and the results of edge detection, all of the block edges in the image are divided into the following three categories :

- Type I: $\eta < \tau$, no further process;
- Type II: $\eta \geq \tau$, and neither of the two adjacent blocks are *edge blocks*;
- Type III: $\eta \geq \tau$, and at least one of the adjacent blocks is an *edge block*.

A detailed description of the proposed blocking artifacts reduction algorithm is given later.

A. Edge detection in DC image

Since all of the DCT coefficients in the image are available, the DC image can be formed simply by taking the DC component from each block in the image. So the size of the DC image is only $\frac{1}{64}$ of the original image. The Sobel gradient operator [20] is used to calculate the magnitude of the gradient at each pixel $x_{m,n}$ in the DC image

$$\begin{aligned} mag(\bigtriangledown x_{m,n}) &= |x_{m-1,n-1} + 2x_{m,n-1} + x_{m+1,n-1} \\ &\quad -x_{m-1,n+1} - 2x_{m,n+1} - x_{m+1,n+1}| \\ &\quad +|x_{m-1,n-1} + 2x_{m-1,n} + x_{m-1,n+1} \\ &\quad -x_{m+1,n-1} - 2x_{m+1,n} - x_{m+1,n+1}|. \end{aligned}$$

A pixel is classified as an edge pixel *only if* $mag(\bigtriangledown x_{m,n}) \geq T$, where T is a preset threshold. After obtaining the binary edge map from the edge detection process, a binary morphological operation [20] is employed to remove isolated edge pixels, which might cause false alarms during the edge detection. Since each pixel in the DC image maps an 8×8

block in the original image, each block corresponding to the edge pixel in the DC image is marked as *edge block* in the original image. Fig. 3 shows an example of edge detection, where it can be seen that most primary edges in the image have been detected. However, the computational cost of the edge detection is quite low since it is done on the DC image and since the relatively simple Sobel operator is employed.

B. Reduction of Blocking Effect for Type II Block edges

In Fig. 2, if the edge between blocks b_1 and b_2 belongs to Type II, it means that neither b_1 nor b_2 is an *edge block*. In this case, we replace the 2-D step block s with a 2-D linear block in the shifted block $\hat{b}$. Fig. 4 shows the 1-D case, where $f(x) = \frac{1}{8}sign(x)$ is the step function, and $g(x)$ is the linear function used to replace $f(x)$. Using the parameters in Fig. 4, we find that

$$g(x) = \frac{1}{28}x. \tag{15}$$

Thus, eight pixel values on the linear function can be obtained

$$d = [g(-3.5), g(-2.5), g(-1.5), g(-0.5), g(0.5), g(1.5), g(2.5), g(3.5)].$$

The 2-D 8×8 linear block l can be constituted by simply stacking the vector d row by row, *i.e.,*

$$l = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{bmatrix}_{8\times 8}.$$

Note that the block l is anti-symmetric horizontally and constant in the vertical direction. Therefore, the 8×8 DCT transform of l has only four non-zero elements in the first row like the 2-D step block s in (1). Let $\lambda = [\lambda_0, \ldots, \lambda_7]$ be the vector of the first row of the 8×8 DCT transform of l. Then $\lambda_0 = \lambda_2 = \lambda_4 = \lambda_6 = 0$ due to the anti-symmetric property of l in the horizontal direction.

Define a vector $\delta = \lambda - \upsilon$, where υ, defined in Section II, is the vector of the first row of the 8×8 DCT transform of the 2-D step function s. Let $\hat{b}'$ be the new block after replacing the 2-D step block s with the 2-D linear linear l in the block $\hat{b}$ (See Fig. 2). Then

$$\hat{b}'(i,j) \;=\; \hat{b}(i,j) - \beta \cdot s(i,j) + \beta \cdot l(i,j)$$

$$= \hat{b}(i,j) + \beta \cdot [l(i,j) - s(i,j)] \qquad (16)$$
$$i,j = 0,\ldots,7,$$

where β is given in (8). The DCT transforms of both s and l only have four non-zero elements in the first row, represented by the vectors υ and λ, respectively. The DCT-domain implementation of (16) turns out to be very simple. That is

$$\hat{B}'(0,j) = \hat{B}(0,j) + \beta \cdot (\lambda_j - \upsilon_j) = \hat{B}(0,j) + \beta \cdot \delta_j;$$
$$\hat{B}'(i,j) = \hat{B}(i,j); \quad i > 0, \quad j = 0,\ldots,7 \qquad (17)$$

where $\hat{B}'$ is the 8×8 DCT transform of $\hat{b}'$.

The block $\hat{B}'$ is then used to update both B_1 and B_2 (See Fig. 2) by using matrix multiplications similar to (4). However, since only the elements of the first row of $\hat{B}$ have been changed to obtained $\hat{B}'$ in (17), it can be seen from (4), that the changes only affect the elements in the first row of B_1 and B_2. Therefore, a simple way to update the blocks B_1 and B_2 can be found. Let δ_1 and δ_2 be the delta vectors in B_1 and B_2, respectively. Then

$$\delta_1 = \delta Q_2; \quad \delta_2 = \delta Q_1$$

where Q_1 and Q_2 are defined in (4). Finally, the blocks B_1 and B_2 are updated as follows:

$$B_1(0,j) = B_1(0,j) + \beta \cdot (\delta_1)_j; \qquad j = 0,\ldots,7.$$
$$B_2(0,j) = B_2(0,j) + \beta \cdot (\delta_2)_j; \quad j = 0,\ldots,7. \qquad (18)$$

Note that δ_1 and δ_2 are constant vectors, which can be pre-computed and stored.

C. Post-filtering Type II and Type III Block Edges in the DCT-domain

In the last subsection, after updating the two adjacent blocks B_1 and B_2 according to (18), a new and artificial discontinuity, though less visible, might have been created at the middle of the two blocks. To eliminate newly-created artifacts, a DCT-domain post-filtering method [14] is applied to both updated blocks. Meanwhile, the same post-filtering method is also applied to those blocks that have associated Type III block edges to remove blocking artifacts in the edge areas.

Let $b_{m,n}$ be the 8×8 block at the m^{th} block row and n^{th} block column in the image, and $b_{m,n}^{k,l}$ be the block with a displacement of k pixels in the y direction and l pixels in the x direction, relative to the block $b_{m,n}$. Fig. 5 shows an example with $k = -1, l = -1$. Obviously, the block $b_{m,n}^{k,l}$ would overlap the block $b_{m,n}$ if $|k| < 8$ and $|l| < 8$ as shown in Fig. 5. Let $B_{m,n}$ and $B_{m,n}^{k,l}$ be the DCT's of the block $b_{m,n}$ and $b_{m,n}^{k,l}$, respectively. DCT-domain post-filtering is employed to manipulate $B_{m,n}$ as follows:

$$\hat{B}_{m,n}(u,v) = \frac{1}{W} \sum_{k=-h}^{h} \sum_{l=-h}^{h} w_{k,l} B_{m,n}^{k,l}(u,v) \tag{19}$$

where $\hat{B}_{m,n}(u,v)$ is the filtered DCT block; h is the maximum displacement in both x and y directions; $w_{k,l}$ is the weighting factor associated with the block $B_{m,n}^{k,l}$; and W is the summation of all weighting factors, *i.e.*,

$$W = \sum_{k=-h}^{h} \sum_{l=-h}^{h} w_{k,l}.$$

Here we select $h = 1$ and the weighting factors are chosen as

$$w_{k,l} = \begin{cases} 3 & \text{for } (k,l) = (0,0) \\ 1 & \text{otherwise.} \end{cases}$$

and $k, l = -1, 0, 1$. A large weight on the central block to avoid excessively blurring the image detail.

The DCT block $B_{m,n}^{k,l}$ can be directly computed from its four overlapped neighboring blocks in the DCT-domain [15], *i.e.*

$$B_{m,n}^{k,l} = \sum_{x=0}^{1} \sum_{y=0}^{1} S_{xy(1)} B_{m+k+x,n+l+y} S_{xy(2)} \tag{20}$$

where $S_{xy(1)}$ and $S_{xy(2)}$ are the DCT's of the pre-matrix $s_{xy(1)}$ and post-matrix $s_{xy(2)}$, respectively, as defined in [15]. Brute-force implementation of (20) is very costly in terms of computation. Recently, several fast algorithms have been proposed to reduce the computation complexity of (20) [16, 21, 22].

D. Quantization Constraint

After the post-processing described above, the quantization constraint [8, 9] is applied to each DCT coefficient in the processed blocks to ensure that all DCT coefficients remain

within their original quantization interval. Since the quantization parameters are known at the receiver side, the corresponding quantization interval $[B_{m,n}^{min}(u,v), B_{m,n}^{max}(u,v)]$ of coefficient $B_{m,n}(u,v)$ is

$$\begin{cases} B_{m,n}^{min}(u,v) = Q(u,v)(B_{m,n}(u,v) - 0.5) \\ B_{m,n}^{max}(u,v) = Q(u,v)(B_{m,n}(u,v) + 0.5) \end{cases}$$

where $Q(u,v)$ is the quantization step size for $B_{m,n}(u,v)$. Therefore, the coefficient $B_{m,n}(u,v)$ is clipped into the interval $[B_{m,n}^{min}(u,v), B_{m,n}^{max}(u,v)]$, *i.e.*

$$B_{m,n}(u,v) = \begin{cases} B_{m,n}^{min} & \text{if } B_{m,n}(u,v) < B_{m,n}^{min}(u,v) \\ B_{m,n}(u,v) & \text{if } B_{m,n}^{min}(u,v) \le B_{m,n}(u,v) \le B_{m,n}^{max}(u,v) \\ B_{m,n}^{max} & \text{if } B_{m,n}(u,v) > B_{m,n}^{max}(u,v). \end{cases}$$

IV. Experiments and Discussion

To demonstrate the performance of the proposed algorithm, experiments have been conducted over a number of 512×512 monochrome images. These images are coded at different bit-rates using the baseline JPEG standard [1]. First, the proposed algorithm for blind measurement of blocking artifacts in the DCT-domain is evaluated; Then, the performance of the proposed blocking artifact reducing method is demonstrated and compared with those of existing post-processing methods. Finally, the computational complexity of the proposed algorithm and its applications are discussed.

A. Measurement of Blocking Artifact

Fig. 6 shows the map of measured visible blocking artifacts in the image *Lena* JPEG-coded at 1.0 bpp and 0.217 bpp, respectively. The visibility threshold τ is set to 0.02 as a result of experiments we performed using many images. As can be seen, at a bit rate of 0.217 bpp, the detected visible blocking artifacts are in accordance with that perceived by the human eye (See Fig. 1(b)). However, at the bit rate of 1.0 bpp, the algorithm treats actual edges in the image as visible blocking artifacts. It is a most challenging problem to distinguish between true edges and blocking artifacts in the blind measurement of blocking artifacts. Fortunately, in our post-processing algorithm, the primary edges in the image are first detected (See Fig. 3) so that they are not destroyed by mistake.

TABLE I

MEASURE OF BLOCKING ARTIFACTS OF JPEG-CODED IMAGES.

Image	JPEG coded bit rate (bits/pixel)											
	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.75	1.0	1.5	2.0
Mandrill	15.6725	12.3556	9.2143	6.3455	3.9984	3.1289	2.8064	1.9174	0.6803	0.3649	0.0062	0.0053
Peppers	16.8721	10.3115	4.2813	3.0030	2.3309	1.8415	1.3752	1.1451	0.3540	0.1576	0.0128	0.0113
Zelda	12.4376	5.2844	2.9831	1.9571	1.3431	0.8154	0.6460	0.5306	0.2004	0.0301	0.0128	0.0105
Goldhill	15.9662	8.9181	5.9695	4.4884	2.0828	1.4991	1.3069	1.0954	0.2528	0.1313	0.0111	0.0098
Bridge	19.9025	12.7618	9.8781	8.6188	4.8995	4.0530	2.1007	2.0917	0.7223	0.2957	0.0121	0.0096
Barbara	21.0135	15.8596	14.1304	6.3170	3.4481	2.5484	2.2649	1.8571	0.8301	0.4035	0.1010	0.0182
Boats	16.2819	10.1758	4.2883	3.0963	2.5450	2.0105	1.7166	1.3083	0.5935	0.3620	0.0311	0.0106

The map of measured blocking artifacts can also be combined using (14) to compute the parameter Θ, which indicates the overall picture quality with respect to the blocking effect. Fig. 7 shows the relation between the proposed measure and the corresponding bit rate for the image *Lena*. It can be seen that the value of Θ increases drastically for bit rates lower than 1.0 bpp, which is coincident with the subjective visual quality. Similar results were also reported in [11, 18]. More experimental results over different images are tabulated in Table I, showing that the proposed measure is robust and consistent across images.

B. Blocking Artifact Reduction

In order to evaluate the performance of the proposed technique and to compare it with other post-processing techniques, two DCT-domain post-processing techniques proposed in [13] and [14] were also implemented. The first experiment is conducted to show the processed image quality, measured in PSNR, for the image *Lena* coded at different bit rates. The results are shown in Fig. 8. It can be seen that the zero-masking method proposed in [13] only improves the PSNR marginally, and sometimes even has lower PSNR than the decoded JPEG image. The proposed technique achieves nearly the same PSNR as that proposed in [14], especially at bit rates lower than 0.25 bpp. However, the computational cost of the method in [14] is much higher than our technique, as discussed in the next subsection. More experimental results over different images are tabulated in Table II. Note that the data for the techniques suggested in the H.263 standard [4] and the MPEG4 standard [7], and the POCS technique proposed in [8] are copied from [14] for comparison.

TABLE II

COMPARISON OF PSNR FOR DIFFERENT POST-PROCESSING TECHNIQUES

Image	Bit rate	PSNR(dB)						
	(bpp)	Decoded	H.263	MPEG4	POCS	Method in [13]	Method in [14]	Proposed
Lena	0.217	29.47	30.20	30.02	30.23	29.56	30.32	30.27
Mandrill	0.300	22.05	22.35	22.15	22.44	21.92	22.49	22.39
Goldhill	0.227	27.90	28.50	28.31	28.46	27.85	28.51	28.40
Peppers	0.221	29.21	30.02	30.04	29.85	29.39	29.95	29.93
Airplane	0.240	28.72	29.34	29.32	29.34	28.41	29.39	29.33

For the subjective evaluation of the proposed technique, two examples are shown in Fig. 9 and Fig. 10, respectively. In Fig. 9, an enlarged part of the image *Lena* JPEG-coded at 0.217 bpp and the corresponding results post-processed by different techniques are presented. Fig. 10 shows an enlarged part of the image *Peppers* JPEG-coded at 0.221 bpp and the results post-processed by different techniques. In the images processed by the zero-masking method in [13], blocking artifacts are not sufficiently suppressed. Also the distortion at edge areas introduced by the method can be perceptually noticed. The visual quality of images processed by the proposed technique is hardly distinguishable with that by the postfiltering method in [14], though the PSNR values of the former images are a little bit lower than that of the latter as indicated in Table II.

C. Discussion

The simulation results show that the PSNR achieved by our technique is slightly lower than that obtained by the DCT-domain post-filtering method in [14]. However, the computational complexity of the latter method is much higher than ours. According to [14], each block was classified as a low-activity block or a high-activity block by computing the activities in the block. The filtering window size was set to 5×5 for the low-activity block, and 3×3 for the high-activity block. As a result, twenty-four shifted blocks had to be computed to filter one low-activity block, and eight shifted blocks for one high-activity block. Even though several fast algorithms have been developed to compute the DCT coefficients of the shifted block in the DCT-domain [16, 21, 22], the computational cost to calculate those shifted blocks in the DCT-domain is still rather high. The authors in [14]

reported that more than 70% of the blocks were classified as low-activity blocks in the test images.

By contrast, in our method, each block edge is classified as one of the three types of block edges according to the measured visibility of the blocking artifacts and the result of edge detection. For Type I block edges, there is no further processing. For Type II, the first process is to manipulate the blocks at both sides of the block edge using (18), which includes at most 16 multiplications. Finally, DCT-domain filtering is applied to those blocks with Type II or Type III block edges. Note that the filtering window size is set to 3×3 (*i.e.*, $h = 1$), which means that only eight shifted blocks are needed for each block to be processed. Obviously, the proposed algorithm is much simpler than that proposed in [14] in terms of computational complexity. The method in [13] employed an iterative approach, which included DCT, IDCT and zero-masking processes. Since it processes all blocks in the same fashion, the resulting image quality was not as good as that achieved by the proposed method. Compared to spatial-domain post-processing techniques, the DCT-domain algorithms are able to exploit the sparseness of the DCT coefficients to reduce the data volume to be processed, which makes them more favorable for real-time applications of image/video processing.

Recently, many image/video processing algorithms have been developed to process compressed image/video directly in the DCT-domain for high efficiency [23–25]. While most of these techniques have been proven to be very effective, one interesting problem is how to monitor and control the quality of the image before and after processing. For example, given two JPEG-coded image bit-streams, fast DCT-domain algorithms have been proposed [15, 26] to combine the images together to create a new JPEG bit-stream. If one of the JPEG images has severe blocking artifacts, the resulting composite picture will likely also suffer from the blocking effect. The process could be improved by employing a block artifact detection method, such as the one proposed here, prior to compositing the images. Then, a blocking effect reducing method could be pre-applied before combining the images. Similar comments apply to other applications that process images/video directly in the DCT-domain. Since only the DCT coefficients are available in applications such as these, spatial-domain post-processing techniques are hard to integrate for quality

monitoring and control. By contrast, DCT-domain processing for the blind measurement and reduction of blocking artifacts offers excellent performance and low complexity for such applications.

V. Conclusion

In this paper, we first proposed a DCT-domain method for the blind measurement of blocking artifacts which is stable and consistent across a variety of different images. A DCT-domain algorithm was also proposed to reduce blocking effects, which operates by adapting to the local measured visibility of the blocking artifacts at each block edge. To avoid excessively blurring the edge information inside the image, the algorithm first detects the primary (true) edges in the image by using the DC image, consisting of DC components from each DCT block. The proposed technique provides satisfying image quality across a wide variety of images. Due to its low computational cost, the technique can be integrated into real-time image/video applications as a method for online quality monitoring and control. The method is especially suitable for applications which process image/video directly in the DCT-domain.

VI. Appendix

According to the definition of 8×8 2-D DCT transform, Q_1 and Q_2 can be written as following, respectively.

$$Q_1(k,l) = \frac{c(k)}{2}\frac{c(l)}{2}\sum_{m=0}^{7}\sum_{n=0}^{7} q_1(m,n)\cos(\frac{2m+1}{16}\cdot k\pi)\cos(\frac{2n+1}{16}\cdot l\pi),$$

$$Q_2(k,l) = \frac{c(k)}{2}\frac{c(l)}{2}\sum_{m=0}^{7}\sum_{n=0}^{7} q_2(m,n)\cos(\frac{2m+1}{16}\cdot k\pi)\cos(\frac{2n+1}{16}\cdot l\pi),$$

where $k,l = 0,\ldots,7$, $c(0) = 1/\sqrt{2}$ and $c(k) = 1$ for $k > 0$. Since both q_1 and q_2 (defined in Section II) only have four non-zero elements, respectively, we have

$$Q_1(k,l) = \sum_{n=0}^{3}\cos(\frac{(2n+9)k\pi}{16})\cos(\frac{(2n+1)l\pi}{16}) \tag{21}$$

$$\begin{aligned} Q_2(k,l) &= \sum_{m=0}^{3}\cos(\frac{(2m+1)k\pi}{16})\cos(\frac{(2m+9)l\pi}{16}) \\ &= \sum_{n=0}^{3}\cos(\frac{(7-2n)k\pi}{16})\cos(\frac{(15-2n)l\pi}{16}) \quad (n=3-m) \end{aligned} \tag{22}$$

where the normalization factors $\frac{c(k)}{2}\frac{c(l)}{2}$ have been omitted since they do not affect our conclusion.

Observation A: As $l = 0, k \neq 0$ or $k = 0, l \neq 0$, $F_+(k,l) = Q_1(k,l) + Q_2(k,l) = 0$.

For the first case $l = 0, k \neq 0$, we have

$$\begin{aligned} F_+(k,l) &= Q_1(k,l) + Q_2(k,l) \\ &= \sum_{n=0}^{3} (\cos(\frac{(2n+9)k\pi}{16}) + \cos(\frac{(7-2n)k\pi}{16})) \\ &= 8 \times \cos(\frac{k\pi}{2})\cos(\frac{k\pi}{4})\cos(\frac{k\pi}{8})\cos(\frac{k\pi}{16}) \end{aligned} \tag{23}$$

Obviously, in (23), if $k \neq 0$, $F_+(k,l) = 0$. Similarly, it can be shown that $F_+(k,l) = 0, for \;\; k = 0, l \neq 0$.

Observation B: If $k+l$ equals to an odd number, then $F_+(k,l) = 0$. Let $l = 2p; p = 0, \ldots, 3$, then

$$\begin{aligned} F_+(k,l) &= Q_1(k,l) + Q_2(k,l) \\ &= \sum_{n=0}^{3} [\cos(\frac{(2n+9)k\pi}{16})\cos(\frac{2(2n+1)p\pi}{16}) + \cos(\frac{(7-2n)k\pi}{16})\cos(\frac{2(15-2n)p\pi}{16})] \\ &= \sum_{n=0}^{3} \{\cos(\frac{(2n+1)p\pi}{8})[(\cos(\frac{(2n+9)k\pi}{16}) + \cos(\frac{(7-2n)k\pi}{16})]\} \\ &= 2 \times \cos(\frac{k\pi}{2}) \sum_{n=0}^{3} \cos(\frac{(2n+1)p\pi}{8})\cos(\frac{(2n+1)k\pi}{16}). \end{aligned} \tag{24}$$

In (24), if k is an odd number, then $F_+(k,l) = 0$. Similarly, it can be proven that $F_+(k,l) = 0$ if k is an even number and l is an odd number, Therefore, $F_+(k,l) = 0$ if $k+l$ is an odd number.

Observation C: If $k + l$ equals to an even number, then $F_-(k,l) = Q_1(k,l) - Q_2(k,l) = 0$. A similar method to that used in *Observation B* can be employed to prove it.

Therefore, about 60% of the elements in F_+ are zero and 50% of the elements in F_- are zero.

References

[1] W. Pennebaker and J. Mitchell, *JPEG still image data compression standard*. New York: Van Nostrand, 1993.

[2] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York, NY: Chapman & Hall, 1997.

[3] "Video codecs for audiovisual services at $p \times 64kb/s$." ITU-T Rec. H.261, Mar. 1993.

[4] "Video coding for low bitrate communication." ITU-T Rec. H.263, 1998.

[5] D. Pearson and M. Whybray, "Tranform coding of images using interleaved blocks," *Proc. inst. Elect. Eng.*, vol. 131, pp. 466–472, Aug. 1984.

[6] Y. Zhang, R. Pickholtz, and M. Loew, "A new approach to reduce the blocking effect of transform coding," *IEEE Trans. on Comm.*, vol. 41, pp. 299–302, Feb. 1993.

[7] "MPEG4 video verification model version 15.0." ISO/IEC/JTC1/SC29/WG11, 1999.

[8] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 3, pp. 421–432, Dec. 1993.

[9] A. Zakhor, "Iterative procedure for reduction of blocking effects in transform image coding," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 2, pp. 91–95, Mar. 1992.

[10] R. L. Stevenson, "Reduction of coding artifacts in transform image coding," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 5, (Minneapolis, MN), pp. 401–404, Mar. 1993.

[11] F. X. Coudoux, M. Gzalet, and P. Corlay, "Reduction of blocking effect in DCT-coded images based on a visual perception criterion," *Signal Processing: Image Communication*, vol. 11, pp. 179–186, 1998.

[12] B. Jeon and J. Jeong, "Blocking artifacts reduction in image compression with block boundary discontinuity criterion," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 345–357, June 1998.

[13] B. Zeng, "Reduction of blocking effect in DCT-coded images using zero-masking techniques," *Signal Processing*, vol. 79, pp. 205–211, 1999.

[14] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 11, pp. 594–602, May 2001.

[15] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. on Selected Areas in Comm.*, vol. 13, pp. 1–11, Jan. 1995.

[16] N. Merhav and V. Bhaskaran, "Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 7, pp. 468–476, June 1997.

[17] S. A. Karunasekera and N. G. Kingsbury, "A distortion measure for blocking artifacts in image based on human visual sensitivity," *IEEE Trans. on Image Processing*, vol. 4, pp. 713–724, June 1995.

[18] Z. Wang and A. C. Bovik, "Blind measurement of blocking artifacts in images," in *Proc. IEEE Int. Conf. Image Proc.*, (Vancouver, Canada), pp. 981–984, Oct. 2000.

[19] B. A. Wandell, *Foundations of Vision*. Sunderland, MA: Sinauer Associates, Inc., 1994.

[20] A. Bovik, *Handbook of Image & Video Processing*. San Diego, CA: Academic Press, 2000.

[21] J. Song and B.-L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 767–775, Aug. 2000.

[22] S. Liu and A. C. Bovik, "Look-up-table based DCT-domain inverse motion compensation," in *Proc. IEEE Int. Conf. Image Proc.*, (Thessaloniki, Greece), Oct. 2001, accepted.

[23] B. Shen, K. Sethi, and V. Bhaskaran, "DCT convolution and its application in compressed domain," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 947–952, Dec. 1998.

[24] R. Kresch and N. Merhav, "Fast DCT domain filtering using the DCT and the DST," *IEEE Trans. on Image Processing*, vol. 8, pp. 821–833, June 1999.

[25] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction

of MPEG-2 bit streams," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 953–967, Dec. 1998.

[26] B. C. Smith and L. A. Rowe, "Compressed domain processing of JPEG-encoded images," *Real-Time Imaging*, vol. 1, pp. 3–17, Sept. 1996.

(a) 1 bit/pixel. (b) 0.217 bits/pixel.

Fig. 1. Blocking effects in the image *Lena* JPEG-coded at different bit-rates.

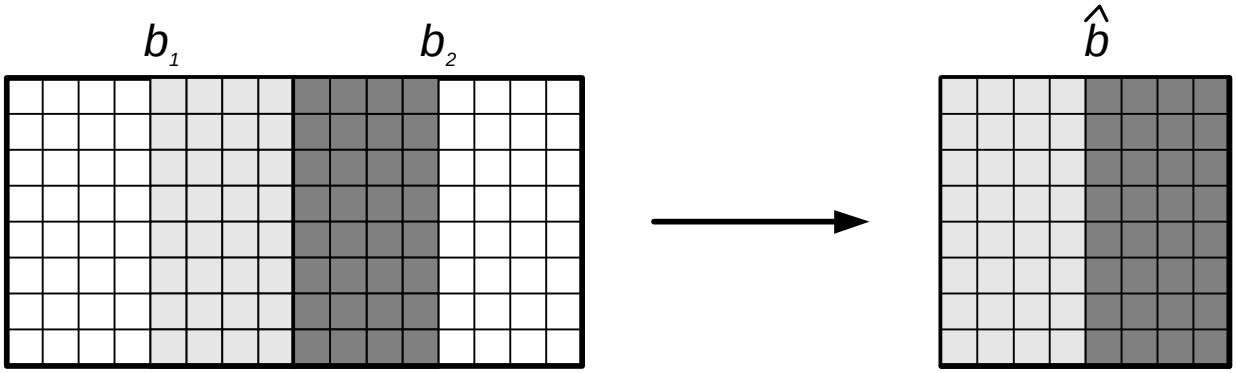


Fig. 2. Illustration of constituting the new shifted block $\hat{b}$.

Fig. 3. The result of edge detection in the image *Lena* JPEG-coded at 0.217 bpp (the threshold $T = 120$). The detected *edge blocks* are marked as black blocks .

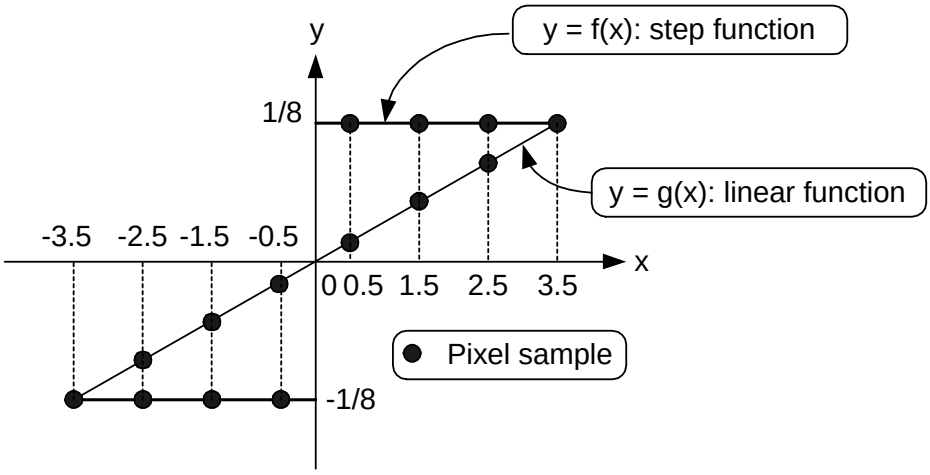


Fig. 4. Illustration of replacing the step function with a linear function in the 1-D case.

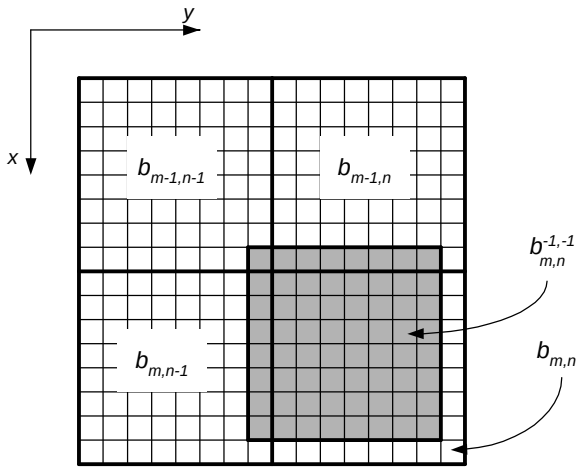
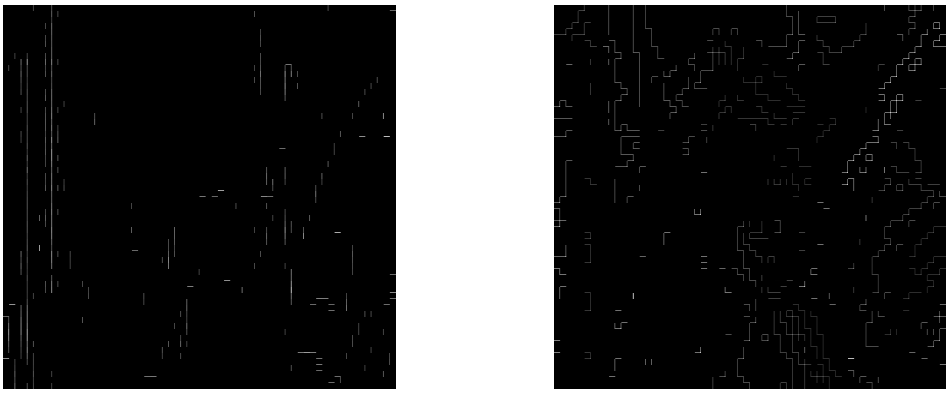


Fig. 5. Illustration of shifted block with $k = -1, l = -1$.



(a) 1 bit/pixel. (b) 0.217 bits/pixel.

Fig. 6. Measured visible blocking artifacts in the image *Lena* JPEG-coded at different bit-rates.

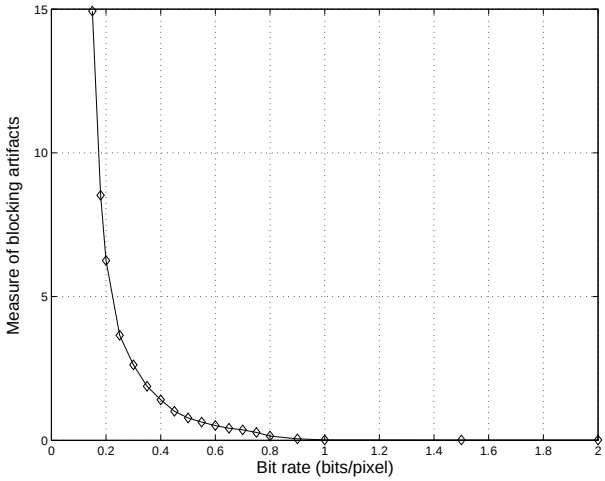


Fig. 7. Global measure of blocking artifacts at different coding bit rates for the image *Lena*.

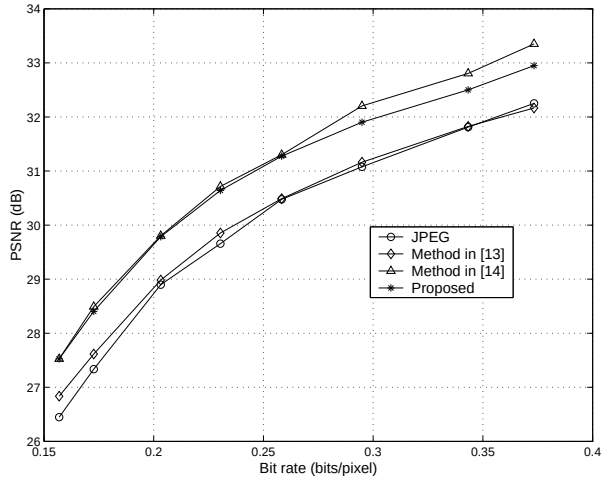


Fig. 8. PSNR versus bit rate for the JPEG-coded image *Lena*.

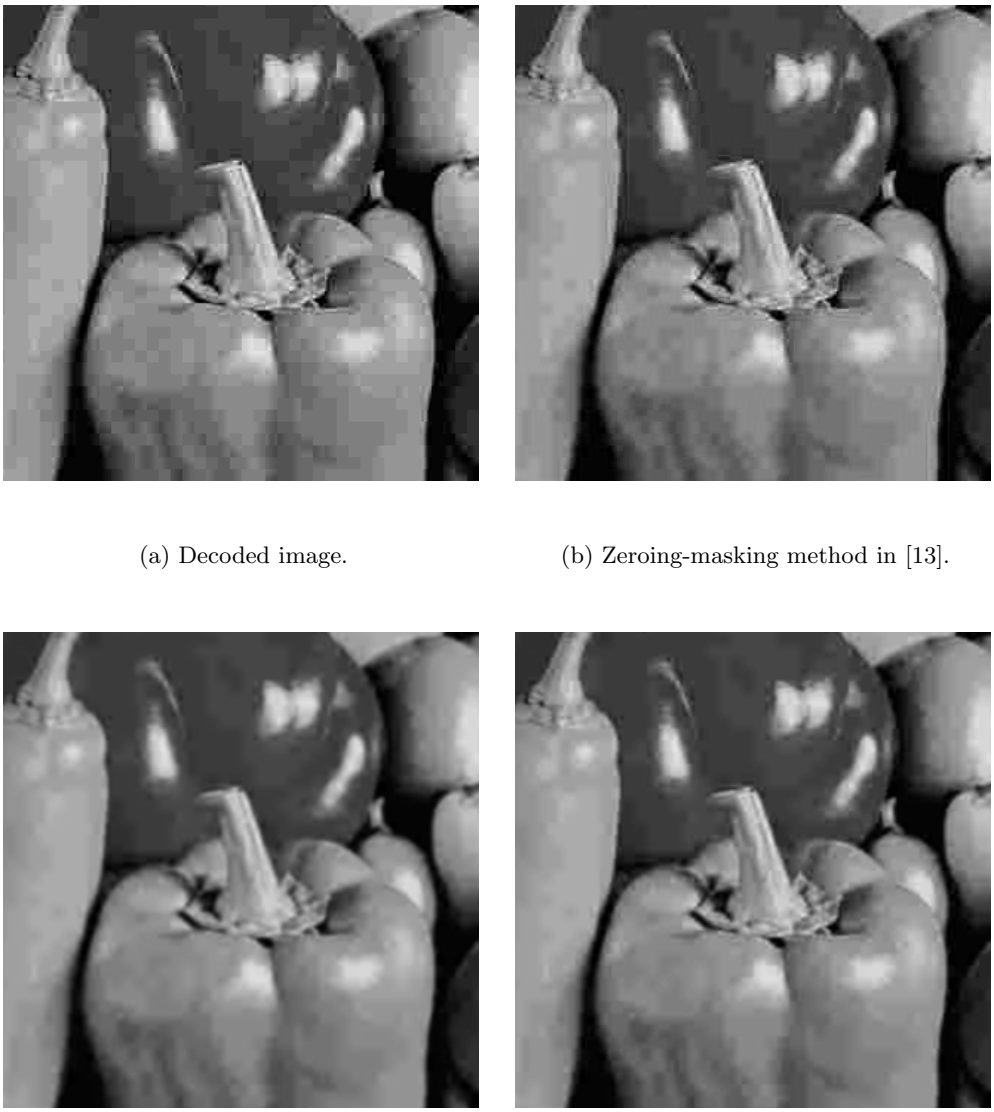
(a) Decoded image.

(b) Zero-masking method in [13].

(c) Postfiltering method in [14].

(d) Proposed technique.

Fig. 9. Subjective quality comparison of the image *Lena* JPEG-coded at 0.217 bpp and post-processed by different methods.



(a) Decoded image. (b) Zeroing-masking method in [13].

(c) Postfiltering method in [14]. (d) Proposed technique.

Fig. 10. Subjective quality comparison of the image *Peppers* JPEG-coded at 0.221 bpp and post-processed by different methods.