2014-12

Optimization Method to Reduce Blocking Artifacts in JPEG Images

UTD AUTHOR(S): Reza Pourreza-Shahri and Nasser Kehtarnavaz

Pourreza-Shahri, R., S. Yousefi, and N. Kehtarnavaz. 2014. "Optimization method to reduce blocking artifacts in JPEG images." Journal of Electronic Imaging 23(6): 063023.

Journal of Electronic Imaging

JElectronicImaging.org

Optimization method to reduce blocking artifacts in JPEG images

Reza Pourreza-Shahri
Siamak Yousefi
Nasser Kehtarnavaz

SPIE.
IS&T
imaging.org

Journal of Electronic Imaging 23(6), 063023 (Nov/Dec 2014)

Optimization method to reduce blocking artifacts in JPEG images

Reza Pourreza-Shahri,[a,*] Siamak Yousefi,[b] and Nasser Kehtarnavaz[a]
[a]University of Texas at Dallas, Department of Electrical Engineering, Richardson, Texas 75080, United States
[b]University of California at San Diego, Department of Ophthalmology, La Jolla, California 92093, United States

Abstract. This paper presents an optimization method to reduce blocking artifacts in JPEG images by utilizing the image gradient information. A closed-form solution is derived for the optimization method. To address the computational feasibility aspect of the large matrices involved in the closed-form solution, a sliding window approach is devised. The performance of the developed method is compared with several blocking artifacts reduction methods in the literature and also with the deblocking filter deployed in high efficiency video coding by examining the three measures of peak signal-to-noise ratio, generalized block-edge impairment metric (M_{GBIM}), and structural similarity. The comparison results indicate the effectiveness of the introduced method in particular for low bit-rate JPEG images. [DOI: 10.1117/1.JEI.23.6.063023]

Keywords: blocking artifacts reduction in JPEG images; optimization to reduce blocking artifacts; JPEG images.

Paper 14488 received Aug. 15, 2014; accepted for publication Dec. 3, 2014; published online Dec. 29, 2014.

1 Introduction

The JPEG compression standard[1] is extensively used for image compression. Since this compression is performed on image blocks, decompressed images exhibit blocking artifacts, particularly at low bit-rates. The visibility of blocking artifacts is dependent on the quantization level used in the compression. The coarser the quantization level, the more visible blocking artifacts become. There exist many postprocessing methods, i.e., after decompression is performed, for reducing blocking artifacts in JPEG images. In what follows, an overview of the existing methods is provided.

The early reduction methods involved low-pass filtering at block borders[2] as well as nonlinear filtering.[3] For instance, Reeve and Lim[4] applied a 3×3 Gaussian filter to the pixels around block boundaries to reduce blocking artifacts. Averaging in either the discrete cosine transform (DCT) domain or the spatial domain was used in a number of papers by considering the dependency of adjacent blocks in images. For example, a point-wise shape adaptive DCT was utilized by Foi et al.[5] In Ref. 6, a decompressed JPEG image was shifted both horizontally and vertically. Every shifted image was then compressed and decompressed using the same compression setting as the original compressed image. The output was considered to be the average of all the decompressed shifted images. A similar method was used by Chen et al.,[7] where a weighted average of DCTs of shifted blocks was considered. In the work in Ref. 8, the average of neighboring blocks was used to reduce blocking artifacts. Other methods involving Markov random field (MRF) have also been used for reduction of blocking artifacts. For example, Meier et al.[9] segmented decompressed images using image texture. An MRF-based enhancement was then applied to reduce blocking artifacts according to the segmented textures. The so-called projection onto convex sets (POCS) method[10,11] was also developed to find an optimal solution by searching among all the images with the same quantized DCT coefficients. In another POCS-based method,[12] N-point and $2N$-point one-dimensional DCTs were used to represent the frequency characteristics of adjacent blocks. Blocking was then reduced by suppressing high-frequency components of $2N$-point DCTs.

Kim et al.[13] used a method named offset and shift to filter blocking artifacts on block borders both horizontally and vertically. In this method, blocks were classified according to the directional activity of pixels. Depending on the class, the weights and lengths of the filters were determined which were then applied to the corresponding blocks to reduce blocking artifacts. In a similar method in Ref. 14, the wavelet transform was employed to classify blocks and to apply appropriate filters to reduce blocking artifacts. In a more recent work, Kim and Sim[15] developed a signal adaptive weighted sum of block boundary pixels to alleviate blocking artifacts in highly compressed images, where the weights were adjusted adaptively according to the directional information of local areas. Wong et al.[16] presented a general framework named hypothesis selection filter for noise removal and examined the application of JPEG artifact reduction. The framework consisted of a number of filters and a pixel classifier. The outputs of the filters were combined based on the output of the pixel classifier.

Two most recent works on blocking artifacts reduction have appeared in Refs. 17 and 18. In Ref. 17, Golestaneh and Chandler discussed blocking artifacts reduction in two stages. In the first stage, blocking artifacts were reduced via boundary smoothing and guided filtering. In the second stage, blurring and aliasing were reduced around edges via a local edge-regeneration procedure. In Ref. 18, Chang et al. presented an artifact reduction approach via using a sparse and redundant representation over a learned dictionary. The approach consisted of two steps. The first step involved

*Address all correspondence to: Reza Pourreza-Shahri, E-mail: reza.pourrezashahri@utdallas.edu

0091-3286/2014/$25.00

dictionary learning and the second step involved a regularization procedure.

Finally, it is important to mention the in-loop deblocking filter[19] for reducing blocking artifacts in the most recent video coding standard, high efficiency video coding (HEVC).[20] This filter has been shown to provide more reduction in blocking artifacts compared to the deblocking filter of the H.264/AVC coding standard.[21]

In this paper, a new method to reduce blocking artifacts is introduced together with an approach to make its computation feasible. In this method, blocking artifacts are reduced by utilizing the gradient information within an optimization framework.

The rest of the paper is organized as follows. The developed blocking artifacts reduction method as well as its computation aspect is described in detail in Sec. 2. The results and comparisons with four representative existing methods as well as HEVC are then presented in Sec. 3, followed by the conclusion in Sec. 4.

2 Optimization Method for Blocking Artifacts Reduction

Let $\mathbf{F}$ and $\mathbf{G}$ represent the original and the decompressed images, respectively, with $\mathbf{G}$ containing blocking artifacts. The goal is to estimate the unknown image $\mathbf{F}$ given the observed image $\mathbf{G}$ in such a way that blocking artifacts are reduced. Let $\mathbf{G}_h'$ and $\mathbf{F}_h'$ denote the gradient of $\mathbf{G}$ and an approximation of the gradient of $\mathbf{F}$ along the horizontal direction, and $\mathbf{G}_v'$ and $\mathbf{F}_v'$ similar gradients along the vertical direction. Given that the approximations $\mathbf{F}_h'$ and $\mathbf{F}_v'$ can be computed in advance, the original image $\mathbf{F}$ can be estimated using the following optimization equation:

$$\hat{\mathbf{F}} = \arg\min_{\mathbf{F}}\{\|\mathbf{F}-\mathbf{G}\|_F^2 + \lambda\|k_h(\mathbf{F}) - \mathbf{F}_h'\|_F^2 + \lambda\|k_v(\mathbf{F}) - \mathbf{F}_v'\|_F^2\}, \tag{1}$$

where $\|.\|_F^2$ denotes Frobenius matrix norm, λ is a weighting parameter, and k_h and k_v represent the horizontal and vertical gradient operators, respectively. In Eq. (1), $\mathbf{G}$, λ, k_h, and k_v are known quantities, whereas $\mathbf{F}_h'$ and $\mathbf{F}_v'$ are computed based on $\mathbf{G}_h'$ and $\mathbf{G}_v'$, thus making $\mathbf{F}$ the only unknown. The first term in Eq. (1) keeps the structural similarity (SSIM) between the images, whereas the second and third terms impose piecewise smoothness on block borders along the horizontal and vertical directions, respectively. The smoothness is achieved via reducing the blocking effect in the approximated gradient images $\mathbf{F}_h'$ and $\mathbf{F}_v'$. In Secs. 2.1 and 2.2, we show how to find an approximation to $\mathbf{F}_h'$ and $\mathbf{F}_v'$ and how to solve the above optimization problem in a computationally feasible manner.

2.1 *Approximation of Gradient Images*

The computation of an approximation of the gradient images $\mathbf{F}_h'$ and $\mathbf{F}_v'$ is discussed in this section. Noting that the compression treats each block independently, the blocking appears as discontinuity of pixel intensities on block borders. Therefore, the following model is considered for the horizontal gradient images $\mathbf{G}_h'$ and $\mathbf{F}_h'$:

$$\mathbf{G}_h' \approx \mathbf{F}_h' + \mathbf{B}_h', \tag{2}$$

where $\mathbf{B}'$ denotes the blocking effect in the gradient image $\mathbf{G}'$. An example of the horizontal gradient of the original and decompressed images (using the simple gradient operator $[-1 \quad 1]$) is shown in Figs. 1(c) and 1(e). As can be seen from Fig. 1(e), the blocking in the gradient image is visible on horizontal block borders whereas the image details are preserved elsewhere. Similarly, Figs. 1(d) and 1(f) exhibit $\mathbf{G}_v'$ and $\mathbf{F}_v'$ for the vertical direction. One can see that $\mathbf{B}_h'$ and $\mathbf{B}_v'$ mostly occur on block borders.

A simple linear interpolation is used here to substitute those samples of $\mathbf{G}_h'$ and $\mathbf{G}_v'$ which are located on block borders by interpolating the neighboring pixels as follows:

$$\mathbf{F}_h'(i,j) = \begin{cases} \frac{1}{2}(\mathbf{G}_h'(i,j-1) + \mathbf{G}_h'(i,j+1)) & j = \beta c \\ \mathbf{G}_h'(i,j), & \text{otherwise} \end{cases},$$

$$\text{for } 1 \le i \le M, \quad 1 \le j \le N, \quad 1 \le c \le N/\beta, \tag{3}$$

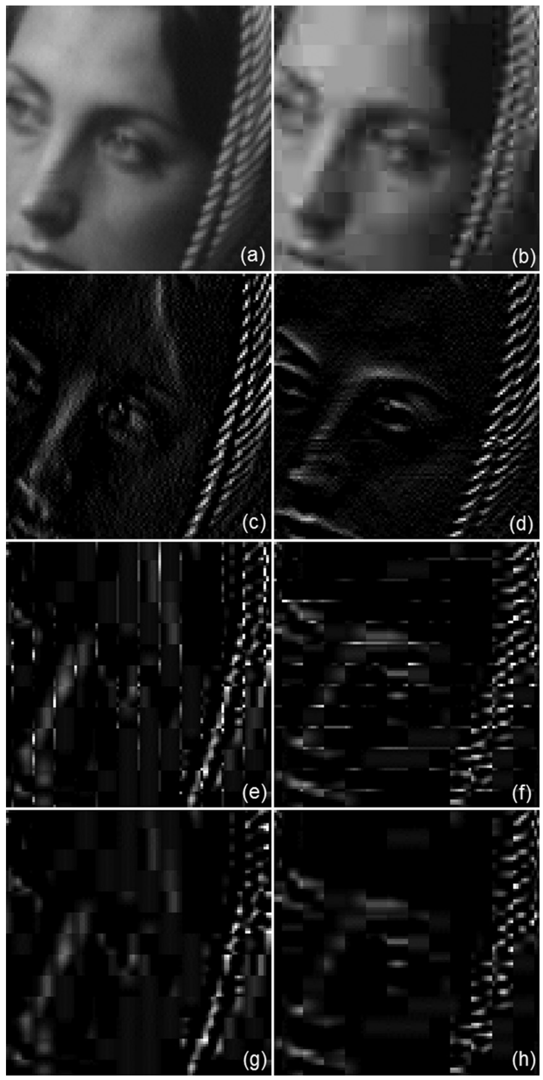

Fig. 1 Gradient approximation: (a) a part of Barbara image, (b) JPEG image with compression level 9, (c) horizontal gradient of the original image, (d) horizontal gradient of the compressed image, (e) vertical gradient of the original image, (f) vertical gradient of the compressed image, (g) approximated horizontal gradient, and (h) approximated vertical gradient.

$$\mathbf{F}_v'(i,j) = \begin{cases} \frac{1}{2}(\mathbf{G}_v'(i-1,j) + \mathbf{G}_v'(i+1,j)) & i=\beta c \\ \mathbf{G}_v'(i,j) & \text{otherwise} \end{cases},$$
$$\text{for } 1 \le i \le M, \quad 1 \le j \le N, \quad 1 \le c \le M/\beta, \tag{4}$$

where β denotes the block size, M and N are the image dimensions along the vertical and horizontal directions, respectively. It is worth mentioning that it is possible to use other interpolation schemes here. The approximations of the gradient images along the horizontal and vertical directions are shown in Figs. 1(g) and 1(h). One can see that $\mathbf{B}'$ is reduced to a great extent for both directions.

2.2 *Closed-Form Solution*

In order to avoid iterations associated with deploying a numerical optimization technique, a closed-form solution is obtained in this section. First, the optimization formulation is made more manageable by considering vector representations, i.e., by expressing $\mathbf{F}$, $\mathbf{G}$, $\mathbf{F}_h'$, and $\mathbf{F}_v'$ columnwise as vectors $\mathbf{f}$, $\mathbf{g}$, $\mathbf{f}_h'$, and $\mathbf{f}_v'$, respectively. All the vectors are of size $M*N$. As a result, the optimization formulation can be rewritten as

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{\|\mathbf{f}-\mathbf{g}\|^2 + \lambda\|\mathbf{D}_h\mathbf{f} - \mathbf{f}_h'\|^2 \quad + \lambda\|\mathbf{D}_v\mathbf{f} - \mathbf{f}_v'\|^2\}, \tag{5}$$

where $\mathbf{D}_h\mathbf{f}$ and $\mathbf{D}_v\mathbf{f}$ represent the horizontal and vertical gradients of $\mathbf{f}$, and $\mathbf{D}_h$ and $\mathbf{D}_v$ denote gradient operator matrices along the horizontal and vertical directions, respectively, which can be expressed as follows:

$$\mathbf{D}_h = \mathbf{H}_1 + \mathbf{H}_2, \tag{6}$$

$$\mathbf{H}_1 = \begin{bmatrix} [-\mathbf{I}]_{(M*N-M)\times(M*N-M)} [\mathbf{O}]_{M*M} \\ [\mathbf{O}]_{M\times(M*N)} \end{bmatrix}_{(M*N)\times(M*N)},$$

$$\mathbf{H}_2 = \begin{bmatrix} [\mathbf{O}]_{M*M} [\mathbf{I}]_{(M*N-M)\times(M*N-M)} \\ [\mathbf{O}]_{M\times(M*N)} \end{bmatrix}_{(M*N)\times(M*N)},$$

$$\mathbf{D}_v = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}_{(M*N)\times(M*N)} \tag{7}$$

with $\mathbf{I}$ representing the identity matrix and $\mathbf{O}$ the zero matrix. In the above columnwise vector representation, two elements in a vector that are $M+1$ samples apart become horizontal neighbors in the original matrix formulation. Every row of $\mathbf{D}_h$ provides the difference between two elements of a vector which are $M+1$ samples apart. Therefore, the outcome of the multiplication of $\mathbf{D}_h$ and a vector is identical to the horizontal gradient of the corresponding matrix. The last M rows of $\mathbf{D}_h$ are zero since they correlate with the most right column of the matrix that does not have neighbors on the right hand side. Similarly, every row of $\mathbf{D}_v$ provides the difference between two sequential elements of a vector. Since the lowest row of the matrix does not have elements underneath, similar to $\mathbf{D}_h$, all the elements of $\mathbf{D}_v$ that correspond to the lowest row become zero. These elements are located on those rows of $\mathbf{D}_v$ whose indices are divisible by M.

To compensate for a possible blurring effect in fine texture regions, two weights w_h and w_v are introduced to reflect the characteristics of the horizontal and vertical textures. As a result, the optimization formulation is restated as follows:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{\|\mathbf{f}-\mathbf{g}\|^2 + \lambda\|\mathbf{W}_h(\mathbf{D}_h\mathbf{f} - \mathbf{f}_h')\|^2 + \lambda\|\mathbf{W}_v(\mathbf{D}_v\mathbf{f} - \mathbf{f}_v')\|^2\}, \tag{8}$$

where $\mathbf{W}_h$ and $\mathbf{W}_v$ are diagonal matrices of size $(M*N)\times(M*N)$ with the terms w_h and w_v appearing along their diagonal elements. Higher values of w_h and w_v lead to more local smoothing in the outcome and vice versa. Since it is desired to have less smoothing in fine texture regions, these parameters can be set inversely proportional to the smoothness characteristics of local textures according to the following directional texture smoothness measures:

$$\mathbf{W}_h = (\mathbf{I}+\alpha\,\mathrm{diag}(\|\mathbf{f}_h'\|))^{-1} \quad \mathbf{W}_v = (\mathbf{I}+\alpha\,\mathrm{diag}(\|\mathbf{f}_v'\|))^{-1}. \tag{9}$$

Now to solve the optimization problem, Eq. (8) is expanded to

$$\mathbf{f}^T\mathbf{f} + \mathbf{g}^T\mathbf{g} - 2\mathbf{g}^T\mathbf{f} + \lambda\mathbf{f}^T\mathbf{D}_h^T\mathbf{W}_h^T\mathbf{W}_h\mathbf{D}_h\mathbf{f} + \lambda\mathbf{f}_h'^T\mathbf{W}_h^T\mathbf{W}_h\mathbf{f}_h' - 2\lambda\mathbf{f}_h'^T\mathbf{W}_h^T\mathbf{W}_h\mathbf{D}_h\mathbf{f} + \lambda\mathbf{f}^T\mathbf{D}_v^T\mathbf{W}_v^T\mathbf{W}_v\mathbf{D}_v\mathbf{f} + \lambda\mathbf{f}_h'^T\mathbf{W}_v^T\mathbf{W}_v\mathbf{f}_v' - 2\lambda\mathbf{f}_h'^T\mathbf{W}_v^T\mathbf{W}_v\mathbf{D}_v\mathbf{f}.$$

By taking the derivative with respect to $\mathbf{f}$, the following closed-form optimal solution is obtained:

$$\hat{\mathbf{f}} = (\mathbf{I} + \lambda\mathbf{D}_h^T\mathbf{W}_h^T\mathbf{W}_h\mathbf{D}_h + \lambda\mathbf{D}_v^T\mathbf{W}_v^T\mathbf{W}_v\mathbf{D}_v)^{-1} (\mathbf{g} + \lambda\mathbf{D}_h^T\mathbf{W}_h^T\mathbf{W}_h\mathbf{f}_h' + \lambda\mathbf{D}_v^T\mathbf{W}_v^T\mathbf{W}_v\mathbf{f}_v'). \tag{10}$$

Once Eqs. (3), (4), (6), (7), and (9) are computed for the input vector $\mathbf{g}$, Eq. (10) can be computed to obtain $\hat{\mathbf{f}}$. By rearranging $\hat{\mathbf{f}}$ into its original $M\times N$ matrix format, the blocking-reduced image $\hat{\mathbf{F}}$ results.

Finally, to ensure that block border transitions remain soft in smooth image areas, the so-called guided filter described in Ref. 22 is applied to the above outcome as the final step of our deblocking reduction. This filter is shown to be an effective edge-preserving smoothing filter.

2.3 *Computational Aspect*

Although the above optimization solution is in closed-form, it suffers from implementation limitations. As can be seen from Eqs. (6) and (7), $\mathbf{W}_h$, $\mathbf{W}_v$, $\mathbf{D}_h$, and $\mathbf{D}_v$ are of size $(M*N)\times(M*N)$. For a medium size image, say of size 512×512, $\mathbf{W}_h$, $\mathbf{W}_v$, $\mathbf{D}_h$, and $\mathbf{D}_v$ become of size 262, 144×262, 144, which makes the computation prohibitive.

To overcome the computational aspect of dealing with such large matrices, a sliding window computation approach is devised here. The steps mentioned above can be performed within deblocking windows of size $\beta\times\beta$. The deblocking

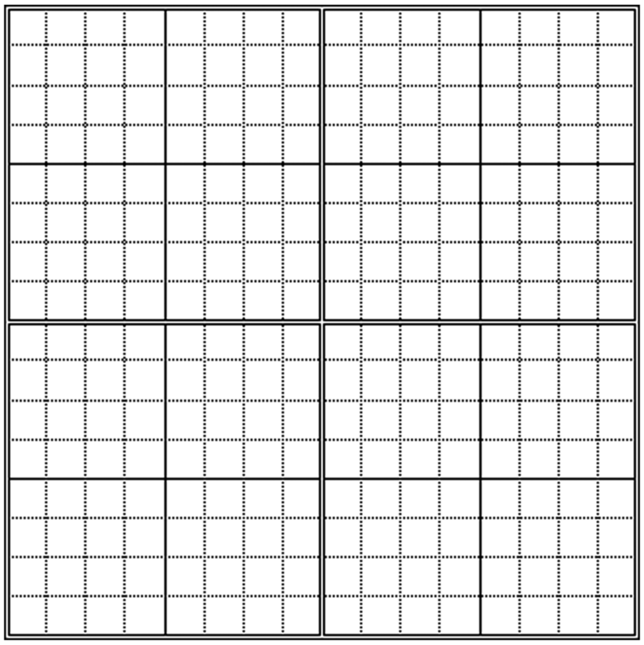

Fig. 2 Image blocks (solid lines) versus deblocking windows (double lines).

window is centered at the crossing point of four neighbors as shown in Fig. 2.

Let $\mathbf{X}$ and $\mathbf{Y}$ denote the original and the decompressed images, respectively, of this computation approach. To find the optimal $\hat{\mathbf{X}}$, the solution in Sec. 2.2 is found for every $\beta \times \beta$ window noting that $\mathbf{f}$ and $\mathbf{g}$ are now defined on such windows. Once $\hat{\mathbf{F}}$ is calculated for every window, it is placed in $\hat{\mathbf{X}}$.

3 Results and Discussion

The performance of our method was evaluated by applying it to JPEG compressed images. Seven images of Lena, Barbara, Peppers, Baboon, Boat, Zelda, and Goldhill were examined. These images are extensively used in the image processing literature and are shown in Fig. 3. These images are of size 512×512 pixels. The images were JPEG compressed at various compression or quality (Q) levels using the MATLAB compression toolbox.

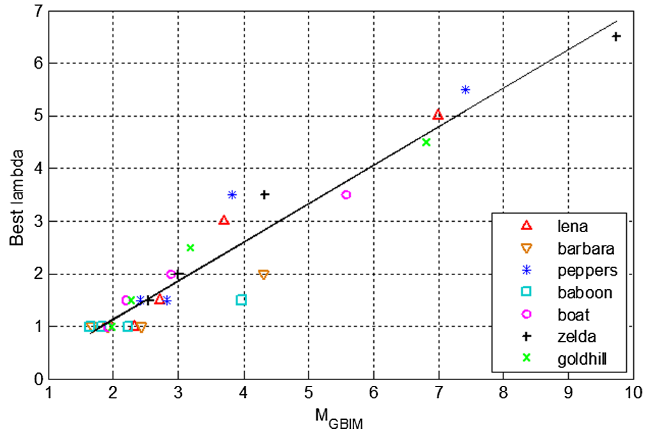

Fig. 4 Best λ versus M_{GBIM} for seven images.

3.1 *Parameter Selection*

Our first set of experimentations involved finding appropriate values for the parameters α and λ that appear in the optimization formulation as well as the radius r and regularization ε parameters that are associated with the guided filter. The parameter selection was conducted in two steps. The first step involved finding an appropriate (α, λ) parameter pair, whereas the second step involved finding an appropriate (r, ε) parameter pair.

In order to find an appropriate (α, λ) parameter pair, the three different quality measures of peak signal-to-noise ratio (PSNR), generalized block-edge impairment metric (M_{GBIM}),[23] and SSIM[24] were considered. A grid search was conducted on the $\alpha - \lambda$ space by computing these

Fig. 3 Examined images: (a) to (d) Lena, Barbara, Peppers, Baboon; (e) to (g) Boat, Zelda, Goldhill.

Table 1 Comparison of different methods in terms of peak signal-to-noise ratio (dB).

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	High efficiency video coding (HEVC)	Ours
Lena	10	0.244	30.41	30.70	30.76	31.22	30.63	31.23	**31.28**
	15	0.308	31.95	31.84	32.06	32.58	31.99	32.50	**32.67**
	20	0.363	32.96	32.50	32.82	33.38	32.68	33.33	**33.55**
	25	0.415	33.70	32.98	33.33	34.03	33.13	33.94	**34.24**
	30	0.463	34.28	33.32	33.71	34.50	34.50	34.40	**34.74**
Barbara	10	0.338	25.70	25.49	25.82	26.00	25.81	26.02	**26.22**
	15	0.444	27.05	26.43	27.10	27.26	27.03	27.26	**27.48**
	20	0.537	28.25	27.19	28.23	28.19	27.87	28.40	**28.63**
	25	0.621	29.31	27.76	29.19	29.02	28.56	29.39	**29.66**
	30	0.695	30.16	28.19	29.95	29.64	29.44	30.20	**30.49**
Peppers	10	0.247	30.13	30.47	30.58	30.71	30.32	30.92	**31.22**
	15	0.306	31.53	31.51	31.67	31.91	31.82	32.07	**32.44**
	20	0.361	32.42	32.15	32.34	32.66	32.34	32.81	**33.20**
	25	0.414	33.04	32.57	32.80	33.22	32.84	33.32	**33.74**
	30	0.462	33.53	32.86	33.09	33.62	33.81	33.71	**34.16**
Baboon	10	0.457	23.42	23.21	23.46	23.61	23.68	23.64	**23.76**
	15	0.621	24.50	23.98	24.51	24.54	24.61	24.63	**24.79**
	20	0.758	25.26	24.49	25.22	25.20	25.23	25.33	**25.50**
	25	0.885	25.89	24.90	25.80	25.78	25.77	25.92	**26.11**
	30	1.000	26.45	25.25	26.31	26.29	26.24	26.46	**26.66**
Boat	10	0.291	28.13	28.14	28.41	28.60	27.92	28.67	**28.85**
	15	0.382	29.53	29.17	29.70	29.81	29.42	29.89	**30.14**
	20	0.460	30.49	29.82	30.48	30.61	29.94	30.74	**31.01**
	25	0.532	31.23	30.30	31.09	31.25	30.52	31.39	**31.72**
	30	0.597	31.83	30.67	31.54	31.74	31.75	31.89	**32.28**
Zelda	10	0.209	32.04	32.70	32.75	33.11	32.64	**33.15**	33.10
	15	0.260	33.82	34.19	34.09	34.69	34.35	**34.65**	34.62
	20	0.303	34.94	35.05	34.79	35.62	35.05	**35.57**	35.56
	25	0.344	35.70	35.58	35.23	36.25	35.59	36.14	**36.20**
	30	0.382	36.27	35.97	35.51	36.71	36.62	36.50	**36.70**

Table 1 (*Continued*).

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	High efficiency video coding (HEVC)	Ours
Goldhill	10	0.266	28.65	28.89	28.98	29.24	28.97	29.20	**29.28**
	15	0.360	29.95	29.92	30.13	30.39	30.09	30.32	**30.49**
	20	0.445	30.87	30.62	30.90	31.17	30.78	31.14	**31.33**
	25	0.522	31.56	31.11	31.41	31.73	31.33	31.73	**31.96**
	30	0.590	32.10	31.48	31.80	32.16	32.20	32.18	**32.45**
Mean	20	0.454	30.49	30.04	30.44	30.76	30.44	30.82	**31.04**

The bold values indicate the best outcome for that particular case.

Table 2 Comparison of different methods in terms of structural similarity.

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	HEVC	Ours
Lena	10	0.244	0.899	0.919	0.924	0.922	0.925	0.927	**0.929**
	15	0.308	0.933	0.943	0.945	0.946	0.946	0.949	**0.951**
	20	0.363	0.950	0.953	0.953	0.958	0.956	0.959	**0.961**
	25	0.415	0.961	0.961	0.959	0.966	0.963	0.966	**0.969**
	30	0.463	0.967	0.965	0.963	0.970	0.970	0.970	**0.974**
Barbara	10	0.338	0.884	0.866	0.898	0.881	0.881	0.900	**0.902**
	15	0.444	0.925	0.897	0.930	0.918	0.913	0.934	**0.935**
	20	0.537	0.948	0.916	0.947	0.938	0.931	0.953	**0.954**
	25	0.621	0.960	0.927	0.956	0.951	0.942	0.963	**0.965**
	30	0.695	0.968	0.935	0.962	0.959	0.952	0.970	**0.973**
Peppers	10	0.247	0.900	0.925	0.930	0.927	0.931	0.931	**0.938**
	15	0.306	0.933	0.945	0.946	0.948	0.950	0.949	**0.955**
	20	0.361	0.949	0.956	0.954	0.959	0.958	0.959	**0.964**
	25	0.414	0.959	0.962	0.959	0.965	0.964	0.965	**0.970**
	30	0.462	0.965	0.966	0.962	0.970	0.969	0.969	**0.974**
Baboon	10	0.457	0.871	0.853	0.873	0.864	0.864	**0.879**	0.873
	15	0.621	0.917	0.891	0.914	0.906	0.907	**0.921**	0.918
	20	0.758	0.940	0.910	0.933	0.928	0.928	**0.943**	0.941
	25	0.885	0.954	0.921	0.945	0.943	0.941	**0.956**	0.955
	30	1.000	0.963	0.930	0.952	0.952	0.951	**0.964**	0.965

Table 2 (*Continued*).

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	HEVC	Ours
Boat	10	0.291	0.891	0.901	0.907	0.905	0.901	**0.910**	0.909
	15	0.382	0.930	0.930	0.936	0.936	0.934	**0.941**	0.940
	20	0.460	0.949	0.944	0.948	0.951	0.948	0.955	**0.956**
	25	0.532	0.960	0.953	0.955	0.960	0.957	0.964	**0.965**
	30	0.597	0.967	0.958	0.959	0.966	0.966	0.969	**0.972**
Zelda	10	0.209	0.898	0.927	0.930	0.927	0.933	0.933	**0.937**
	15	0.260	0.938	0.951	0.950	0.953	0.954	0.955	**0.958**
	20	0.303	0.955	0.962	0.958	0.965	0.964	0.966	**0.969**
	25	0.344	0.966	0.969	0.963	0.972	0.970	0.972	**0.975**
	30	0.382	0.972	0.973	0.965	0.976	0.975	0.975	**0.979**
Goldhill	10	0.266	0.874	0.887	0.887	0.889	0.882	**0.891**	0.887
	15	0.360	0.918	0.920	0.920	0.924	0.919	**0.927**	0.925
	20	0.445	0.941	0.938	0.937	0.943	0.939	**0.947**	0.945
	25	0.522	0.955	0.948	0.946	0.954	0.952	**0.958**	0.957
	30	0.590	0.963	0.955	0.952	0.961	0.961	0.965	**0.965**
Mean	20	0.454	0.938	0.933	0.941	0.942	0.940	0.947	**0.949**

The bold values indicate the best outcome for that particular case.

three measures for the images compressed at different Qs while not using the guided filter. It was found that α values between 0.2 and 0.3 generated the best and more or less identical outcomes, whereas the best λ varied depending on the Q value. It was found that there was a high correlation between the best λ and the blocking measure M_{GBIM} obtained from the distorted image G. Figure 4 shows M_{GBIM} versus best λ for the seven images at four different compression levels. A line was thus fitted to the samples, depicted by the solid line in Fig. 4, indicating the following equation:

$$\lambda = 0.7313 M_{GBIM} - 0.3263. \tag{11}$$

In order to find an appropriate (r, ε) parameter pair, another grid search was conducted over the $r - \varepsilon$ space based on the best (α, λ) parameter pair by computing the three measures of PSNR, SSIM, and M_{GBIM}. It was found that in 91% of the cases considered, the guided filter radius $r = 1$ generated the best outcomes. The parameter ε was found to be dependent on the compression level Q according to this line fitting equation:

$$\varepsilon = 0.0035e^{-0.0743Q} - 0.0063e^{-0.5052Q}. \tag{12}$$

Table 3 Comparison of different methods in terms of $|M_{GBIM}\text{-}1|$.

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	HEVC	Ours
Lena	10	0.244	3.279	0.487	1.187	0.380	0.584	0.600	**0.329**
	15	0.308	2.124	0.591	0.658	**0.302**	0.413	0.658	0.315
	20	0.363	1.577	0.633	0.421	**0.266**	0.318	0.624	0.297
	25	0.415	1.265	0.685	0.277	**0.252**	0.253	0.660	0.289
	30	0.463	1.076	0.687	0.200	**0.251**	0.206	0.770	0.281

Table 3 (*Continued*).

	Q	Bit rate	JPEG	Kim et al.	Kim and Sim	Chen et al.	Golestaneh and Chandler	HEVC	Ours
Barbara	10	0.338	1.777	0.620	0.849	0.342	**0.271**	0.478	0.327
	15	0.444	1.136	0.735	0.538	0.288	**0.141**	0.511	0.301
	20	0.537	0.834	0.761	0.383	0.284	**0.061**	0.477	0.265
	25	0.621	0.657	0.791	0.262	0.280	**0.006**	0.449	0.212
	30	0.695	0.567	0.795	0.194	0.285	**0.057**	0.469	0.185
Peppers	10	0.247	3.586	**0.318**	1.119	0.389	0.743	0.633	0.367
	15	0.306	2.273	0.421	0.698	**0.322**	0.515	0.749	0.387
	20	0.361	1.696	0.513	0.493	**0.291**	0.454	0.736	0.392
	25	0.414	1.375	0.572	0.386	**0.287**	0.376	0.773	0.374
	30	0.462	1.205	0.591	0.285	0.286	**0.283**	0.872	0.363
Baboon	10	0.457	1.484	0.678	0.959	0.299	**0.297**	0.654	0.440
	15	0.621	0.978	0.735	0.617	0.273	**0.139**	0.627	0.359
	20	0.758	0.755	0.763	0.446	0.277	**0.053**	0.572	0.300
	25	0.885	0.613	0.780	0.331	0.278	**0.009**	0.525	0.254
	30	1.000	0.518	0.793	0.248	0.283	**0.057**	0.487	0.214
Boat	10	0.291	2.33	0.507	0.998	0.307	0.410	0.501	**0.281**
	15	0.382	1.445	0.631	0.534	0.251	**0.249**	0.552	0.251
	20	0.460	1.059	0.687	0.320	0.220	**0.161**	0.528	0.233
	25	0.532	0.875	0.716	0.198	0.228	**0.110**	0.557	0.210
	30	0.597	0.753	0.735	0.117	0.230	**0.047**	0.608	0.194
Zelda	10	0.209	4.311	0.387	0.973	0.485	0.616	0.429	**0.095**
	15	0.260	2.590	0.551	0.394	0.366	0.456	0.499	**0.132**
	20	0.303	1.851	0.579	**0.139**	0.267	0.354	0.477	0.150
	25	0.344	1.489	0.645	**0.019**	0.224	0.285	0.575	0.152
	30	0.382	1.235	0.648	0.147	0.191	0.239	0.761	**0.144**
Goldhill	10	0.266	2.795	0.516	1.087	0.359	0.537	0.480	**0.190**
	15	0.360	1.621	0.664	0.556	**0.253**	0.311	0.570	0.200
	20	0.445	1.150	0.717	0.317	0.195	**0.186**	0.538	0.191
	25	0.522	0.943	0.730	0.184	0.189	**0.123**	0.594	0.188
	30	0.590	0.807	0.749	0.096	0.180	**0.079**	0.660	0.173
Mean	20	0.454	1.544	0.640	0.475	0.282	0.269	0.590	**0.258**

The bold values indicate the best outcome for that particular case.

3.2 Comparison Results

The performance of the developed method was assessed by computing the three measures of PSNR, SSIM, and M_{GBIM} for different compression levels while using the parameters indicated above. In addition, a comparison was carried out with four representative methods appearing in the literature (Chen et al.,[7] Kim et al.,[13] Kim and Sim,[15] and Golestaneh and Chandler[17]) that have been shown to generate superior performance over other existing methods. In addition, our method was compared to the HEVC deblocking filter. The HEVC deblocking filter was applied to all block borders in two passes, horizontal and vertical. Moreover, the

Fig. 5 Visual examination of different methods, from top to bottom: original image, compressed at $Q = 10$, Kim et al., Kim and Sim, Chen et al., Golestaneh Chandler, HEVC, and our method.

blocking strength was set equal to 2 as all the blocks were encoded in intramode. The HEVC quantization level (Q') was chosen to be

$$Q' = \begin{cases} 50 - Q, & Q \leq 50 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

as this level was found to generate the best outcomes.

All the above methods were applied to the seven images compressed at 10 to 30 compression levels in steps of 1. PSNR, SSIM, and $\mathrm{M_{GBIM}}$ were computed for every compressed image. Note that according to Ref. 23, $\mathrm{M_{GBIM}}$ values closer to one represent better quality in terms of the visibility of blocking. Considering that values above one represent blocking artifacts and values below one represent over-smoothing on block borders, $|\mathrm{M_{GBIM}} - 1|$ was used here instead of $\mathrm{M_{GBIM}}$. All the above methods produced $\mathrm{M_{GBIM}}$ values greater than one except for the Kim et al. method.

The comparison results are provided in Tables 1–3 for the seven images at different Qs. Furthermore, for closer visual examination, parts of the images as well as close-ups of the Zelda image are displayed in Figs. 5 and 6, respectively.

3.3 *Discussion*

Although PSNR is a widely used error measure between an original image and its restored version, it does not adequately represent the subjective quality of a restored image. That is why we also considered the structural similarity measure SSIM, and the blocking visibility measure $\mathrm{M_{GBIM}}$.

As can be seen from Tables 1 and 2, our method outperformed all the other methods in terms of PSNR and SSIM. The results shown in Table 3 show that our method generated outcomes comparable to those obtained by the Chen et al., Kim et al., and Golestaneh and Chandler methods in terms of $\mathrm{M_{GBIM}}$. Subjectively, the difference between the outcomes of our method and the other methods can be visually observed in Figs. 5 and 6.

Finally, it is important to emphasize that our method operates in a sliding window manner. That is, regardless of the size of the input image, the optimization is always done for windows of size $\beta \times \beta$ (normally 8×8 for JPEG compressed images). This constant window size makes the computation independent of the size of the input image.

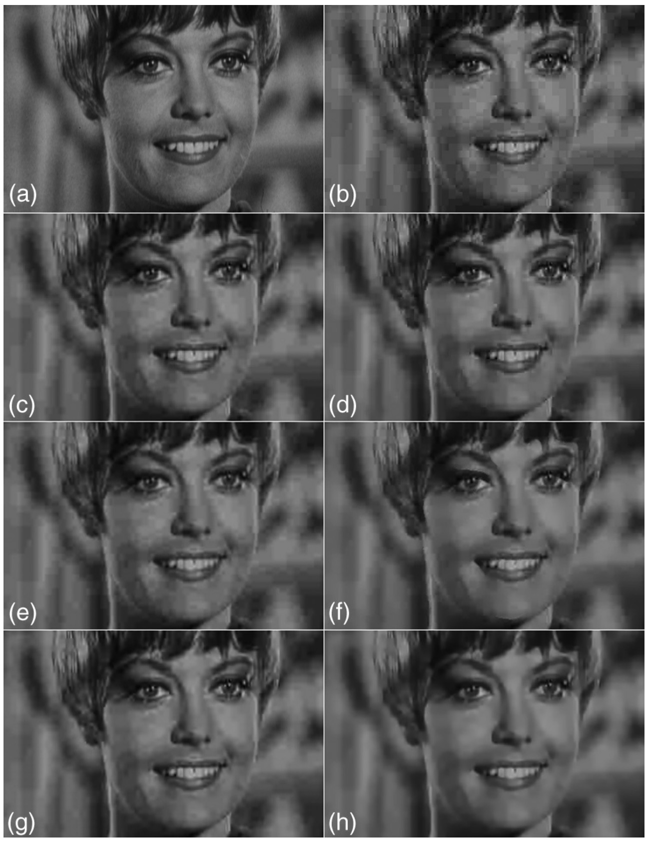

Fig. 6 Visual examination of different methods: (a) a closeup part of Zelda image, (b) compressed at $Q = 10$ or 0.209 bpp, postprocessed using, (c) Kim et al., (d) Kim and Sim, (e) Chen et al., (f) Golestaneh, (g) HEVC, and (h) our method.

4 Conclusion

A method for reducing blocking artifacts in JPEG compressed images was introduced in this paper by solving an optimization formulation based on the gradient information. The optimization involved approximating the gradient of the original image with the gradient of the decompressed image. A closed-form solution was then derived. To overcome the computation aspect of the solution due to the matrix sizes involved, a sliding window approach was introduced. The results obtained based on three distortion measures indicate that our method provides a superior performance over the existing methods, in particular in low bit-rate JPEG images.

References

1. W. Pennebaker and J. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York (1993).
2. H. C. Reeve, III and J. S. Lim, "Reduction of blocking effects in image coding," *Opt. Eng.* **23**(1), 230134 (1984).
3. B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Trans. Acoust. Speech Signal Process.* **34**(5), 1258–1268 (1986).
4. H. Reeve and J. Lim, "Reduction of blocking effect in image coding," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-83)*, Vol. 8, pp. 1212–1215, IEEE, Boston, Massachusetts (1983).
5. A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.* **16**(5), 1395–1411 (2007).
6. A. Nosratinia, "Enhancement of JPEG-compressed image by reapplication of JPEG," *J. VLSI Signal Process.* **27**(1/2), 69–79 (2001).
7. T. Chen, H. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.* **11**(5), 594–602 (2001).
8. G. Zhai et al., "Efficient image deblocking based on postfiltering in shifted windows," *IEEE Trans. Circuits Syst. Video Technol.* **18**(1), 122–126 (2008).
9. T. Meier, K. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.* **9**(3), 490–500 (1999).
10. A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.* **2**(1), 91–95 (1992).
11. Y. Yang, N. Galastanos, and A. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.* **3**(6), 421–432 (1993).
12. H. Paek, R. Kim, and S. Lee, "On the POCS-based postprocessing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits Syst. Video Technol.* **8**(3), 358–367 (1998).
13. J. Kim, M. Choi, and J. Jeong, "Reduction of blocking artifacts for HDTV using offset-and-shift technique," *IEEE Trans. Consum. Electron.* **53**(4), 1736–1743 (2007).
14. J. Kim, "Adaptive blocking artifact reduction using wavelet-based block analysis," *IEEE Trans. Consum. Electron.* **55**(2), 933–940 (2009).

15. J. Kim and C. Sim, "Compression artifacts removal by signal adaptive weighted sum technique," *IEEE Trans. Consum. Electron.* **57**(4), 1944–1952 (2011).
16. T. Wong, C. Bouman, and I. Pollak, "Image enhancement using the hypothesis selection filter: theory and application to JPEG decoding," *IEEE Trans. Image Process.* **22**(3), 898–913 (2013).
17. S. A. Golestaneh and D. M. Chandler, "Algorithm for JPEG artifact reduction via local edge regeneration," *J. Electron. Imaging* **23**(1), 013018 (2014).
18. H. Chang, M. Ng, and T. Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE Trans. Signal Process.* **62**(3), 718–728 (2014).
19. A. Norkin et al., "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1746–1754 (2012).
20. JCT-VC, "HEVC draft and test model editing (AHG2)," presented at *13th JCTVC Meeting, ISO/IEC MPEG and ITU-T VCEG* Incheon, Korea, JCTVC-M0002 (2013).
21. P. List et al., "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.* **13**(7), 614–619 (2003).
22. K. He, J. Sun, and X. Tang, "Guided Image Filtering," *IEEE Trans. PAMI.* **35**(6), 1397–1409 (2013).
23. H. Wu and M. Yuen, "A generalized block-edge impairment metric for video coding," *IEEE Signal Process. Lett.* **4**(11), 317–320 (1997).
24. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).

Reza Pourreza-Shahri is currently pursuing his PhD degree in the Department of Electrical Engineering, University of Texas at Dallas, as a member of the Signal and Image Processing Laboratory. His research interests include image and video processing, medical image analysis, and pattern recognition.

Siamak Yousefi received his PhD degree from the University of Texas at Dallas. He is currently a postdoctoral fellow at the Hamilton Glaucoma Center, University of California at San Diego, where he is conducting research on ophthalmic image and data analysis. His research areas include biomedical image analysis, pattern recognition, and machine learning. He is a member of the IEEE and ARVO.

Nasser Kehtarnavaz is a professor of electrical engineering and director of the Signal and Image Processing Laboratory with the University of Texas at Dallas, Richardson, Texas, USA. His research interests include signal and image processing, real-time processing, and biomedical image analysis. He is currently vice chair of the IEEE Dallas Section, and the editor-in-chief of *Journal of Real-Time Image Processing*. He is a fellow of IEEE, a fellow of SPIE, and a licensed professional engineer.