Enhancement of JPEG-Compressed Images by Re-application of JPEG

Aria Nosratinia (aria@utdallas.edu)
Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75083

Abstract. A novel method is proposed for post-processing of JPEG-encoded images, in order to reduce coding artifacts and enhance visual quality. Our method simply re-applies JPEG to the shifted versions of the already-compressed image, and forms an average. This approach, despite its simplicity, offers better performance than other known methods, including those based on nonlinear filtering, POCS, and redundant wavelets.

Keywords: JPEG, image compression, enhancement, postprocessing

1. Introduction

Block transform coding of images, via the Discrete Cosine Transform (DCT), has proved to be a simple yet effective method of image compression. Different implementations of this method have found widespread acceptance via international standards for image and video compression, such as JPEG and MPEG standards.

The basic approach for block-transform compression is fairly simple. The encoding process consists of dividing the image into blocks, typically of size 8×8. A block transform, typically the DCT, is applied to these blocks, and the transform coefficients are individually quantized (scalar quantization). To efficiently represent the resulting data, certain lossless compression operations are performed on the quantized data, typically consisting of a zig-zag scan of coefficients and entropy coding. A simplified diagram of this overall process is shown in Figure 1.

The block encoding process, while simple and efficient, also introduces a number of undesirable artifacts into the image; the most notable are blocking artifacts (discontinuities at the block boundaries) and ringing artifacts (oscillations due to the Gibbs phenomenon). These artifacts become more pronounced with increasing compression ratio.

A significant body of work has evolved to address the enhancement of DCT-compressed images. The problem of JPEG image enhancement, in particular, is of great interest due to the fact that the number of JPEG encoded images is currently in the millions, and will continue to rise for at least the next few years, well beyond the impending introduction of JPEG 2000. A prime example of this proliferation is

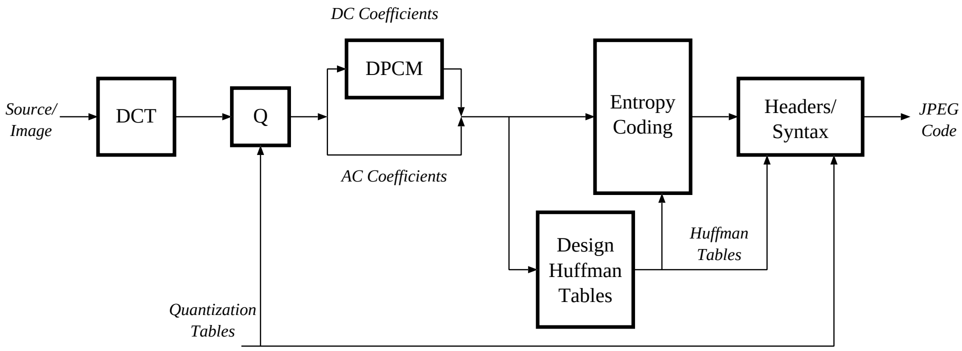

Figure 1. The JPEG encoding system

on the Internet, where numerous web pages use JPEG encoded images. Another example are the images produced by digital cameras.

In this paper we present a novel postprocessing technique for the reduction of compression artifacts in JPEG-encoded images. This approach is a significant departure from the previous signal processing methods, in that it does not specifically look at the discontinuities at block boundaries, neither does it make direct use of smoothness criteria. It uses the JPEG process itself to reduce the compression artifacts of the JPEG-encoded image. This approach is very easy to implement and, despite its simplicity, has highly competitive performance.

2. Background

Past work on the enhancement of JPEG-encoded images has largely focused on enforcing various smoothness criteria on the compressed image. The model for smoothness or continuity of the image can be deterministic or stochastic, and the enforcement of the model can vary, from regularization-based optimization to projection on convex sets (POCS) to adaptive and space-varying filters.

The earliest attempts in enhancing block-encoded images involved space-invariant filtering [1]. It was quickly discovered, however, that space-invariant filters are generally not very effective for this application; they either do not remove enough of the artifacts, or oversmooth the image.

Space-varying filters provide a more flexible framework for the reduction of compression artifacts. An early example of the application of space-varying operations to block-encoded images appeared in [2]. Space varying methods usually involve a classification step. For example, Kuo and Hsieh [3] classify image blocks according to hight or low AC activity, and apply the enhancement process only on the active blocks. The algorithm in [3] involves edge detection, and the space-varying filter is designed such that it does not smooth the edges.

In a different space-varying approach, Chou *et al.* [4] and Xiong *et al.* [5] classify the block boundaries according to the local activity in the image: If the discontinuity at a block boundary is small compared to the local energy in the image, then it is likely that the discontinuity is entirely due to quantization, therefore a strong filtering operation is performed on it. However, very large discontinuities at block boundaries are less likely to be due to quantization alone, therefore a milder smoothing operation is performed on them, so that the image edges are preserved.

A number of other related methods also depend on classification and space-varying operations, e.g. [6, 7, 8].

Another class of postprocessors utilize a reconstruction method known as Projection on Convex Sets (POCS). Usage of POCS for image reconstruction goes back to the work of Yula [9], and Yula and Webb [10]. This method is based on the well-known topological property that the nonempty intersection of a set of closed convex sets is itself a closed convex set. This intersection can be reached through repeated alternate projections onto the original sets. In the postprocessing application, one convex set consists of all original images that are quantized to the given compressed image. The other convex sets are defined to express the smoothness of the original image. The intersection of all these sets, as found by POCS, is a better approximation to the original image than the compressed image itself. POCS is elegant in design, but its convergence is critically dependent on the *a-priori* assumption that the representative sets have nonempty intersection.

One of the earliest POCS postprocessors for JPEG was proposed by Zakhor [11], [1] where the smoothness convex set consists of lowpass bandlimited images. Reeves and Eddins [12] pointed out that a non-ideal lowpass filter, like the one used in [11], is not a projection operator and therefore the algorithm, strictly speaking, cannot be classified as POCS, but is rather a constrained optimization method. Yang *et al.* [15] proposed a different convex set consisting of images with a total discontinuity across block boundaries less than a given threshold (Figure 2); this work was extended in [16] via a spatially adaptive convex set. Other work utilizing POCS for JPEG postprocessing includes [17, 18].

Constrained optimization is the basis of another family of JPEG postprocessors. A subset of this class is known as *regularization*, a

[1] This work has been subject to repeated inaccurate citation, including in [12, 13, 14, 15, 16]

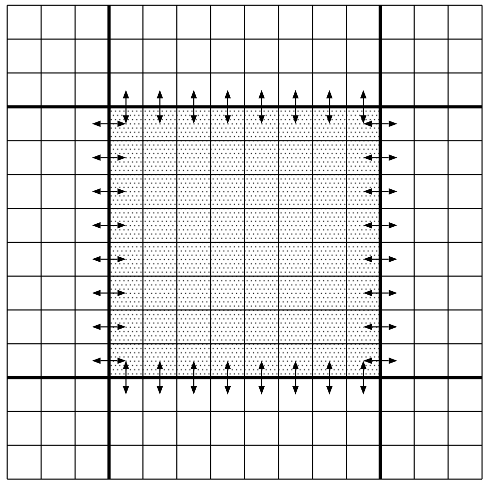

Figure 2. Traditional JPEG denoising concentrates on block discontinuities.

method to solve ill-posed inverse problems. Yang *et al.* [15] proposed a regularization scheme for a constrained least squares solution to the postprocessing problem. The constrained least squares approach arises from the desire to remain within the quantization convex set (constraint) but at the same time minimize the highpass energy of the signal (expressed as least squares). [15] used a regularization method to solve this problem. Hong *et al.* [19] applied regularization methods in the subband domain to reduce DCT artifacts in images.

Another family of postprocessors are based on sophisticated stochastic modeling of the image. All post-processors use *a priori* knowledge of the image properties. However, in the model-based approach, the *a priori* assumptions and their introduction into the algorithm are more explicit. Markov Random Fields (MRF) are among the more successful models applied to image enhancement. The algorithm of O'Rourke and Stevenson [13] applies maximum *a-posteriori* (MAP) estimation under a Markov prior, while constraining the solution to the DCT quantization hypercube. Li and Kuo [20] developed a multiscale MAP technique, again under the MRF prior. Because of the iterative procedure necessary for the generation of Markov Random Fields, MRF techniques have a high computational complexity.

Strictly speaking, compression distortion is not a random noise, in the sense that the additive distortion induced by compression, conditioned on the original (input) image, is completely deterministic. Under certain conditions, however, compression noise is uncorrelated with the quantized (output) image. Therefore, denoising techniques originally intended for random noise situations can sometimes be applied to the enhancement of compressed images. Among the most simple and effective denoising algorithms are those using the wavelet transform. Gopinath *et al.* [21] proposed an enhancement method involving the oversampled wavelet transform, in conjunction with a soft thresholding motivated by the minimax arguments of Donoho [22]. Gopinath *et al.* find the threshold based on a MMSE estimation of the quantization noise. Another version of oversampled wavelet denoising was employed by Xiong *et al.* [5]. We note that the oversampled wavelet denoising of Gopinath as well as that of Xiong are both variations on the so-called translation-invariant denoising introduced by Coifman and Donoho [23].

In the above we presented a quick overview of the main approaches to postprocessing JPEG encoded images. For the sake of brevity, some existing algorithms were not individually mentioned, among them [14, 24, 25, 26, 27, 28, 29]. These algorithms use variants or combinations of the techniques already mentioned in this section.

3. JPEG denoising through JPEG

3.1. Algorithm

We present a simple and powerful technique for the enhancement of JPEG-compressed images. Our algorithm is a dramatic departure from the known enhancement techniques, and simply consists of applying shifted versions of the JPEG compression operator to the JPEG-compressed image. The algorithm is summarized below:

1. Shift the compressed images in vertical and horizontal directions by (i, j).
2. Apply JPEG to shifted image.
3. Shift the result back, i.e. vertically and horizontally by $(-i, -j)$.
4. Repeat for all possible shifts in the range $[-3, 4] \times [-3, 4]$
5. Average all images

The quantization parameter and the quantization matrix of the JPEG, for the postprocessing purposes, is set to the same values as the compressed image. This should present no difficulties, since the header of the original JPEG image contains all necessary information. The block diagram of our postprocessing algorithm is shown in Figure 3.

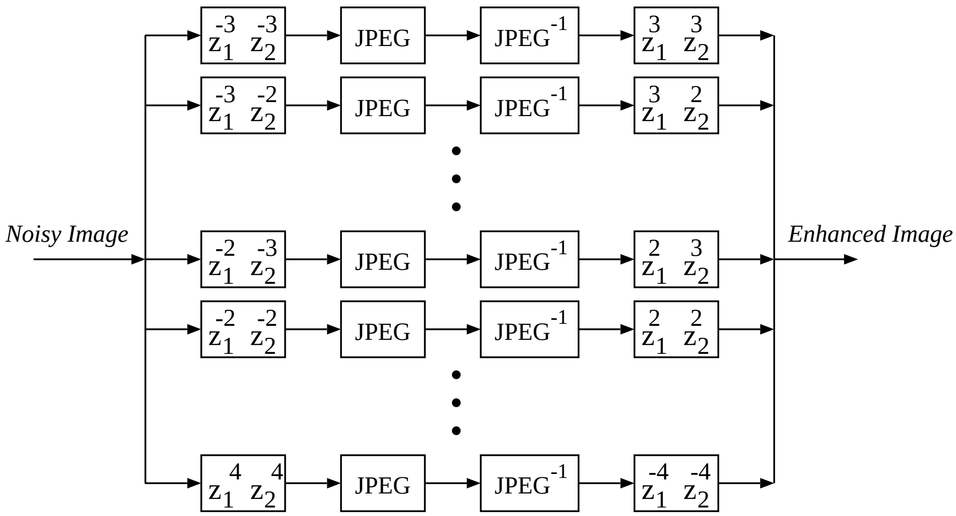

Figure 3. System diagram

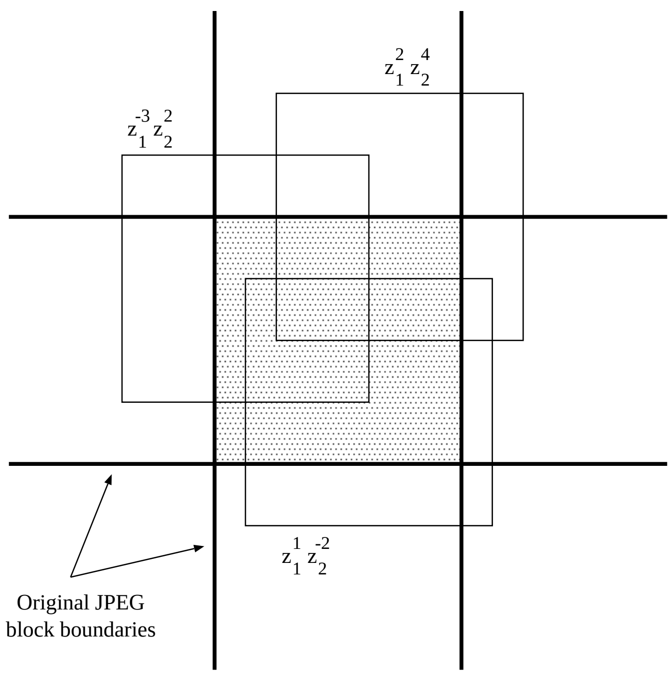

Figure 4. Our JPEG postprocessing uses various shifts of JPEG, three representative shifts shown in this figure.

The dual shifting of the image in each branch of Figure 3 essentially amounts to a net shift of the boundaries of the block encoding process, as illustrated in Figure 4. This demonstrates a basic motivation behind our algorithm: The JPEG encoding process is known to reduces the high-frequency content of the image. In other words, the high-frequency components of the image are quantized more coarsely than lower frequencies. But at the same time, high frequency components are introduced at the edges of the blocks, because these edges effectively are not "seen" in the DCT block-spectrum of JPEG. By taking various shifts of JPEG, the original block boundaries will be exposed to the frequency shaping of the JPEG encoding process, thus the magnitude of the blockiness will be reduced.

This secondary encoding process itself will produce new block boundaries, albeit smaller than the original one. One can put these new block boundaries at any given location, by controlling the shift of the secondary JPEG encoding. However, there is no reason to prefer any given location over another, therefore we average all shifts so that the secondary blockiness is diffused over all pixels. In fact, with this process, almost no blocking effects are visible in the final postprocessed image.

3.2. Relation to Known Denoising Techniques

The algorithm proposed in this paper, while in appearance and operation very different from previous approaches, in fact combines two powerful ideas from image denoising: redundant representations and the duality of quantization and denoising.

Recently, wavelet expansions have emerged as a robust and powerful tool for many signal processing applications, in particular image denoising [22]. Wavelet bases provide efficient representations of signals, which is desirable in many applications, e.g. compression. However, in other applications, such as denoising, efficiency of representation is not an object. It has been known, in fact, that redundant representations (frames) perform better than bases in denoising applications [30].

A simple explanation for the performance of redundant denoising algorithms is that the inverse transform for a frame (as opposed to a basis) is a Moore-Penrose pseudoinverse. Loosely speaking, the frame inverse transform contains averaging, which helps reduce the effect of noise. Xiong *et al.* [5], for example, harnessed the power of this technique and used redundant wavelets, along with edge classification and soft thresholding nonlinearities for their enhancement algorithm.

Our method is closely related to the oversampled (redundant) wavelet denoising techniques. The diagram in Figure 3 shows that JPEG encoding and decoding are performed successively in each branch, therefore

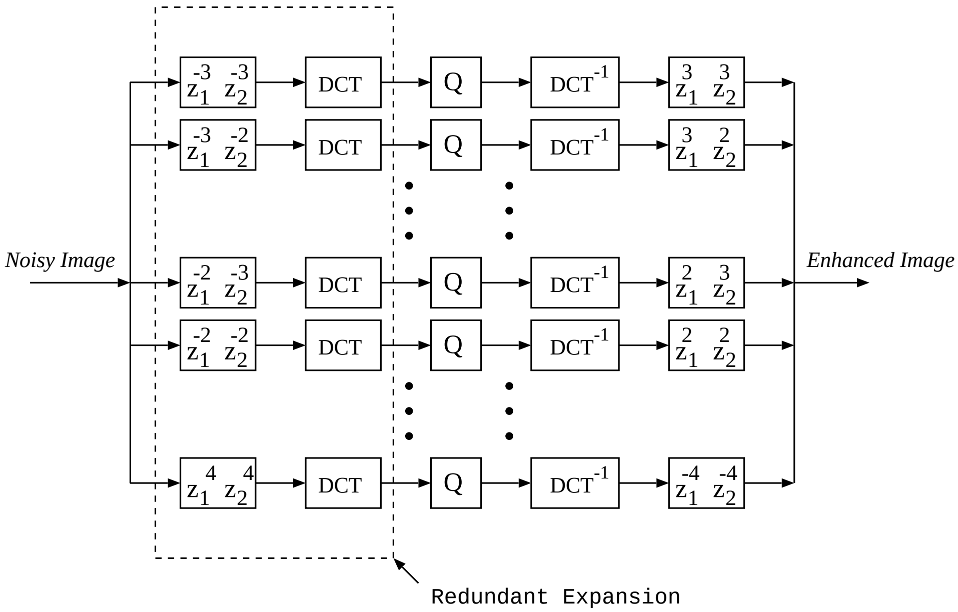

Figure 5. JPEG enhancement algorithm seen from the viewpoint of redundant expansions

the lossless parts of JPEG can be removed for our purposes, leading to the simplified diagram of Figure 5. This diagram shows that our JPEG enhancement algorithm can be viewed as a redundant denoising algorithm, where quantization plays the role of denoising nonlinearities.

This brings us to the second main idea underlying the proposed algorithm: the duality of quantization and denoising [31]. To express this relationship and its utilization in our algorithm, we look at optimal quantization, optimal MMSE denoising, and their relationship.

Assume the available observations x are a summation of a Gaussian signal s and a memoryless Gaussian noise n.

$$x = s + n \tag{1}$$

Bayesian quadratic mean estimation requires that the decorrelated components of the observed signal be scaled according to:

$$\hat{x} = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2}\, x \tag{2}$$

Therefore, optimal denoising of Gaussian signals require a linear operation on the diagonalized version of the signal.

On the other hand, optimal quantization, again for the Gaussian signals, is achieved via the inverse water-filling algorithm [32]. This argument, which is used to justify transform coding, states that optimal quantization is achieved via a diagonalizing transform. To achieve optimality, transform components below a certain energy are eliminated, and others are quantized so as to produce equal error energy.

At first sight, these two approaches may seem different, but in application they are very close. To see this point, note that the linear expression for the optimal denoising operator requires knowledge of the power of the original (uncorrupted) signal and noise. But in practice, estimating these quantities can be a rather difficult task. One can look at the inverse water-filling algorithm as an approximation of the optimal Gaussian denoising in many practical situations, because:

- For components where the signal is weaker than the noise, the linear expression (2) is approximated by zero. This corresponds to a coarse quantization that maps small signals to zero.
- For components where the signal is greater than the noise, the denoising fraction (2) is close to one. This expression can again be approximated by quantization, because when quantized distortion is much smaller than signal power, quantization itself is almost equivalent to an attenuation [33].

To summarize, this algorithm effectively uses a redundant (frame) expansion approach to signal denoising, where the frame expansion is an oversampled DCT. The denoising nonlinearities are provided by the scalar quantizers in JPEG.

3.3. Quantization Levels

The question remains: how to set the quantization levels in the secondary (denoising) JPEG? Experiments show that the best results are obtained when the secondary JPEG quantization is identical to the quantization matrix in the original image. In our experiments, we perturbed the quantization matrix by a multiplying constant α. We tested values of α both greater than and less than unity. In all such experiments, both at high and low bitrates, the quantization matrix best suited for denoising was the same as the one used for compression.

From a practical point of view, this should present no difficulties, since the quantization matrix is represented in the JPEG header of the original image, and can be extracted easily.

3.4. Image Boundaries

The shifting operation in our algorithm needs to be modified close to the image boundaries. We offer three solutions at the image boundaries:

- Symmetric extension
- Row and column replication
- Zero-shift replacement

The symmetric extension works as follows: when the shift requires that part of the existing block go outside of the image boundaries, the data is “thrown away.” When it requires data from outside the image boundaries, the image is extended symmetrically. The symmetric extension can be either odd or even at the boundary.

The replication method is similar to the symmetric extension, except that the data shifted from outside of image boundaries is simply a replication of the boundary row/column. Our simulation results were obtained using this method.

The zero-shift replacement technique for a boundary block works as follows: any shifts that can be performed without reference to pixels outside of image boundary are performed as usual. If in any branch in Figure 3 a knowledge of pixels outside the image boundary is required, then that branch will be replaced with the zero-shift branch. This means that boundary blocks will receive less smoothing than other blocks.

4. Computational Issues

At first glance the system shown in Figure 3 seems fairly involved. While a direct implementation of this system is simpler than optimization-based and model-based approaches mentioned in Section 2, it still involves 64 times JPEG compression and decompression.

A direct implementation of this system has the advantage that virtually no additional software or hardware is required. Existing JPEG code and/or hardware can be applied with the addition of some shift operators. When computational complexity becomes an issue, however, one can improve the speed of the algorithm by a number of very simple modifications:

- The simplest modification is in the branch with zero shift in Figure 3. The JPEG encoding and decoding in this branch can be removed. The reason: JPEG is an idempotent operator, in the sense that reapplication of JPEG with identical parameters to a JPEG-compressed image will result in the same JPEG-compressed image. Therefore the branch with zero shift can be replaced with an identity operator, saving the computation of one JPEG compression and decompression.

- The second modification is much more significant, and involves the removal of the lossless parts of JPEG (Figure 1). Since JPEG compression is directly followed by decompression in our algorithm, the lossless parts of JPEG play no role, and can be removed. This includes DPCM on DC values, zig-zag scan, generation of Huffman tables, entropy coding, and the generation of syntax and headers. The only parts needed are the DCT and the scalar quantization. Scalar quantization is implemented as a truncation operation, therefore the bulk of the computational complexity of our method will reside in the DCT and inverse DCT. This is a significant reduction in computational complexity.

- Finally, we note that not all shifts in Figure 3 are necessary. In fact, we observed that removing half of the shifts (in a quincunx pattern) does not significantly change the output of the algorithm. We therefore recommend it as a computational shortcut.

The operations of our algorithm depends on little else except the DCT and IDCT, so the complexity can be easily determined. The exact number of operations needed for this algorithm depends on the implementation chosen for DCT and IDCT. For example, we show below a complexity analysis of our algorithm with the 2-D DCT implementation of Feig and Linzer [34]. This is a 8×8 DCT that takes advantage of the redundancies in the two-dimensional lattice of the DCT to design an implementation with 60 multiplications and 262 additions per block. We saw above that the block with zero shift need not be recalculated during postprocessing. From among the other shifts, only one-half need be computed. Therefore we need 31 DCT and IDCT operations over the image. This gives a total of $\frac{2\times 31\times 60}{64} \approx 58$ multiplications per pixel and $\frac{2\times 31\times 262}{64} \approx 254$ additions per pixel for this algorithm. To put these numbers in perspective, the computational complexity of the proposed algorithm is roughly similar to the complexity reported for the "simplified algorithm" in [16], but is substantially smaller than that of [11].

Before leaving the subject of implementation and computation, we note that the implementation of the proposed algorithm involves only a small engineering effort (hence cost), since it needs little beyond the DCT and IDCT, and the transform already exists in each implementation of JPEG.

5. Optimal MMSE Estimation

The proposed algorithm can be viewed as a linear combination of a number of estimates of the image. To illustrate, we use the following notation. Assume that the original image is denoted by a vector $\mathbf{x}$, the JPEG-encoded image by $\hat{\mathbf{x}}$, and the denoised image by $\mathbf{y}$. Assume that the succession of JPEG encoding and decoding process is represented by the operator $\mathcal{Q}$. Using the delay operator notation D, we can write:

$$\mathbf{y} = \sum_{i,j} D^{(-i,-j)}\ \mathcal{Q}(D^{(i,j)}\,\hat{\mathbf{x}}) \tag{3}$$

Each term in the sum represents one branch of the system in Figure 3. A simple and direct extension is to replace the sum with a linear combination:

$$\mathbf{y} = \sum_{i,j} \alpha_{i,j}\ D^{(-i,-j)}\ \mathcal{Q}(D^{(i,j)}\,\hat{\mathbf{x}}) \tag{4}$$

where the coefficients $\alpha_{i,j}$ can be determined, via a training set, to make $\mathbf{y}$ an optimal MMSE estimator of the original image.

We applied this technique to a training set of images, and the resulting coefficients are shown in Figure 6. We see that, with the exception of zero shift, all shifts have almost the same coefficient. The zero-shift coefficient is significantly larger than others.

While this is a large deviation from the uniform coefficients otherwise used in this paper, we found that the optimal coefficients result in little if any additional improvement in the PSNR of the enhanced image. This leads us to believe that, in the space of coefficients, the distortion cost function must be rather flat. We therefore recommend the simpler uniform coefficient set over the optimal one.

6. Experimental Results

The results are very encouraging both in terms of PSNR and visual quality. In fact, the PSNR improvements are superior to previously reported results known to us. Table II compares the performance of the new algorithm with some results in the literature. The test image is the green component of the 512×512 pixel "Lenna." For comparison purposes, we use the three quantization tables originally introduced in [11], and also used in [5, 16]. These quantization tables are presented in Table I. Table III presents the results of the application of our algorithm to a number of 512×512-pixel test images.

We observed a slightly different JPEG PSNR compared to [5, 16] (on the order of a few hundredths of a dB) which we attribute to

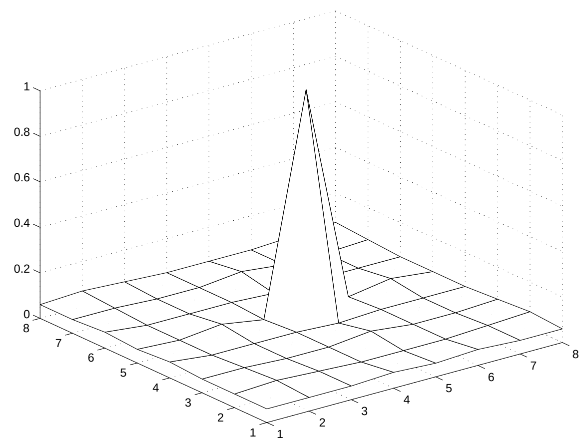

Figure 6. Coefficients for optimal combining of JPEG shifts

small differences in JPEG implementation. In order to maintain fairness despite small implementation differences, we report not the absolute PSNR, but the improvement in PSNR in each case.

Figure 7 shows part of the JPEG and enhanced image at the lower PSNR range. Note the improvement in de-blocking, as well as perseverance of edges.

7. Conclusion

In this paper we presented a novel approach to the enhancement of JPEG encoded images. Most previous approaches involve a smoothness criterion, and in one way or another focus on the discontinuities generated by the block-encoding process. In contrast, our algorithm uses the JPEG encoding itself to enhance the JPEG-compressed image. This is performed through application of various shifts of JPEG to the encoded image. The boundaries of the image can be treated in a number of ways. The computational complexity of the algorithm is smaller than the optimization-based approaches, and can be further reduced by removing the lossless parts of JPEG, as well as downsampling the shifts at which it is applied. Experimental results demonstrate excellent performance, and a large-scale reduction of both blockiness and ringing artifacts.

Table I. *Quantization tables used in experiments.*

Q_1

20	24	28	32	36	80	98	144
24	24	28	34	52	70	128	184
28	28	32	48	74	114	156	190
32	34	48	58	112	128	174	196
36	52	74	112	136	162	206	224
80	70	114	128	162	208	242	200
98	128	156	174	206	242	240	206
144	184	190	196	224	200	206	208

Q_2

50	60	70	70	90	120	255	255
60	60	70	96	130	255	255	255
70	70	80	120	200	255	255	255
70	96	120	145	255	255	255	255
90	130	200	255	255	255	255	255
120	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Q_3

110	130	150	192	255	255	255	255
130	150	192	255	255	255	255	255
150	192	255	255	255	255	255	255
192	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

Figure 7. Top: Part of JPEG encoded 512 × 512 Lenna at 26.65 dB. Bottom: Enhanced image through re-application of JPEG.

Figure 8. Top: JPEG encoded 512×512 *Lenna at 26.65 dB. Bottom: Enhanced image.*

Table II. *Improvements in PSNR on JPEG-encoded images via different algorithms, on image "Lenna" (green component).*

JPEG PSNR	Improvement in PSNR			
	POCS [16]	Wavelet [5]	Adaptive [4]	Our method
26.65	1.14	1.14	1.06	1.16
29.74	0.85	0.79	0.79	1.03
32.34	0.45	0.10	0.45	0.66

Table III. *Improvements in PSNR, through the proposed algorithm, on a number of test images at various bitrates.*

Quantization	Lenna	Mandrill	Stream	Goldhill	Barbara
Q_1	0.66	0.19	0.23	0.52	1.04
Q_2	1.03	0.23	0.42	0.74	0.89
Q_3	1.16	0.40	0.55	0.91	1.04

References

1. H. C. Reeves and J. S. Lim, "Reduction of blocking artifacts in image coding," *Optical Engineering*, vol. 23, pp. 34–37, Jan./Feb. 1984.
2. B. Ramamurthi and A. Gersho, "Nonlinear space-variant postprocessing of block coded images," *IEEE Transactions on Acoust. Speech Signal Process.*, vol. 34, pp. 1258–1267, October 1986.
3. C. J. Kuo and R. J. Hsieh, "Adaptive postprocessing for block encoded images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 4, pp. 298–304, 1995.
4. J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Letters*, vol. 5, pp. 33–35, February 1998.
5. Z. Xiong, M. Orchard, and Y. Zhang, "A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 433–437, April 1997.
6. A. Kundu, "Enhancement of JPEG coded images by adaptive spatial filtering," in *Proc. IEEE ICIP*, vol. II, (Washington, D.C.), pp. 187–191, October 1995.
7. W. E. Lynch, A. Reibman, and B. Liu, "Postprocessing transform coded images using edges," in *Proc. IEEE ICASSP*, vol. IV, (Detroit, MI), pp. 2323–2326, May 1995.
8. K. Sauer, "Enhancement of low bit-rate encoded images using edge detection and estimation," *Computer Vision, Graphics and Image Processing*, pp. 52–62, 1991.

9. D. C. Yula, “Generalized image restoration by the method of alternating orthogonal projections,” *IEEE Transactions on Circuits and Systems*, vol. 25, pp. 694–702, 1978.
10. D. Yula and H. Webb, “Image restoration by the method of convex projections: Part I - theory,” *IEEE Transactions on Medical Imaging*, vol. 1, pp. 81–94, October 1982.
11. A. Zakhor, “Iterative procedures for reduction of blocking artifacts in transform image coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, pp. 91–95, March 1992.
12. S. J. Reeves and S. I. Eddins, “Comments on iterative procedures for reduction of blocking effects in transform coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 439–440, December 1993.
13. T. P. O’Rourke and R. L. Stevenson, “Improved image decompression for reduced transform coding artifacts,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 490–499, December 1995.
14. M.-Y. Shen and C.-C. J. Kuo, “Review of postprocessing techniques for compression artifact removal,” *Journal of Visual Communication and Image Representation*, vol. 9, pp. 2–14, March 1998.
15. Y. Yang, N. Galatsanos, and A. Katsaggelos, “Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 421–432, December 1993.
16. Y. Yang, N. Galatsanos, and A. Katsaggelos, “Projection-based spatially adaptive reconstruction of block-transform compressed images,” *IEEE Transactions on Image Processing*, vol. 4, pp. 896–908, July 1995.
17. J. K. Su and R. M. Mersereau, “Post-processing for artifact reduction in JPEG-compressed images,” in *Proc. IEEE ICASSP*, vol. IV, (Detroit, MI), pp. 2363–2366, May 1995.
18. K. Y. Kwak and R. A. Haddad, “Projection-based eigenvector decomposition for reduction of blocking artifacts of DCT coded image,” in *Proc. IEEE ICIP*, vol. II, (Washington, D.C.), pp. 527–530, October 1995.
19. S.-W. Hong, Y.-H. Lee, and W.-C. Siu, “Subband adaptive regularization method for removing blocking artifacts,” in *Proc. IEEE ICIP*, vol. II, (Washington, D.C.), pp. 523–527, October 1995.
20. J. Li and C.-C. J. Kuo, “Coding aritact removal with multiscale postprocessing,” in *Proc. IEEE ICIP*, vol. I, (Santa Barbara, CA), pp. 45–48, October 1997.
21. R. A. Gopinath, M. Lang, H. Guo, and J. E. Odegard, “Wavelet-based postprocessing of low bit rate transform coded images,” in *Proc. IEEE ICIP*, pp. 913–917, 1994.
22. D. Donoho, “Denoising by soft thresholding,” *IEEE Transactions on Information Theory*, vol. IT-41, pp. 613–627, May 1995.
23. R. R. Coifman and D. L. Donoho, “Translation-invariant denoising,” in *Wavelets and Statistics, Springer Lecture Notes in Statistics 103*, (New York: Springer Verlag), pp. 125–150, 1994.
24. S. Minami and A. Zakhor, “An optimization approach for removing blocking effects in transform coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 74–81, April 1995.
25. N. C. Kim, I. H. Jang, D. H. Kim, and W. H. Hong, “Reduction of blocking artifact in block-coded images using wavelet transform,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 253–257, June 1998.

26. Y.-H. Chan, S.-W. Hong, and W.-C. Siu, “A practical postprocessing technique for real-time block-based coding system,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 4–8, February 1998.
27. S.-C. Hsia, J.-F. Yang, and B.-D. Liu, “Efficient postprocessor for blocky effect removal based on transform characteristics,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 924–929, December 1997.
28. G. Lakhani, “Improved equations for JPEG's blocking artifacts reduction approach,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 6, pp. 930–934, 1997.
29. Z. Wang and D. Zhang, “A novel approach for reduction of blocking effects in low-bit-rate image compression,” *IEEE Transactions on Communications*, vol. 46, pp. 732–734, June 1998.
30. M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, “Noise reduction using an undecimated discrete wavelet transform,” *IEEE Signal Processing Letters*, vol. 3, pp. 10–12, January 1996.
31. B. Natarajan, “Filtering random noise from determinic signals via data compression,” *IEEE Transactions on Signal Processing*, vol. 43, pp. 2595–2605, November 1995.
32. G. Davis and A. Nosratinia, “Wavelet-based image coding: an overview,” *Applid and Computational Control, Signals, and Circuits*, vol. 1, no. 1, 1998.
33. N. S. Jayant and P. Noll, *Digital Coding of waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
34. E. Feig and E. Linzer, “Discrete cosine transform algorithms for image data compression,” in *Electronic Imaging '90 East*, (Boston, MA), pp. 84–87, October 1990.

Author's Vitae

Aria Nosratinia received his B.S. in Electrical Engineering from University of Tehran, M.A.Sc. from University of Windsor, and Ph.D. in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 1996. During the academic year 1995-96, he was with Princeton University, Princeton, New Jersey. From 1996 to 1999, he was a visiting professor and faculty fellow at Rice University, Houston, Texas. In July 1999 he joined the faculty of the University of Texas at Dallas as Assistant Professor of Electrical Engineering.

Dr. Nosratinia has received the National Science Foundation Career award (2000) and the Texas Higher Education Coordinating Board Advanced Research Program award (1999). His research is in the area of digital signal processing, image processing, and coding of images and video. He has contributed to two books, *Applied and Computational Control, Signals and Circuits* and *Wavelets, Subbands, and Block Transforms in Communications and Multimedia*. His latest interests are in wireless communication of multimedia signals.