Embedded Post-Processing for Enhancement of Compressed Images

Aria Nosratinia

Department of Electrical and Computer Engineering,
Rice University, Houston, TX 77005
aria@rice.edu

Abstract

This paper presents a simple and effective post-processing method for compressed images. This work focuses on the cyclic time-variance introduced by block-based and subband transform coders. We propose an algorithm to (almost) restore stationarity to the cyclo-stationary output of the conventional transform coders. Despite a simple, non-iterative structure, this method outperforms other methods of image enhancement known to us, e.g. linear and nonlinear filtering, projection on convex sets (POCS), wavelet-based and optimization-based methods. In particular, the proposed method performs very well in suppressing both blocking and ringing artifacts. Furthermore, it admits a solution with successive approximation. The resulting embeddedness is very useful for multimedia applications such as image browsing on the world wide web.

1 Introduction

Transform coding is used extensively in image compression. Fast and efficient implementations as well as reasonable rate-distortion characteristics have contributed to the popularity of linear transforms. However, transform encoded images exhibit visually unpleasant ringing and, in the case of block transforms, blocking artifacts at lower bitrates. At least part of these artifacts can be removed through image enhancement and post-processing. In particular, JPEG image enhancement is of considerable practical interest due to its widespread application in the compression of continuous-tone images.

A significant body of work exists on the subject of compressed image enhancement. The earliest attempts involved space-invariant [1] and space-varying filters [2]. Another family of methods utilized projection on convex sets (POCS) and its variations [3, 4]. Chou et al. proposed a simple but effective nonlinear filtering approach based on continuity on the edges of the JPEG blocks [5]. Regularized least squares [6] and other optimization-based approaches [7] have also been applied to this problem. Still other works in this area include [8, 9, 10, 11, 12, 13].

Historically, reconstruction methods have relied on manipulating the behavior of encoded images at, or close to, the block boundaries of block transforms. We propose an entirely different approach to the post-processing problem. We focus on the periodic time variance introduced into the encoded image by the coding system, and our solution is based on restoring the local stationarity of the signal.

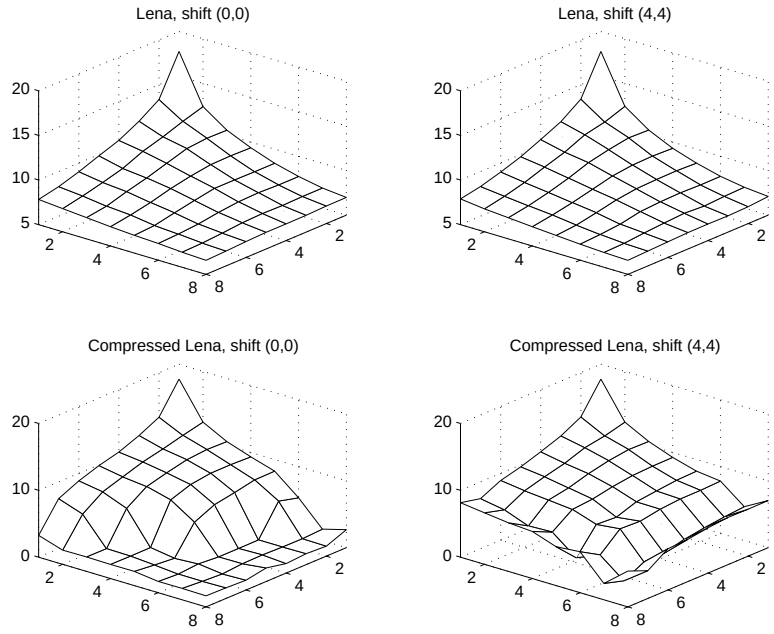

Figure 1: *Energy distribution of DCT coefficients in original (top) and compressed (bottom) "Lena" image at two different shifts. Energies shown on logarithmic scale.*

On block encoded images, the proposed algorithm exhibits excellent performance. Previously known methods perform best at very low bitrates, i.e. on high-distortion images. At moderate bitrates, the encoded image still has annoying visible imperfections, but not much blockiness. Because most enhancement methods operate on block boundaries, they lose effectiveness at moderate bitrates. In comparison, our method provides useful gain even at moderate bitrates.

Blockiness of block encoded images is only one manifestation of cyclostationarity; in that sense the proposed method is more general than its predecessors. As a result of this general approach, our method is applicable not only to block-based transform coders, but also to subband and wavelet encoded images. Research into post-processing for subband coding is fairly recent, including stochastic regularization techniques [14], and morphological filtering aided by edge detection [15]. Our method, although much simpler, performs nicely on subband encoded images and reduces the ringing artifacts significantly.

The implementation of our algorithm is quite easy, and free of computationally-heavy iterative optimization. The algorithm admits an embedded realization, a very attractive feature for multimedia applications such as browsing on the world wide web.

2 Transform Coding and Periodic Time Variance

We first motivate our approach with a simple experiment. Figure 1 shows the energy distribution of DCT coefficients for original and compressed (JPEG) "Lena" image. In each case (original and compressed) the DCT coefficients were calculated at two different block segmentations that are $(x, y) = (4, 4)$ pixels apart.

In the original image, these block spectra are virtually identical when measured at dif-

ferent shifts. In the compressed image, however, the DCT spectra computed at different shifts are distinctly different. This effect is easily explained: For the original image (no compression), one expects by symmetry that the average DCT spectrum is approximately the same at all shifts, and experiment verifies this intuition (Figure 1, top right and left). Compression eliminates many of the high frequency DCT coefficients (Figure 1, bottom left). This operation also creates discontinuities at the block boundaries that contribute high frequencies, but these discontinuities are not "seen" by the block transform at the $(0,0)$ shift, because the discontinuities will lie at the block boundaries. But if the DCT spectrum is computed at other shifts, the discontinuities will contribute to high frequency DCT energy to varying degrees (Figure 1, bottom right).

Since the blockiness of the encoded image is clearly a periodical (therefore *not* space invariant) property, one may attempt to reduce these artifacts by an operation that restores stationarity. In the following, we assume that the original signal x is stationary and is encoded using transform quantization to yield $\hat{x}$. We seek a method to approximately stationarize $\hat{x}$. Consider a typical length-N block transform $\mathbf{U}$ as follows

$$\mathbf{U} = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & \\
... & 0 & u_{11} & u_{12} & ... & u_{1N} & 0 & 0 & ... & & & \\
... & 0 & u_{21} & u_{22} & ... & u_{2N} & 0 & 0 & ... & & & \\
... & ... & ... & ... & ... & ... & ... & ... & ... & & & \\
... & 0 & u_{N1} & u_{N2} & ... & u_{NN} & 0 & 0 & ... & & & \\
 & & & ... & 0 & 0 & u_{11} & u_{12} & ... & u_{1N} & 0 & ... \\
 & & & ... & 0 & 0 & u_{21} & u_{22} & ... & u_{2N} & 0 & ... \\
 & & & ... & ... & ... & ... & ... & ... & ... & ... & ... \\
 & & & ... & 0 & 0 & u_{N1} & u_{N2} & ... & u_{NN} & 0 & ... \\
 & & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix} \tag{1}$$

The block transform of size N is the focus of our discussion, but our arguments easily extend to the more general subband decomposition, where basis functions of adjacent blocks overlap. The quantization operation can be characterized as a nonlinear matrix operator of the form

$$\mathcal{Q} = \ diag(... \ , \ q_1 \ , \ q_2 \ , \ ... \ , \ q_N \ , \ q_1 \ , \ q_2 \ , \ ... \ , \ q_N \ , \ ...) \tag{2}$$

where each $q_i(.)$ is a nonlinear quantizer function applied to the appropriate vector element. Transform coding can be characterized as follows

$$\hat{x} = \mathbf{U}^t \, \mathcal{Q} \, \mathbf{U} \, x \tag{3}$$

We denote $\mathcal{A} = \mathbf{U}^t \, \mathcal{Q} \, \mathbf{U}$ to be the transform coding operator. Since each of $\mathbf{U}^t$, $\mathcal{Q}$, and $\mathbf{U}$ are periodically time varying with period N, $\mathcal{A}$ is also periodically time varying with period N. I.e., denoting by D the one-sample delay operator (thus D^N is the N-sample delay), we have

$$D^N \, \mathcal{A} = \mathcal{A} \, D^N \tag{4}$$

Assuming that the input signal x is drawn from a stationary source, the output $\hat{x}$ will be periodically time varying with period N. By symmetry, one would expect that if the

transform coding operator $\mathcal{A}$ is applied at all possible shifts to the signal (N distinct shifts), and the outputs are averaged, the final result is a time-invariant operation. We proceed to show this formally. To create each i-shift of the transform coding operator, we shift the signal i samples, apply transform coding, and shift the result back by i

$$\hat{x}_i = D^{-i}\left[\mathcal{A}\left(D^i\, x\right)\right] \tag{5}$$

The average of all these shifts is our new signal

$$\begin{aligned} y &= \frac{1}{N}\sum_{i=0}^{N-1} \hat{x}_i \\ &= \frac{1}{N}\sum_{i=0}^{N-1} D^{-i}\,\mathcal{A}\,D^i\, x \end{aligned} \tag{6}$$

Even though each component $\hat{x}_i$ of the sum is nonstationary, the sum will be stationary if the operator $\sum D^{-i}\,\mathcal{A}\,D^i$ is time invariant. That is, if operators D^k and $\sum D^{-i}\,\mathcal{A}\,D^i$ commute for arbitrary k.

$$\begin{aligned} D^k\left(\sum_{i=0}^{N-1} D^{-i}\,\mathcal{A}\,D^i\right) &= \sum_{i=0}^{N-1} D^{k-i}\,\mathcal{A}\,D^i \\ &= \sum_{n=-k}^{N-1-k} D^{-n}\,\mathcal{A}\,D^{k+n} \\ &= \left(\sum_{n=-k}^{-1} D^{-n}\,\mathcal{A}\,D^n + \sum_{n=0}^{N-k-1} D^{-n}\,\mathcal{A}\,D^n\right) D^k \\ &= \left(\sum_{n=-k}^{-1} D^{-n}\,\mathcal{A}\left(D^N D^{-N}\right) D^n + \sum_{n=0}^{N-k-1} D^{-n}\,\mathcal{A}\,D^n\right) D^k \\ &= \left(\sum_{n=-k}^{-1} D^{N-n}\,\mathcal{A}\,D^{n-N} + \sum_{n=0}^{N-k-1} D^{-n}\,\mathcal{A}\,D^n\right) D^k \qquad (7) \\ &= \left(\sum_{m=N-k}^{N-1} D^{-m}\,\mathcal{A}\;D^m + \sum_{n=0}^{N-k-1} D^{-n}\,\mathcal{A}\,D^n\right) D^k \\ &= \left(\sum_{i=0}^{N-1} D^{-i}\,\mathcal{A}\,D^i\right) D^k \qquad (8) \end{aligned}$$

where Equation (7) results from commutability of $\mathcal{A}$ and D^N.

To summarize, if a transform coding operation is applied at all possible shifts and averaged, the resulting operator conserves stationarity. But for our purposes we don't have access to the original signal x. Our best estimate of x is the encoded version $\hat{x}$, which we will use as an approximation. Therefore, the new (almost) stationarized estimate of x given $\hat{x}$ is:

$$\tilde{x} = \frac{1}{N}\sum_{i=0}^{N-1} D^{-i}\mathcal{A}\,D^i\,\hat{x} \tag{9}$$

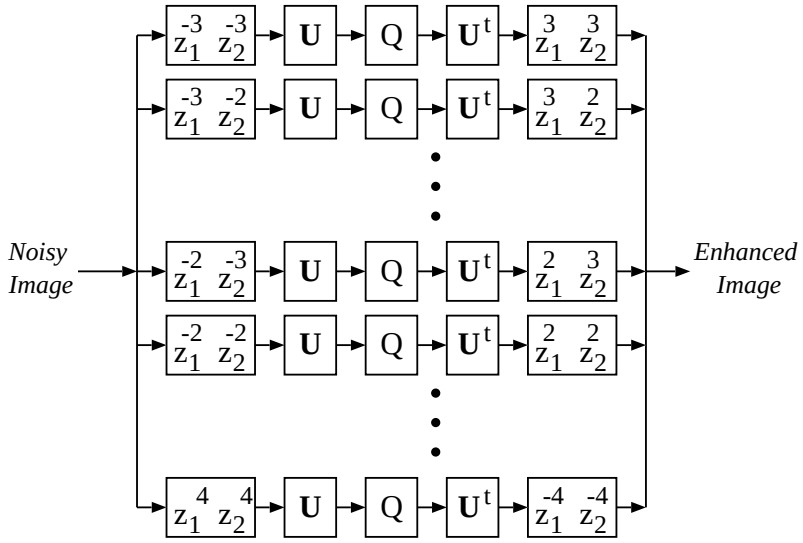

Figure 2: *System diagram*

Each term in Equation (9) is an approximation of a term in (6), except for the term at zero shift which is exact, since the transform coding operator $\mathcal{A}$ is idempotent.

For a typical block transform coder of block size 8×8 (e.g. JPEG), the direct form implementation of the system is shown in Figure 2. Blocks at image boundaries can be processed via symmetric extension.

3 Successive Approximation (Embedded) Approach

Let us take another look at Equation (9)

$$\tilde{x} = \frac{1}{N} \sum_{i=0}^{N-1} D^{-i} \mathbf{U}^t \, \mathcal{Q} \, \mathbf{U} \, D^i \, \hat{x}$$

We rewrite this in a slightly different way. Consider another linear transform $\mathbf{V}$

$$\mathbf{V} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & 0 & u_{11} & u_{12} & \dots & u_{1N} & 0 & 0 & \dots \\ \dots & 0 & u_{21} & u_{22} & \dots & u_{2N} & 0 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & u_{N1} & u_{N2} & \dots & u_{NN} & 0 & 0 & \dots \\ \dots & 0 & 0 & u_{11} & u_{12} & \dots & u_{1N} & 0 & \dots \\ \dots & 0 & 0 & u_{21} & u_{22} & \dots & u_{2N} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 0 & u_{N1} & u_{N2} & \dots & u_{NN} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

For each row in $\mathbf{U}$, there are N rows in $\mathbf{V}$, making it a redundant transform. It is not difficult to see that

$$\tilde{x} = \frac{1}{N} \mathbf{V}^t \, \mathcal{Q} \, \mathbf{V} \, \hat{x}$$

Therefore, our simple image enhancement algorithm is equivalent to nonlinear processing (using $\mathcal{Q}$) in a non-orthogonal (redundant) linear transform domain. The redundancy of $\mathbf{V}$ as defined above is by a factor of N. In other words, one can think of of $\mathbf{V}$ as application of $\mathbf{U}$ at N different sample shifts. But are all N shifts equally necessary/helpful? Obviously in the original expression (6), without complete redundancy the output will not be stationary. But Equation (9) is already approximate (because we use $\hat{x}$ instead of x), and it is instructive to see what happens if one further approximates it by discarding some of the terms in the sum.

In particular, we consider the case of an 8×8 block transform coder. Multiple levels of redundancy are obtained by changing the resolution of shifts of $\mathcal{A}$. At higher redundancies, $\mathcal{A}$ is applied at more shifts (e.g., all possible shifts are used at full redundancy). At lower redundancies, fewer shifts of $\mathcal{A}$ are computed. Figure 3 shows the positions of the application of $\mathcal{A}$ for a number of cases. To apply this on an example, we take the image "Lena," encoded by JPEG at 26.54 dB. Figure 4 shows post-processing results for this example. The horizontal axis, marked "sampling factor," indicates the level of redundancy. Sampling factor of one corresponds to the case where the operator $\mathcal{A}$ is applied only once (at zero shift). Because of the idempotency of quantization, this is equivalent to no processing at all, thus there is no gain in PSNR. Higher sampling factors represent the presence of more and more terms in Equation (9), and the shifts at which $\mathcal{A}$ is applied in each case is shown in Figure 3.

The experiment of Figure 4, as well as other similar experiments, show that in varying the redundancy of $\mathbf{V}$ (or equivalently, changing the sampling factor) there exists a point of diminishing returns. In fact, one can harvest virtually all possible gain with a sampling factor of 32 (the quincunx lattice) thus reducing computational complexity by a factor of two with almost no performance penalty.

Because each of the lattices shown in Figure 3 is a subset of the previous one, it is possible to compute successive approximations to the reconstructed image. To compute $\tilde{x} = \sum D^{-i} \mathcal{A} D^i \hat{x}$, the choice of which shifts to compute first is free, thus we can create a scan order that first goes through the shifts of the lowest resolution, then complete the next higher resolution, and so on. When the computation of $\tilde{x}_i$ at one resolution is finished, one can compute the reconstructed image at that resolution. It is straight forward to show that using accumulation, and implementing division by power-of-two with a right shift operation, virtually little or no additional computation is required for this embeddedness.

4 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 1 compares the performance of the new algorithm with some results in the literature. The test image is the 512×512 pixel "Lena." We use three different quantization tables introduced in [3]. These tables have also been used in [4, 16] for comparison purposes. We observed a slightly different JPEG PSNR compared to [4, 16] (on the order of a few hundredths of a

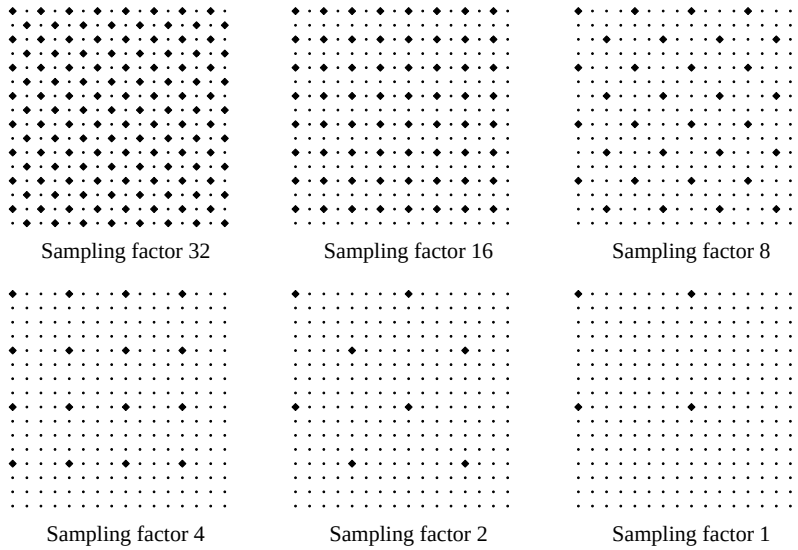

Figure 3: *Image enhancement with a* 8×8 *block transform. Each circle represents one application of operator* $\mathcal{A}$*. Sampling factor is the number of terms retained in the sum in Equation (9). Sampling factor 64 corresponds to full redundancy (not shown), where Equation (9) is fully implemented.*

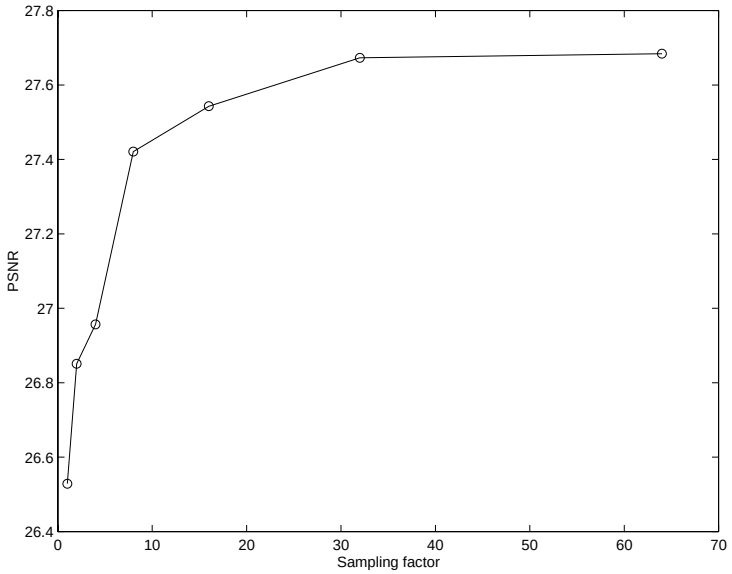

Figure 4: *Performance of the reconstruction algorithm at various degrees of redundancy (Lena, JPEG)*

dB) which we attribute to small differences in JPEG implementation. In order to maintain fairness despite small implementational differences, we report not the absolute PSNR, but the improvement in PSNR. Figure 5 shows part of the enhanced image for JPEG encoded "Lena."

We also applied this method to the output of wavelet compression algorithms. In particular we tested our algorithm with the output of the coder developed by Said and Pearlman [17]. At low bitrates (< 30 dB) significant visual improvements are observed, especially in terms of ringing artifacts. The perceptual improvements seem to be on the same order as dedicated methods designed for post-processing wavelet encoded images [14, 15]. We note that our PSNR improvements for the wavelet encoded images are typically small, on the order of 0.1 dB.

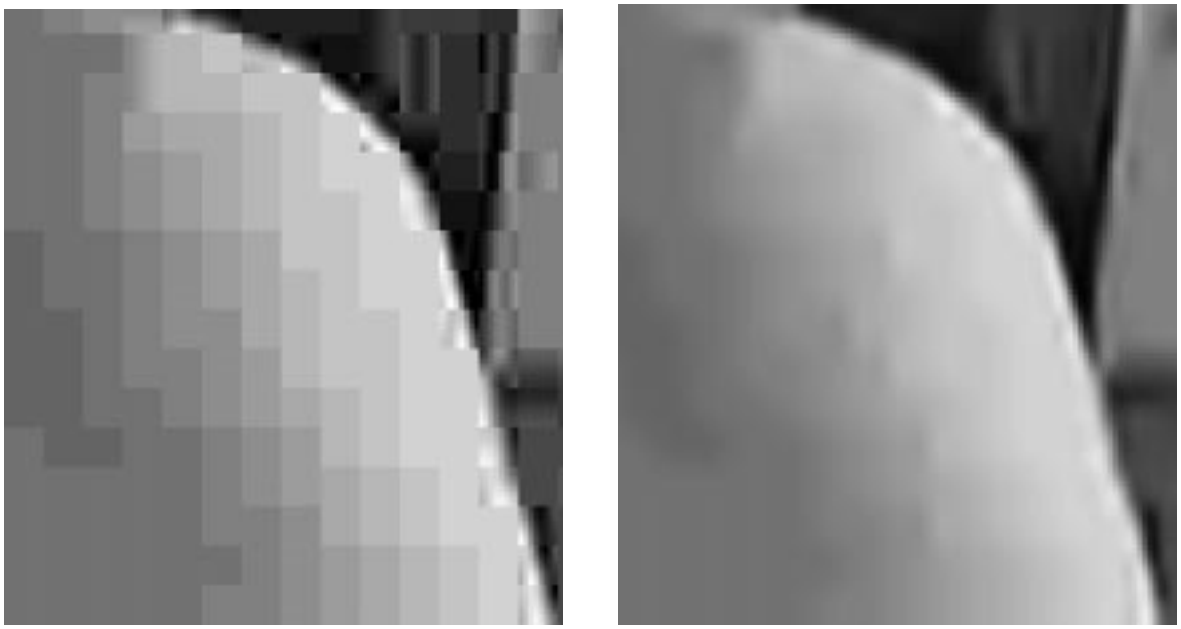

Figure 5: Left: Part of JPEG encoded 512×512 Lena. Right: Enhanced image through re-application of JPEG.

	Improvement in PSNR			
JPEG PSNR	POCS [4]	wavelet [16]	Adaptive [5]	Our method
26.65	1.14	1.14	1.06	1.17
29.74	0.85	0.79	0.79	1.00
32.34	0.45	0.10	0.45	0.65

Table 1: *Improvements in PSNR on JPEG-encoded images via different algorithms. Algorithms use identical quantization tables, given in [3].*

5 Discussion

- In equation (9), one can think of each term as an observation, and the sum as a linear estimator of the original image. In that case, one can generalize the solution as follows:

$$\tilde{x} = \frac{1}{N} \sum_{i=0}^{N-1} \alpha_i \, D^{-i} \mathcal{A} \, D^i \, \hat{x} \tag{10}$$

  where α_i are the estimation coefficients. One can find optimal α_i through training. We found that optimal α_i are nonuniform, with α_0 larger than other α_i. However, the distortion cost function in the α_i-space must be rather flat around the optimum, because the distortion induced by optimal $\{\alpha_i\}$ and uniform $\{\alpha_i\}$ are almost identical.

- We observed that the output image of our algorithm in many cases resides in the hypercube defined by scalar quantization in the transform domain. Therefore following this algorithm with a projection on the quantization hypercube, or incorporating our method into a POCS algorithm, may both prove to be of little utility. A final resolution of this issue, however, needs further work.

- Although $\tilde{x}$ often resides in the quantization convex set, individual terms in Equation (9) do not. One can project each term onto the quantization hypercube before adding them up. This approach leads to varying amounts of additional gain, over and above reported values, at the expense of computation.

6 Conclusion

This paper presents a new method for the post-processing and enhancement of transform encoded images. Transform coding is a time-varying process and thus destroys the local stationarity of the signal. Our approach is based on (partially) restoring the local stationarity. Our algorithm admits embedded implementation, and its performance is both numerically and visually competitive with other known methods. Furthermore, this method is applicable to non-block-based transforms. In the case of subband-encoded images, the perceptual quality of the images are significantly improved, along with a small improvement in the PSNR.

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] S. Miniami 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] S. H. Oguz, Y. H. Hu, and T. Q. Nguyen, "Image coding ringing artifact reduction using morphological post filtering," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, pp. 635–638, December 1998.

[16] 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.

[17] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, pp. 1303–1310, September 1996.