# Post-Processing of JPEG-2000 Images to Remove Compression Artifacts

Aria Nosratinia, *Member, IEEE*

***Abstract*— Motivated by error concealment applications, this letter proposes a method for the post-processing of JPEG-2000 compressed images at very low bitrates. The proposed method counterintuitively employs further compression to achieve image enhancement. This approach, although not widely known, is not entirely new: it is an adaptation of a technique originally designed for the removal of block-transform coding artifacts. The contribution of this work is to demonstrate its applicability to wavelet coders. In its simplest form, this algorithm uses existing system components with little or no additional hardware or software. Experimental results show a distinct reduction of ringing artifacts at very low bitrates.**

## I. Introduction

The work presented in this letter is motivated by decoder error concealment for transmission of a JPEG-2000 bitstream over a noisy channel. JPEG-2000, the recently adopted image compression standard, combines a strong compression performance with useful and practical features, such as progressive transmission. In a progressive bitstream, an unrecoverable (channel) error can render the following bits unusable. Thus, early termination due to channel errors may introduce compression artifacts at the decoder, regardless of the designated bitrate at the encoder. In this case, error concealment in the form of post-processing can improve the quality of the decoded image.

The JPEG-2000 ringing artifacts are in general more difficult to characterize and/or remove than the block-transform compression artifacts. This letter presents a method for reducing JPEG-2000 artifacts and thus improving the overall perceptual quality of the image. The work presented in this letter is related to the general method of translation-invariant denoising introduced by Coifman and Donoho [1].

In a different approach, Fan and Cham [2] performed edge reconstruction on wavelet compressed images using a model-based method. Unlike [2], the present work does not involve edge detection, edge reconstruction or optimization, and is conceptually very simple. Furthermore, this work is directed at JPEG-2000 while [2] was applied to SPIHT. The achievable gains in terms of PSNR are comparable.

## II. JPEG-2000 Post Processing

The basic mechanism behind our approach can be demonstrated by comparing the wavelet transforms of different shifts of an image. Figure 1 (a) is the wavelet transform[1] of the original Lenna image (before compression). The high-frequency wavelet coefficients have been magnified by a factor of 32 to make them more visible. In Figure 1 (b), the Lenna image has been shifted by $(1,1)$ pixels before taking the wavelet transform. These figures show an intuitive result that is easily confirmed by calculations: that the transform coefficients in these two cases have similar second-order statistics. Figure 1 (c) shows the wavelet transform of the Lenna image compressed with JPEG-2000 at 0.1 bits per pixel, clearly showing the suppression of the high-frequency coefficients because of compression. But the high frequency coefficients reappear if we look at the wavelet transform of the shifted version of the compressed image (Figure 1 (d)).

TABLE I
Variance of HL, LH, and HH bands calculated at two different shifts for original, compressed, and post-processed image.

| | | Coefficient Variances | | |
|---|---|---|---|---|
| | | Original | Compressed | Post-processed |
| HL | (0,0) shift | 17.3127 | 0.0809 | 0.1574 |
| | (1,1) shift | 17.3611 | 2.3456 | 0.3269 |
| | ratio | 1.0028 | 28.9938 | 2.0769 |
| LH | (0,0) shift | 43.1195 | 0.0898 | 1.0896 |
| | (1,1) shift | 42.1531 | 19.3274 | 3.9434 |
| | ratio | 0.9776 | 215.2272 | 3.6191 |
| HH | (0,0) shift | 8.7825 | 0.0748 | 0.0749 |
| | (1,1) shift | 8.8884 | 0.1651 | 0.0759 |
| | ratio | 1.0121 | 2.2072 | 1.0134 |

This effect is numerically demonstrated in Table I. We calculate the variance of coefficients when the wavelet transform is applied at zero shift and a non-zero shift, and use the ratio as a measure of shift-variance of the image statistics. Table I shows that coefficient variance in the original image is not sensitive to shifts, unlike for the compressed image, where the variance is very shift-dependent. This phenomenon arises because the maximally decimated wavelet transform is not shift-invariant [3]. As a result, wavelet compression introduces cyclostationarity into the image, resulting in the ringing effects visible around the image discontinuities.

To reduce or remove these artifacts, we propose to address the cyclostationarity introduced into the image by compression. Our objective is to construct at the decoder a version of the decompressed image with approximately shift-invariant statistics. One way to generate a statistically shift-invariant "compressed" image is to perform wavelet compression at all possible shifts

This work was supported in part by the National Science Foundation under grant CCR-9985171.

A. Nosratinia is with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: aria@utdallas.edu)

[1] Throughout this letter we use the Daubechies 7-9 biorthogonal wavelets.

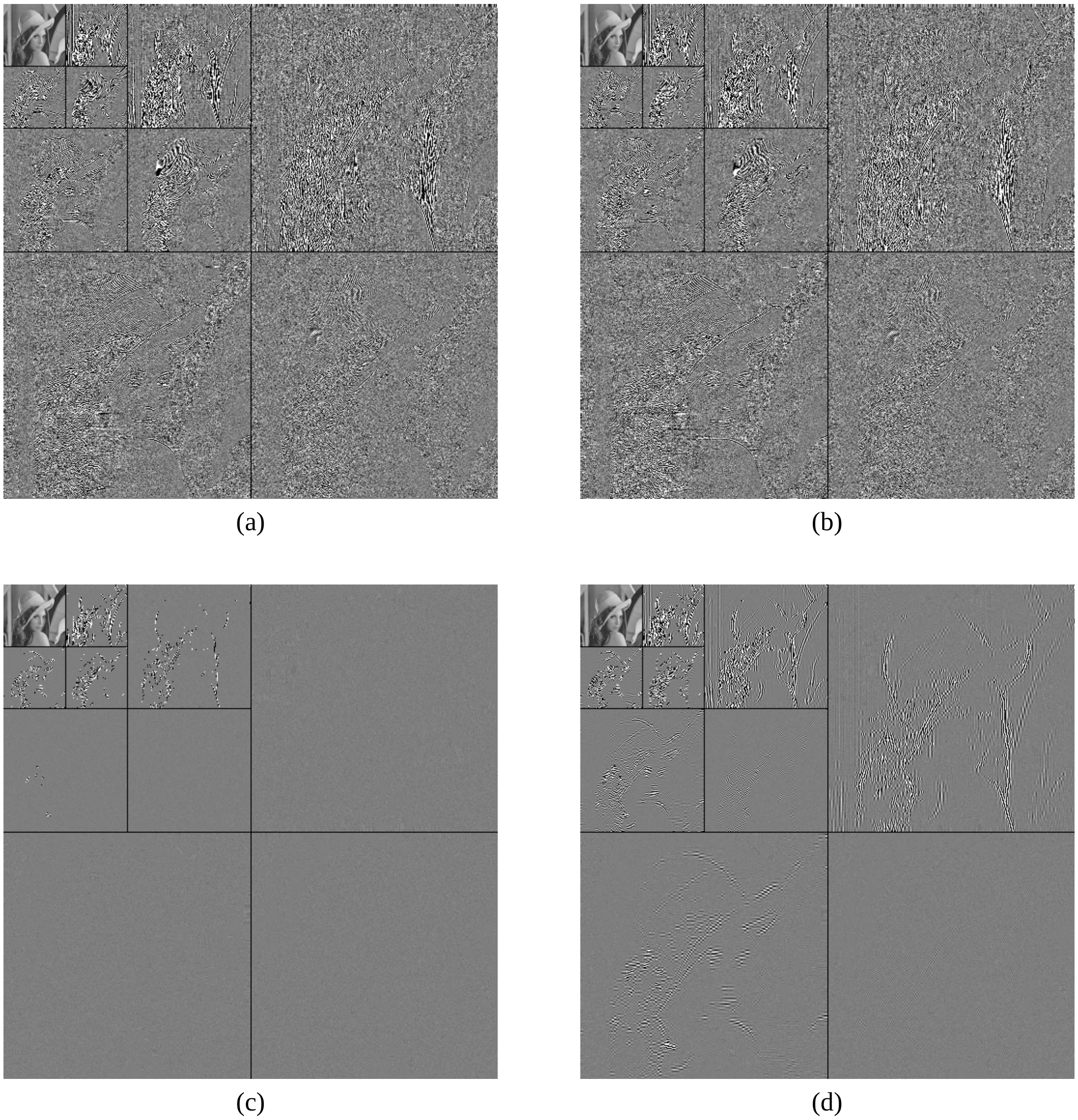



Fig. 1. (a) and (b): wavelet transform of Lenna image with shifts of (0,0) and (1,1) respectively. (c) and (d) are wavelet transforms of compressed Lenna image with shifts of (0,0) and (1,1) respectively.

and average (after adjusting for the shift). This is of course impractical for the simple reason that the decoder does not have access to the original image. However, the idea can be used in an approximate manner: We propose to apply compression to the shifted versions of the *received* image and average the results (Figure 2). This algorithm essentially adopts a methodology proposed by Nosratinia [4] for removing block-DCT compression artifacts. The main contribution of the present work is to demonstrate that this approach is applicable beyond the block-DCT compression for which it was originally proposed; it applies to wavelet coders as well. To our knowledge, this is the first demonstration of a post-processor for JPEG-2000 compression.

In our simulations, we used the Kakadu implementation of JPEG-2000 [5]. Figure 3 exhibits the effect of post-processing on the wavelet coefficients, and Figure 4 shows the postprocessing results for one of the test images. Similar results were obtained with other test images. A comparison of the variance of transform coefficients in Table I also demonstrates that the ratio of variances in the post-processed image is much closer to unity, compared to the compressed image.

We found that the best visual quality[2] is obtained when the compression ratios for the parallel branches of the algorithm are set to the same level as the originally compressed image. Under these conditions, the perceptual quality of the post-processed image is clearly better than the compressed image, but the PSNR of the post-processed image is slightly below the compressed image (on the order of $-0.1$ dB). It is possible instead to produce post-processing *gains* on the order of $0.1$ to $0.2$ dB with our method by reducing the compression ratio of the parallel branches (compared to the zero-shift branch). However, the visual results in that case are less pleasing.

The post-processed images appear slightly low-passed overall, a perception that is due to the removal of ringing from the image. It is noteworthy, however, that the image discontinuities

[2] Perceptual tests were performed by a group of graduate students at UT-Dallas.

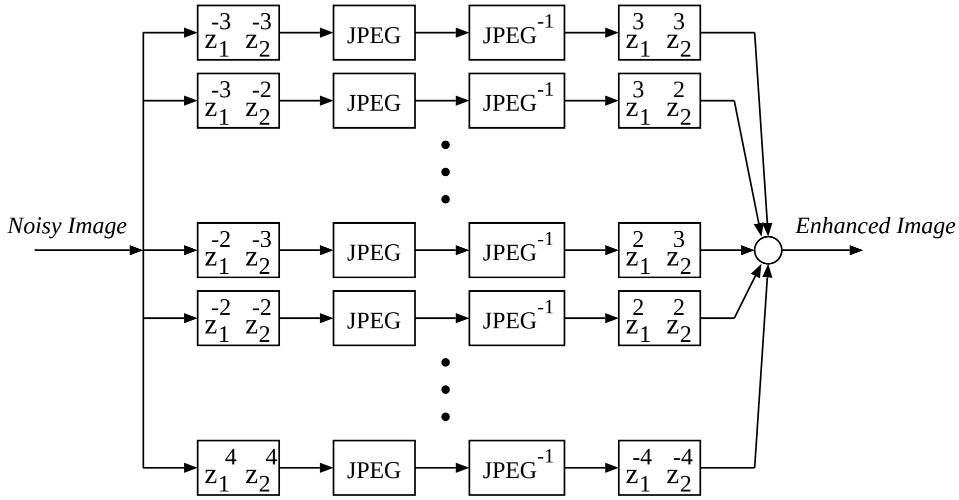
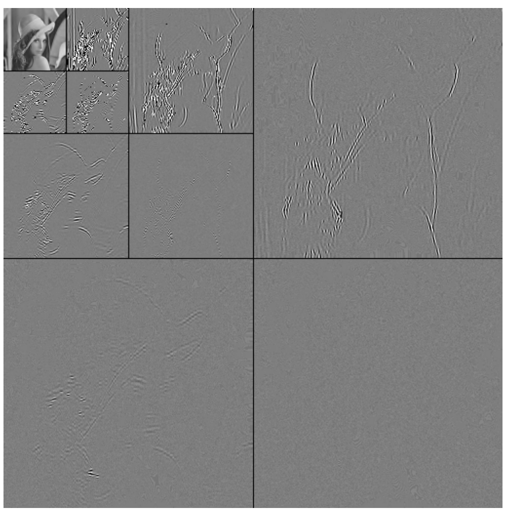
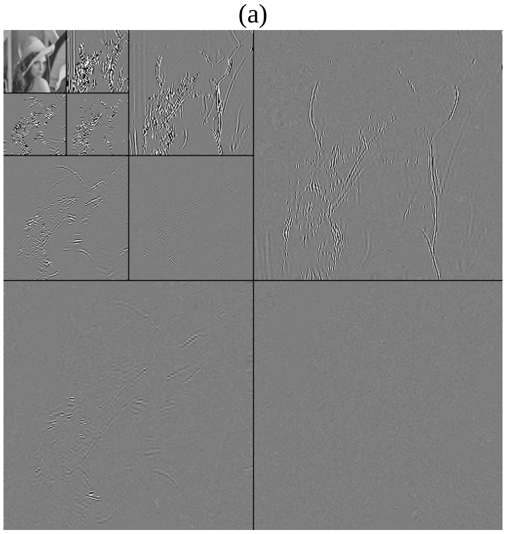



Fig. 2. Block diagram of proposed method. JPEG and JPEG$^{-1}$ denote the JPEG-2000 encoder and decoder, respectively.



(a)



(b)

Fig. 3. Wavelet transform of post-processed image with a shift of (0,0) in (a) and (1,1) in (b).

(edges) are not blurred to any significant degree.

A straight forward implementation of this technique is very simple and requires little additional hardware or software. It is enough to apply JPEG-2000 encoding and decoding to various shifts of the image. The computational complexity of this implementation can be significantly reduced by removing all reversible operations such as entropy coding (the MQ coder) [5] and bitstream generation from both the encoder and decoder. Furthermore, although Figure 2 shows 64 parallel branches (the configuration we used for our simulations), our experiments show that similar results are obtained with as few as 4 to 8 parallel branches.

## III. Conclusion

This letter introduces a method for post-processing of JPEG-2000 compressed images. In its simplest form, this method requires little additional hardware or software: it is implemented by applying JPEG-2000 encoding and decoding operations on various shifts of the image. Experimental results show that this method provides perceptual improvements in the image at very low bitrates; thus, this method can provide a degree of error concealment in progressive transmission over noisy channels.

## References

[1] R. R. Coifman and D. L. Donoho, “Translation-invariant denoising,” in *Wavelet and Statistics* (A. Antoniadis and G. Oppenheim, eds.), Lecture Notes in Statistics, pp. 125–150, Berlin: Springer Verlag, 1995.

[2] G. Fan and W.-K. Cham, “Model-based edge reconstruction for low bit-rate wavelet-compressed images,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 120–132, February 2000.

[3] M. Vetterli and J. Kovačević, *Wavelets and subband coding*. Prentice Hall, 1995.

[4] A. Nosratinia, “Denoising of JPEG images by re-application of JPEG,” *Journal of VLSI Signal Processing*, vol. 27, no. 1, pp. 69–79, 2001.

[5] D. Taubman and M. Marcellin, *JPEG 2000: image compression fundamentals, standards and practices*. Kluwer Academic Publishers, 2002.

Fig. 4. Compressed Lenna image (left) and post-processed version (right). The magnified portions demonstrate reduction in ringing artifacts.