DERINGING AND DEBLOCKING DCT COMPRESSION ARTIFACTS WITH EFFICIENT SHIFTED TRANSFORMS

Ramin Samadani, Arvind Sundararajan and Amir Said
HP Labs Palo Alto, Imaging Technologies Department

ABSTRACT

A new method, using weighted combinations of shifted transforms, is developed for deringing and deblocking DCT compressed color images. The method shows substantial deringing improvement over prior methods, maintains comparable deblocking and shows comparable PSNR gains. The method automatically adapts to input image quality, and it may be implemented using low-complexity, swath-based processing. Multiplier-less transforms better suited for parallel hardware implementation are developed. Finally, PSNR comparisons are provided for the different methods. The new method using the DCT transform offers good visual results with PSNR comparable to prior work, and the multiplier-less transforms offer good visual results at a slight loss in PSNR.

1. BACKGROUND

A fast growing archive of JPEG images and short MPEG video clips is generated by digital still cameras. Unfortunately, ringing and blocking artifacts from the block DCT compression may degrade the quality of these images. Reducing these artifacts may improve the display and printing from cell phone camera photos, video clips, images cropped from high-resolution compressed images and images found on the web. Reducing these artifacts also improves the results from subsequently applied algorithms, such as upsampling and color adjustments, that may otherwise accentuate the compression artifacts. Compression artifact reduction also has the potential to improve the millions of images in the accidental archive that the web has become.

There is a large body of research in compression artifact reduction. The majority of prior methods are in three categories: 1) Filtering along block boundaries [1], which reduces blocking but not ringing; 2) Projection on convex sets optimizations [2], which require an undetermined number of expensive iterations; and 3) Wavelet transform thresholding [3], which performs well for deblocking but may leave ringing artifacts, and may also require large memory. In a related prior work, Nosratinia [4] reapplies shifted JPEG transforms to a JPEG image, and finds favorable visual and PSNR results compared to competing techniques, including POCS and wavelet. Nosratinia's method may, however, leave ringing artifacts that are visible both in display and print, as shown in the next section.

2. DESCRIPTION OF THE ALGORITHM

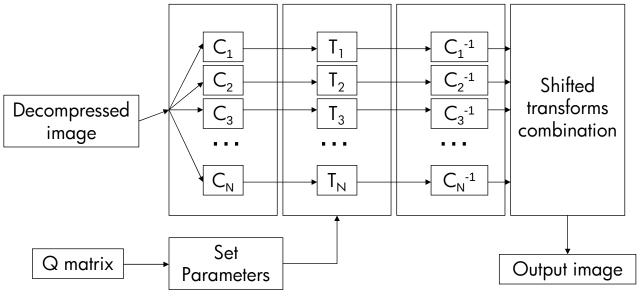

Fig. 1. Block diagram for shifted transform algorithms for compression artifact reduction. The shifted transforms are $C_1, C_2, \ldots, C_N$.

Figure 1 is used to describe both Nosratinia's and our method. Assume the input image is column stacked [5]. Then, the shifted transforms C_i are large sparse matrices that apply to the input image vector. The outputs from these transforms are modified by non-linear denoising point transformations T_i, inverse transformed by inverses C_i^{-1} and finally combined by the *Shifted Transforms Combination* (STC) block to form the output image. The quantization (Q) matrices are used to set parameters of the non-linearities T_i, to adapt to the input image quality. As the Q matrices become finer, the system shown in Figure 1 approaches the identity, the desired outcome when there is no compression.

Nosratinia's algorithm is simply described in terms of Figure 1. The C_i are block DCT transforms aligned to shifted grids i, the nonlinearities T_i are quantizers with parameters given directly by the Q matrices and the STC block is a simple average of its inputs.

Our algorithm modifies the processing in the STC block. We observe that pixels near, but not on, an edge are best reconstructed by those transforms whose block boundaries coincide with the edge. These transforms also result in lower spatial variance at those pixels. Thus, our algorithm, for pixels near edges, increases the weight for transforms with smaller spatial variance. By using pixel range (difference between maximum and minimum pixel values in a spatial window) instead of edge information, the idea is extended to complex structures and textures.

The inputs to the STC block are the intermediate recon-

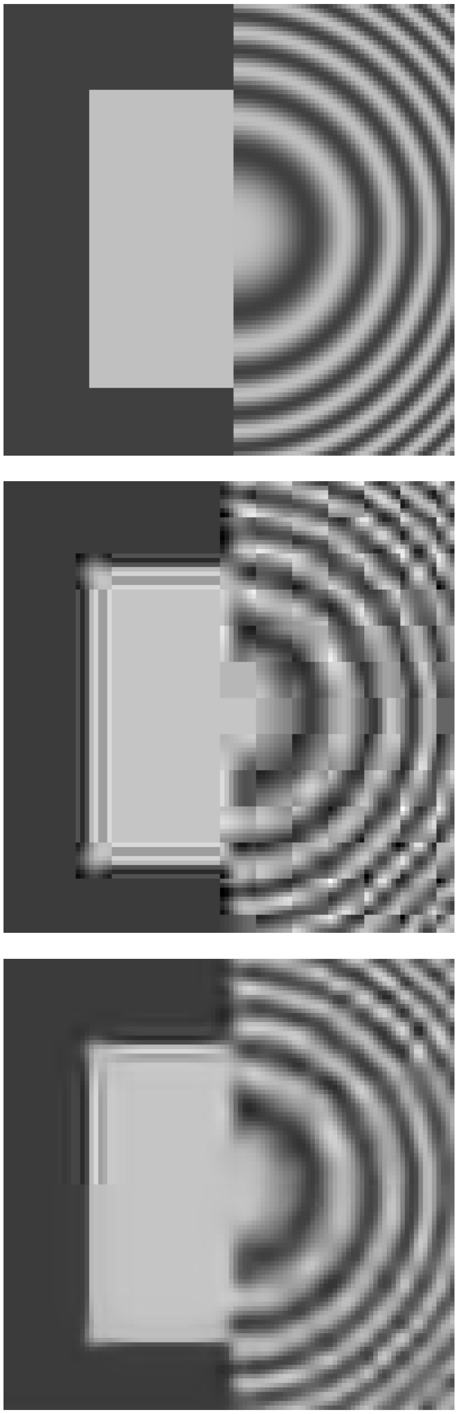

Fig. 2. The top frame shows a synthetic image. The middle frame shows the JPEG compressed version. The top half of the bottom frame applies Nosratinia's algorithm and the bottom half applies our algorithm with block DCT for C_i.

structed pixel values, $p_i(x, y)$, that result from applying C_i, T_i and C_i^{-1}. At a given pixel location, (x, y), our method

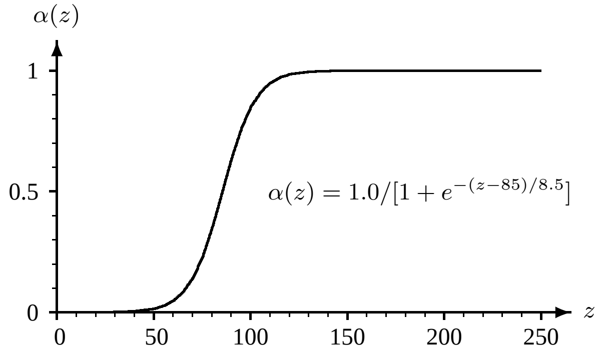

Fig. 3. The sigmoidal function that determines α.

uses information from its spatial neighborhood to adaptively reduce ringing. The output pixel $o \equiv o(x, y)$ is given, in terms of the reconstructed pixel values, $p_i \equiv p_i(x, y)$, by

$$o = (1 - \alpha)\frac{1}{N}\sum_{i=1}^{N} p_i + \alpha\frac{\sum_{i=1}^{N} p_i \mathcal{I}(p_i)}{\sum_{i=1}^{N} \mathcal{I}(p_i)} \qquad (1)$$

The spatially-varying α forms a linear combination of the average, shown in the first term, and a term that selects only those transforms with low reconstructed variance at the pixel of interest. The symbol $\mathcal{I}$ represents an indicator function that, in our implementation, selects only those p_i that are in the lower 30% quantile of spatial variance. Pixels sufficiently far from sharp transitions, where the expected ringing is minimal, have $\alpha \approx 0$. Pixels on a sharp transition have $\alpha \approx 0$ in order to not blur the transition. Pixels sufficiently close to but not on a sharp transition have $\alpha \approx 1$ to reduce the potential ringing.

The image formed by $\frac{1}{N}\sum_{i=1}^{N} p_i$ approximates the original image, and is used to calculate α. We first calculate image contrast using pixel range. The range is calculated for two concentric windows. The first window size is small, 3 by 3, in order to have accurate estimates near edges. The second window size, 15 by 15, is chosen to cover all of the shifted block DCT transforms at a given pixel. The calculations result in a local contrast measure, c_l, and a block contrast measure, c_b. Subsequently, the difference $z = c_b - c_l$, with values from 0 to 255, is the input to a sigmoidal function, shown in Figure 3, that determines α. The parameters of the function in Figure 3 were selected by training with test images using subjective evaluations and objective PSNR values.

The synthetic image in the top frame of Figure 2 is used to test deringing and deblocking. The left half is a rectangle and the right half is a zone plate. The middle frame of the figure shows the image compressed using quantization matrix "Q3" from Nosratinia [4]. This compressed image was processed by Nosratinia's and our algorithm. In the bottom frame of Figure 2, results of Nosratinia's algorithm are shown in the top half and results of our algorithm are shown in the bottom half. Both algorithms show effective deblocking of the zone plane. On the other hand, the rectangle shows considerable residual ringing from Nosratinia's algorithm while ours effectively reduces the ringing.

The top of Figure 4 shows a portion of a JPEG frame from a digital camera, and the bottom shows results of our algorithm using the multiplier-less transforms developed in the next section. In this complex image, the artifacts are reduced while retaining the input image information. Our experience is similar with dozens of low-resolution images.

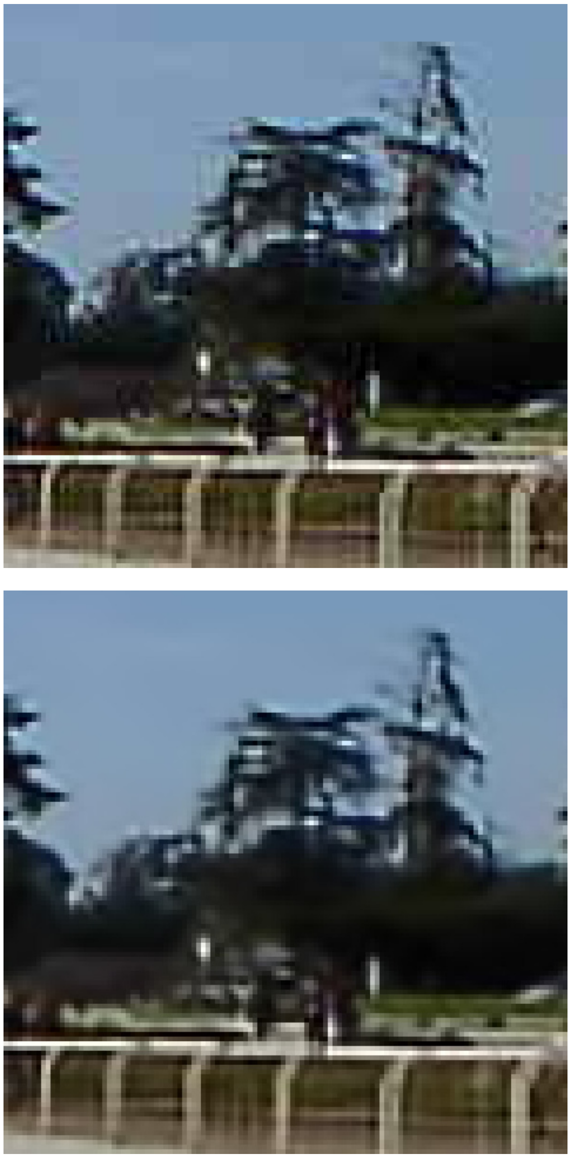

Fig. 4. The top frame shows a portion JPEG compressed image captured by digital camera. The bottom frame shows the output of our algorithm with the multiplier-less transform using $k = 3.5$.

3. MULTIPLIER-LESS TRANSFORMS

For parallel hardware implementation, the DCT transforms may be too expensive. This section develops multiplier-less block transforms, M, that replace the block DCT, to form the C_i in Figure 1. These transforms M retain the ability to automatically set the parameters for the nonlinear T_i, using simple scaling. Techniques similar to those to be described have been used to find efficient transforms [6] for compression. For the artifact reduction case, high coding gain is not a requirement for the derived transforms.

If the transform matrix M has $MM' = \Lambda$, where Λ is a diagonal matrix, then it follows that $M'\Lambda^{-1}M = I$. These two properties allow simple inversion of the transform. If D is the one dimensional DCT matrix, then $M = round(k * D)$ has the desired properties for certain values of k, such as $k = 2.5$, 3.0 and 3.5. We show the details for $k = 3.5$, where M is given by,

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{pmatrix}. \tag{2}$$

The separable [5] application of M in two dimensions is given by,

$$C_M = MXM', \tag{3}$$

where X is an image block, and C_M are the transform coefficients. Recovering the input X from coefficients C_M is given by these steps,

$$\begin{aligned} MXM' &= C_M, \\ XM' &= M'\Lambda^{-1}C_M, \text{ and} \\ X &= M'\left[\Lambda^{-1}C_M\Lambda^{-1}\right]M. \end{aligned} \tag{4}$$

The quantity in the brackets of Equation 4 determines the scaling of the coefficients C_M needed for inversion. The scaling is efficiently computed by combining it with the non-linearities T_i of Figure 1.

We derived a sparse factorization of M that allows the implementation of a fast transform. $M = PF_3F_2F_1$, where P is a permutation matrix, and F_i are sparse factor matrices. These matrices have the following values

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

$$F_3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & -2 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -2 & 0 & 1 \end{pmatrix} \tag{6}$$

$$F_2 = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

$$F_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{8}$$

The next section describes results of using this transform instead of the DCT.

4. RESULTS

A simulation was conducted by extracting Q matrices (for Y, Cb and Cr) from a digital still camera compressed frame, compressing nine color tests images with these Q matrices, and applying the different artifact reduction algorithms to these compressed test images. Eight of the test images were of typical digital camera subject matter such as people and scenery. The ninth test image included a mix of subject matter, including a resolution test chart. Figure 5 compares the PSNR gains over the input compressed images. The figure shows that the gain from Nosratinia's (white bars) and the method in this paper using the DCT (black bars) are comparable. On average Nosratinia's method had 0.043 dB higher gain than our DCT based algorithm. The multiplierless algorithm with $k = 3.5$ (gray bars) lost 0.064 dB in gain compared to our method using the DCT. Visual inspections, however, show good subjective results even with the multiplier-less transforms (see Figure 4). The differences seem to be in textured regions, as was confirmed by looking at difference images between our method using DCT and our method using the multiplierless transforms. Finally, we note that visually, both the DCT based and multiplier-less methods seem preferable to the Nosratinia method, which may leave a visible halo or ringing around object boundaries of images.

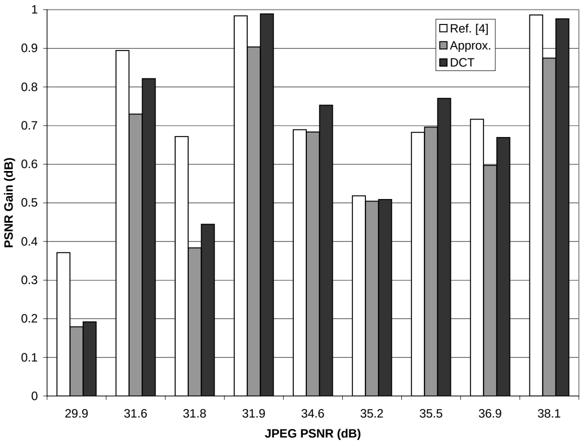

Fig. 5. PSNR comparisons for nine images. Gains in PSNR, Nosratinia (white bars), This paper using DCT (black bars), This paper using multiplier-less transform (gray bars).

5. REFERENCES

[1] J. Chou, M. Crouse, and K. Rachandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Letters*, vol. 5, no. 2, pp. 33–35, February 1998.

[2] Y. Yang, N.P. Galatsanos, and K. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Trans. on Image Processing*, vol. 4, pp. 896–908, July 1995.

[3] Z. Xiong, M. Orchard, and Y. Zhang, "A deblocking algorithm for jpeg compressed images using overcomplete wavelet representations," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, pp. 433–437, April 1997.

[4] A. Nosratinia, "Enhancement of JPEG-compressed images by re-application of JPEG," *Journal of VLSI Signal Procesing Systems for Signal, Image, and Video Technology*, vol. 27, pp. 69–79, February 2001.

[5] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Inc, 1989.

[6] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/avc," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp. 598–603, july 2003.