

- [5] G. Caso and C.-C. J. Kuo, "New results for fractal/wavelet image compression," in *Proc. SPIE Visual Communications and Image Processing*, Orlando, FL, Mar. 1996, vol. 2727, pp. 536–547.
- [6] R. Rinaldo and G. Calvagno, "Image coding by block prediction of multiresolution subimages," *IEEE Trans. Image Processing*, vol. 4, pp. 909–920, July 1995.
- [7] G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Processing*, vol. 7, pp. 141–154, Feb. 1998.
- [8] J. Li, P.-Y. Cheng, and C.-C. J. Kuo, "On the improvements of embedded zerotree wavelet (EZW) coding," in *Proc. SPIE: Visual Communication and Image Processing*, Taipei, Taiwan, R.O.C., May 1995, vol. 2501, pp. 1490–1501.
- [9] J. Li and C.-C. J. Kuo, "Hybrid wavelet-fractal image compression based on a rate-distortion criterion," in *SPIE Visual Communication and Image Processing*, San Jose, CA, Feb. 1997, vol. 3204, pp. 1014–1025.

## Image Sharpening in the JPEG Domain

Konstantinos Konstantinides, Vasudev Bhaskaran,  
and Giordano Beretta

**Abstract**— We present a new technique for sharpening compressed images in the discrete-cosine-transform domain. For images compressed using the JPEG standard, image sharpening is achieved by suitably scaling each element of the encoding quantization table to enhance the high-frequency characteristics of the image. The modified version of the encoding table is then transmitted in lieu of the original. Experimental results with scanned images show improved text and image quality with no additional computation cost and without affecting compressibility.

**Index Terms**— Compression, enhancement, printing, quantization.

### I. INTRODUCTION

We examine the problem of improving the visual quality of digitally captured and JPEG-compressed images. For example, unprocessed, scanned, color text images have very "poor" visual quality because of the inherent limitations of color scanners due to limited modulation transfer function (MTF) and misregistration. MTF is the main cause of pixel aliasing, which results in black text being rendered with a fuzzy, grayish, appearance. An important cause for the MTF degradation is the lens. In most scanners and all cameras, an object is imaged on the sensor through a lens. If a lens is not perfect and the sample is not a monochromatic point source, then the light wave emerging from the lens is not a perfect sphere. This departure is known as *optical path difference* (OPD), and for chromatic light it is caused by six lens aberrations. For example, due to the *chromatic aberration*, where objects of different wavelengths are in focus at different depths, a red object will be focused farther away than a

blue object [1]. To an observer, the OPD appears in the image as a blur of image points away from the center and as color fringes around object outlines.

Misregistration refers to a relative misalignment of the scanner sensors for the various frequency bands. For example, the HP ScanJet IIc has a color misregistration tolerance of  $\pm 0.076$  mm for red and blue with respect to green. (For comparison, at 300 dpi, a pixel is approximately 0.08 mm wide.) Because sharp edges (high frequencies) are very important for reading efficiency, the subjective quality of scanned text is poor at typical scanner resolutions of 200 to 400 dpi.

For scanned images, the visual quality of text can be improved using edge enhancement techniques. Edge enhancement can be performed in either the spatial (e.g., RGB) or frequency domain [2]. In the spatial domain, edge crispening can be performed by discrete convolution of the scanned image with an edge enhancement kernel. This is usually equivalent to filtering the image with a highpass filter. For an  $m \times n$  convolution kernel this approach requires  $mn$  multiplications and additions per pixel.

For edge sharpening in the frequency domain, the full image is first transformed into the frequency domain using the discrete Fourier transform, low-frequency components are dropped, and then the image is transformed back into the time domain. Using a fast Fourier transform, for an  $M \times N$  image, the number of operations for image sharpening in the frequency domain is in the order of  $O(\log_2(MN))$  operations per pixel [3].

In digital cameras and many other applications, such as color facsimile [4], the original captured image is compressed using a frequency-based transformation algorithm before it is stored, transmitted, or printed. In order to compensate for the effects of the capturing device, one may try to sharpen the image before compressing it; however, sharpening before compression may significantly effect the compressibility of the image. Furthermore, it is computationally more effective to combine the coding and sharpening operations into a single step. In this paper we show how sharpening and compression can be combined when one uses the JPEG compression standard. In our scheme, we "enhance" the high frequencies of the compressed image by suitably scaling the encoding quantization tables during decompression. Thus, we can achieve improved text and image quality with no additional computational cost. Furthermore, the enhancement process has no effect on the compressibility of the original image, and the JPEG syntax is not violated.

### II. IMAGE ENHANCEMENT IN THE JPEG DOMAIN

#### A. Preliminaries

Without loss of generality, we consider 24-b color images composed of three color components: one luminance-related and two chrominance. (For example, this could correspond to the CIELAB  $L^*$ ,  $a^*$ , and  $b^*$  components or the  $Y C_b C_r$  components). Since processing is performed either on the luminance component only or on a component-by-component basis, we will describe the technique for only a single component.

For notation purposes, Fig. 1 shows a block diagram of the baseline JPEG encoder [4]. According to the JPEG standard, during compression, pixels of a color component are first divided into  $8 \times 8$  blocks. Let  $y_i$  denote the  $i$ th block. Each block is transformed into the frequency domain using the discrete cosine transform (DCT). Let the output of this step be a new  $8 \times 8$  matrix  $Y_i$ . Next, each element of  $Y_i$

Manuscript received May 6, 1997; revised August 10, 1998. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christine Podilchuk.

K. Konstantinides is with Stream Machine, San Jose, CA 95131 USA (e-mail: kk@streammachine.com).

V. Bhaskaran is with Epson Palo Alto Laboratory, Palo Alto, CA 94304 USA.

G. Beretta is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA.

Publisher Item Identifier S 1057-7149(99)04068-3.

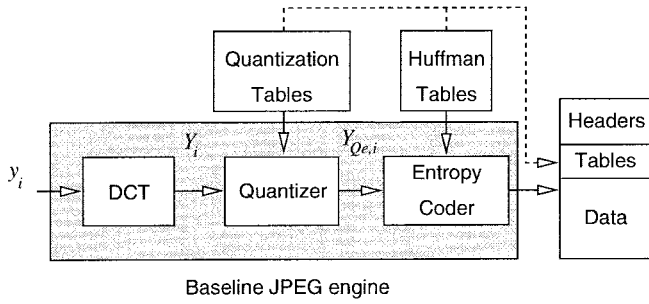


Fig. 1. Block diagram of a baseline JPEG encoder.

is divided by the corresponding element of the encoding quantization table  $Q_e$

$$Y_{Q_e,i}[k, l] = \text{round}\left(\frac{Y_i[k, l]}{Q_e[k, l]}\right), \quad k = 0, 1, \dots, 7, \quad l = 0, 1, \dots, 7. \quad (1)$$

After quantization, the elements of  $Y_{Q_e,i}$  are rearranged in zigzag order and are compressed using lossless entropy coding. According to the JPEG standard, the quantization tables for each color component are defined in the header part of the JPEG file.

During decompression, the process is reversed. An entropy decoder first recovers the  $Y_{Q_e,i}$  blocks. An approximation,  $Y'_i$ , of the  $Y_i$  block is obtained by

$$Y'_i[k, l] = Y_{Q_e,i}[k, l]Q_e[k, l]. \quad (2)$$

Finally, an approximation,  $y'_i$ , of the original  $i$ th block  $y_i$  is obtained from the inverse DCT of  $Y'_i$ .

Since image sharpening is usually equivalent to highpass filtering, it is worth examining the frequency characteristics of scanned images in the DCT domain. Let  $B$  denote the number of  $8 \times 8$  blocks in an image and

$$M_Y[k, l] = \frac{1}{B} \sum_{i=1}^B Y_i[k, l] \quad (3)$$

and

$$V_Y[k, l] = \frac{1}{B} \sum_{i=1}^B (Y_i[k, l] - M_Y[k, l])^2 \quad (4)$$

denote the mean and variance of the  $(k, l)$  DCT frequency component for the whole image. In general, the values for  $M_Y[k, l]$  are close to zero for all components other than the DC component ( $k = 0, l = 0$ ). Fig. 2 shows all 64 values of  $V_Y$  (listed in zigzag order) for the luminance channel of a scanned text image and a synthetic text image. The synthetic image was generated by typing a page using a 10-point serif font (Times New Roman), and then rendering it as an image file. A printed version of the same page was scanned at 300 dpi and 24 b/pixel to generate the scanned text image.

As expected from our previous discussion, Fig. 2 shows that the unprocessed scanned text image has less energy in the high frequencies than the synthetic text image. One would expect a sharper scanned image if its high-frequency components were appropriately "boosted." We will show that for JPEG compressed images, this can be done by appropriately manipulating the quantization table during the decoding of the image.

### B. Scaling of $Q$ Tables

Fig. 3 shows one way for sharpening images in the JPEG domain. The main idea is the following: instead of using the same quantization

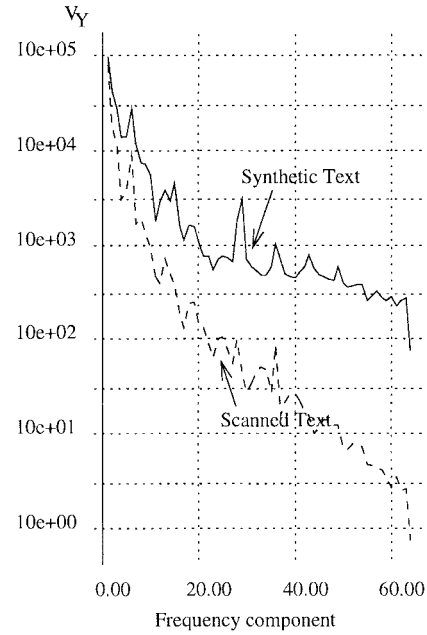
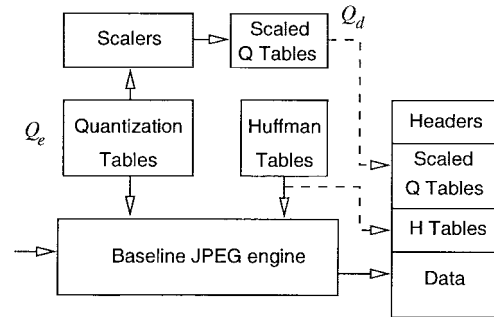

 Fig. 2. Variance of the 64 frequency components in  $8 \times 8$  blocks for synthetic and scanned text images at 300 dpi.


Fig. 3. Data flow for sharpening images in the JPEG domain.

tables for both encoding and decoding a JPEG image, the decoder uses a different set of tables, say  $Q_d$ , which are scaled versions of the tables used in encoding. This also conforms to the JPEG standard. In this section we will describe one possible way to define  $Q_d$  so that the decoded image is a sharpened representation of the original image.

Let  $Q_e$  be an encoding quantization table, and let  $Q_d$  be a decoding quantization table, such that  $Q_d[k, l] = S[k, l]Q_e[k, l]$ . For example, in traditional JPEG, all the  $S[k, l]$  elements are set to one, i.e.,  $Q_d = Q_e$ . Let  $V_r[k, l]$  denote the variance of the  $(k, l)$  frequency component of a reference sharp image. In other words, let  $V_r$  represent the DCT characteristics that our captured images should match.

There are many ways to generate these reference DCT characteristics. For example, for scanners,  $V_r$  may be computed from an image file that has been rendered from a PostScript representation of a typical reference image. Such an example is the "synthetic text" plot shown in Fig. 2.  $V_r$  can also be generated by averaging the DCT characteristics of several sharp images, or from a theoretical model of a sharp image. For a low-end digital camera,  $V_r$  may be generated by measuring the frequency response of a scene captured by another camera of higher quality, or from the DCT characteristics of a digitally captured scene that has been postprocessed using a traditional sharpening algorithm.

Since  $V_r$  represents a reference sharp image, for any decompressed image, we would like to have

$$V_{Y'}[k, l] = V_r[k, l]. \quad (5)$$

From (1) and (2), the variance of such a decoded image is given by

$$V_{Y'}[k, l] = \text{Var}(Y_{Q_e}[k, l]Q_d[k, l]) \approx S[k, l]^2 V_Y[k, l]. \quad (6)$$

From (5) and (6), the scaling factors for the decoding quantization matrix can be obtained by

$$S[k, l]^2 = \frac{V_r[k, l]}{V_Y[k, l]}. \quad (7)$$

Intuitively,  $S$  is a measure of the degradation caused by a particular imaging system in the frequency domain.

As in the case of defining  $V_r$ ,  $V_Y$  does not have to be image dependent. In contrast, it can represent the DCT characteristics of a typical image that has been captured by our target device. For example, for a scanner,  $V_Y$  may be defined from the characteristics of a typical scanned image or from an analytical model of the scanner. Such an example is the "scanned text" plot shown in Fig. 2.

In summary, given a specific device or an image pipeline, the following scheme can be used to compute the scaling matrix for sharpening images in the JPEG domain.

- 1) Generate the reference DCT characteristics for a sharp image, denoted as  $V_r$ .
- 2) Generate typical DCT characteristics for a class of images captured from the specific device, denoted as  $V_Y$ .
- 3) Using  $V_r$  and  $V_Y$ , compute the scaling matrix  $S$  using (7).

After determining the scaling matrix  $S$ , the following scheme, shown in Fig. 3, can be used for sharpening images in the JPEG domain.

- 1) Compress an image using a quantization table  $Q_e$ .
- 2) Generate a new quantization table

$$Q_d[k, l] = S[k, l]Q_e[k, l]. \quad (8)$$

(Note that according to the JPEG standard, all values of the  $Q_d$  table must be bounded between one and 255.)

- 3) Transmit the  $Q_d$  matrix instead of  $Q_e$ .

Assuming that the  $S$  matrix has been computed offline, the above procedure requires only 64 additional multiplications, regardless of the size of the image. Note that the decoder has no knowledge of either the  $S$  or  $Q_e$  matrices, which can remain proprietary if needed.

In an alternative implementation, when one has no control on the way the original JPEG image is generated, the following scheme can be used to sharpen a JPEG image.

- 1) Extract the original quantization table  $Q_e$  from the JPEG file.
- 2) Generate a new quantization table  $Q_d[k, l] = S[k, l]Q_e[k, l]$ .
- 3) Instead of the  $Q_e$  matrix use the  $Q_d$  matrix to decompress the file.

### III. EXPERIMENTAL RESULTS

The original development of this work was motivated by the need to improve the quality of scanned text transmitted using the color facsimile standard. Under this standard, scanned images (at 200 dpi) are first translated to the CIELAB color space, they are compressed using baseline JPEG, and they are transmitted using the standard T.42 facsimile protocol [5]. For the scanned and synthetic (or reference) text images used in Fig. 2, Fig. 4 shows the variances of the 64 DCT frequency components for an unprocessed scanned image, the synthetic text image (used to compute  $V_r$ ), and the image decompressed using the  $Q_d$  matrix derived using (8) (we refer to this as sharpened text.) As expected, there is higher energy in the frequency components of the processed image. Text in the printed processed image was also darker and sharper than text in the

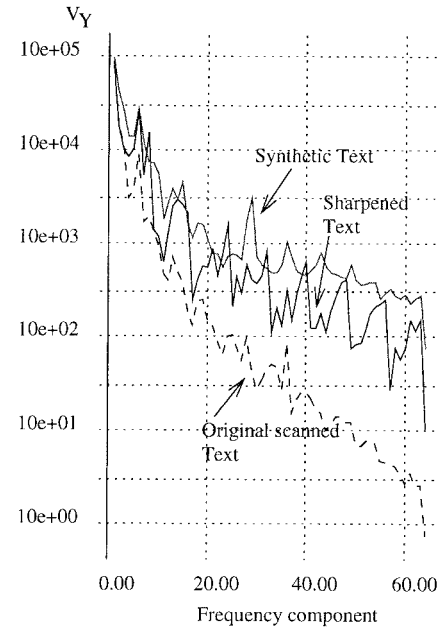


Fig. 4. Variance of frequency components for  $8 \times 8$  blocks in synthetic, scanned, and sharpened text images at 300 dpi.

1.00	1.31	1.46	2.00	2.55	2.54	2.95	4.26
1.03	1.38	1.61	2.15	2.85	3.16	3.70	4.57
1.31	1.58	1.85	2.33	3.01	3.51	4.22	5.10
1.73	2.01	2.16	2.63	3.23	3.89	4.88	5.87
1.78	2.46	2.62	3.01	3.59	4.44	5.69	6.86
2.41	2.98	3.10	3.59	4.42	5.33	6.62	7.16
2.25	3.63	3.83	4.58	5.88	6.80	8.04	8.00
1.97	4.21	4.59	5.76	7.22	7.54	8.04	7.30

Fig. 5. Example scaling matrix for a ScanJet Iic scanner and CIELAB images scanned at 300 dpi.

unprocessed one. The elements of the scaling matrix used are shown in Fig. 5. The same matrix was used for all color images scanned using this scanner.

Experiments also showed that this method is robust with respect to different fonts and font sizes. While holding  $S[k, l]$  constant, we experimented with scanned images with both serif and sans-serif fonts, and with font sizes ranging from 8 points to 18 points.

The technique was also applied successfully to scanned images that contain both text and pictures, without a need for preprocessing them using text-image segmentation algorithms. In this case, we scaled the elements of the quantization table for the luminance channel, except for the (0,0) element. Experimental results showed sharper images and text with no color distortion. As an example, Figs. 6–8 show the  $L^*$  channel of a scanned at 200 dpi image, processed by three different techniques. Fig. 6 is the original unprocessed, and uncompressed image. Fig. 7 shows the output of a conventional  $3 \times 3$  sharpening filter on the RGB data of the uncompressed image. The convolution kernel used was [2]

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (9)$$

The text looks sharper, but the image is corrupted by "salt and pepper" noise. Moire patterns are also enhanced. Fig. 8 shows the output of the new algorithm. Note that, both the text and the image look sharper. The results are far more apparent on color reproductions of these images.



Fig. 6. L\* channel of original and uncompressed image.



Fig. 8. L\* channel of image sharpened in the JPEG domain.

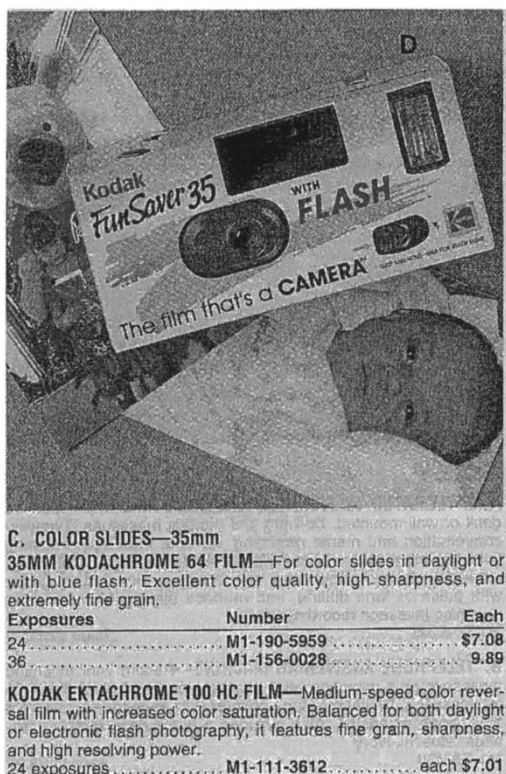


Fig. 7. L\* channel of original processed by a 3 × 3 sharpening filter.

We should also note, that in contrast to Figs. 6 and 7 which are shown uncompressed, Fig. 8 was also compressed at a compression ratio of 24:1 using the example (Annex K) JPEG quantization tables [4]. For decompression, we used the  $Q_d$  matrix of (8) with the

scaling factors shown in Fig. 5. As explained before, those scaling factors were computed using the DCT characteristics of synthetic and scanned text plotted in Fig. 2. In other words, the scaling matrix we used for sharpening Fig. 6 was derived from completely different images, but for the same scanner.

From a complexity point of view, consider an 8.5 × 11 inches page scanned at either 200 or 300 dpi. Assuming we process only the luminance color plane, JPEG-based sharpening requires only 64 multiplications, regardless of the scanning resolution. On the other hand, using the 3 × 3 kernel shown above, conventional sharpening requires 3.7 million multiplications and 14.96 million additions for the 200 dpi image, or 8.41 million multiplications and 33.66 million additions for the 300 dpi image.

As an another example, we applied the JPEG-based sharpening scheme to a JPEG image captured with a digital camera. The original image had a resolution of 640 × 480 pixels, and it was compressed at 12:1. We had no access to the uncompressed data. Fig. 9 (top) shows part of the original image, magnified by a factor of two to better show image details. For JPEG-based sharpening, according to (7), we needed estimates of 1) the DCT characteristics of a typical captured image ( $V_Y[k, l]$ ), and 2) the DCT characteristics of a reference-sharp image ( $V_r[k, l]$ ). Since we had no access to the original uncompressed image,  $V_Y$  was computed from the decompressed captured image. The reference sharp image, shown in Fig. 9 (middle), was generated by decompressing the original image and by sharpening it using an adaptive unsharp-masking algorithm. Fig. 9 (bottom) shows the JPEG-based sharpened image. All images were printed on a Fujix Pictography 3000 printer.

From Fig. 9, as expected, the adaptive unsharp masking procedure yields better results than JPEG-based sharpening. Since Fig. 9 (middle) was the reference sharp image for generating the scaling matrix, there was no way that JPEG-based sharpening could produce an image that would be better than the reference sharp image. How-



Fig. 9. Example of JPEG-based sharpening for a digital camera. Top: Original image. Middle: Conventionally sharpened (reference sharp image). Bottom: JPEG-sharpened.

ever, the results are quite satisfactory, considering that JPEG-based sharpening requires no additional computation resources. Overall, the strength of JPEG-based sharpening is in terms of computational cost and in providing very good results for halftoned scanned images, where traditional sharpening schemes may enhance moire patterns.

#### IV. CONCLUSION

We presented a simple and cost-effective technique for sharpening images in the compression domain. Among the advantages of this technique are the following.

- 1) It is fully compliant with the JPEG standard.
- 2) No extra computation is required: The quantization matrices can be precomputed and can be used in a class of images.

- 3) The receiver has no knowledge of either the quantization matrix used by the encoder or the scaling factors. Hence, encoding quantization matrices can remain proprietary.
- 4) Sharpening is achieved simultaneously with decompression, thus, there is no additional cost during decompression, unlike conventional postprocessing or preprocessing sharpening schemes.

The proposed method is particularly attractive to high-volume, consumer-type, applications, such as low-end digital cameras, scanners, and facsimile machines, where better lenses or more powerful microcontrollers (to implement conventional sharpening schemes) may have a significant impact on the cost.

In summary, JPEG-based processing represents a low-cost alternative to traditional image sharpening schemes and it can easily be extended to other frequency-based image processing applications.

#### ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable comments and suggestions.

#### REFERENCES

- [1] M. Born and E. Wolf, *Principles of Optics*, 5th ed. New York: Pergamon, 1975.
- [2] W. K. Pratt, *Digital Image Processing*, 2nd ed. New York: Wiley, 1991.
- [3] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
- [4] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*, 2nd ed. Boston, MA: Kluwer, 1997.
- [5] ITU-T Recommendation T.42, "Continuous-tone color representation method for facsimile," Feb. 1996.