# Classified JPEG Coding of Mixed Document Images for Printing

Marcia G. Ramos and Ricardo L. de Queiroz

***Abstract*—This paper presents a modified JPEG coder that is applied to the compression of mixed documents (containing text, natural images, and graphics) for printing purposes. The modified JPEG coder proposed in this paper takes advantage of the distinct perceptually significant regions in these documents to achieve higher perceptual quality than the standard JPEG coder. The region-adaptivity is performed via *classified thresholding* being totally compliant with the baseline standard. A computationally efficient classification algorithm is presented, and the improved performance of the classified JPEG coder is verified.**

## I. Introduction

Digital documents have become commonplace in today's printing systems. Color documents have mixed contents and may contain text, graphics, and pictorial data so that scanned or rendered rasters of those documents are commonly represented as continuous-tone images. The digital representation of those images often require a large amount of data. For example, an 8-bit image for an 8.5 in × 11 in full-color page at a resolution of 600 pixels-per-inch (ppi) would demand around 30 megabytes per color channel. In this example, storage of a large document (e.g., hundreds of pages) or processing at high speed (e.g., 100 pages/min) are prohibitive without some form of data compression. Generally, document compression is advantageous for commercial and domestic printing systems. However, the compression scheme not only has to be fast enough to match the large data amount and processing speed, but it also has to provide visually lossless compressed documents for guaranteed quality.

The most popular compression scheme for printing systems is the JPEG baseline standard, or JPEG for short [1]. JPEG requires little buffering and can be efficiently implemented so that existing implementations of JPEG are able to provide the required processing speed for most cases. However, image quality has to be maintained at the cost of low compression. The main problem with this approach to vary compression ratios lies in the fact that all the regions in the mixed document are treated equally by JPEG. Compression artifacts are more easily perceived around the text regions of the document than in smooth or textured areas, and this has to do with our ability to recognize object shapes [2].

By using a region-adaptive approach to the compression of mixed documents, the performance of the JPEG coder can be significantly improved. Region-adaptive compression within the JPEG framework can be achieved by means of blockwise adaptive quantization. Within the JPEG baseline system, there is no specification for varying the quantization on a block-by-block basis. Recently, an extension to the JPEG standard (JPEG-3) has been drafted to support variable quantization through the use of multiple scaling factors that are encoded as part of the bit stream [3]. The specification of the scaling factors is not a part of the standard and they are chosen according to each user and application. Depending on the number of scaling factors adopted and how often they change from block to block, a considerable number of extra bits may be necessary to code the scaling factor information. A JPEG-3 compliant coder is presented in [4], wherein the quantization scaling factor is adjusted for the text and image blocks (only two document regions are used). Blockwise distortion adaptivity in JPEG can also be conceived through the use of multiple quantizer tables [5]–[7], which are switched on a block-by-block basis. Of course, the resulting bit streams resulting from those methods are not compatible with the JPEG baseline standard.

Apart from adaptive quantization, thresholding techniques [8] can achieve blockwise adaptation within JPEG. In that, less relevant coefficients in a rate-distortion (RD) sense are simply discarded (thresholded) from a block, and the RD analysis can be made globally [8] or locally [9]. Also, in a non-RD-based framework, perceptual models can be used for discarding DCT coefficients which are visually less important [10].

Here, the image is segmented into text and nontext regions, and the nontext regions are further classified into edge, smooth, and detailed regions according to recommendations from various psychophysical studies [2], [11], [12]. The studies suggest that higher perceptual importance should be given to the text and edge regions, followed by the smooth and detailed regions. The goal is to devise a JPEG baseline compliant coder whose RD characteristics are guided by the psychophysical classification process.

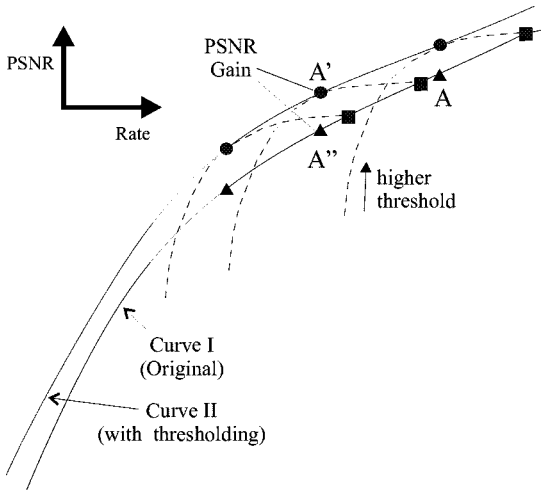


Fig. 1. Illustration of RD curve improvement by thresholding.

## II. Classified Thresholding in JPEG

### A. Thresholding

The proposed algorithm is based on the *thresholding* technique which in this context sets to zero some AC DCT coefficients in JPEG based on RD characteristics [8]. The performance improvement provided by thresholding is illustrated in Fig. 1. Given a method to select quantizer tables in JPEG, let the coder performance (rate versus distortion or RD curve) be depicted in curve I. Thresholding provides short gains by deleting few coefficients in a manner that is more efficient than increasing the quantizer steps in JPEG. In Fig. 1, from the starting points in curve I (marked by squares), by selectively setting some AC coefficients we reduce both rate and PSNR. The RD locus follows the dashed curves, where the rate-by-PSNR trade-off is large at first before decaying drastically. For example, starting

Manuscript received May 12, 1998; revised October 24, 1999. The associate editor coordinating the review of this paper and approving it for publication was Prof. Rashid Ansari.

M. G. Ramos is with Cornell University, Ithaca, NY 14853 USA (e-mail: mramos@anise.ee.cornell.edu).

R. L. de Queiroz is with the Xerox Corporation, Webster, NY 14580 USA (e-mail: queiroz@wrc.xerox.com).

Publisher Item Identifier S 1057-7149(00)02669-5.

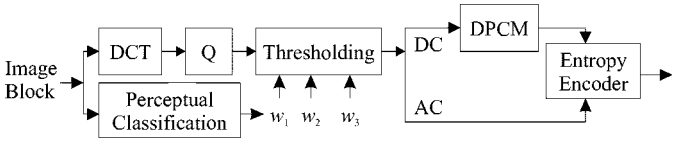


Fig. 2. Classified thresholding encoder block diagram.

TABLE I
HVS-BASED WEIGHTS FOR THE 8 × 8 DCT COEFFICIENTS CHOSEN FOR THE NONEDGE PERCEPTUAL REGIONS. FOR EDGE REGIONS ALL WEIGHTS ARE $1000\alpha$, WHERE $\alpha$ CONTROLS THE DISTORTION TO EDGE (TEXT) REGIONS

| *Smooth* areas | | | | | | | |
|---|---|---|---|---|---|---|---|
| 246 | 1000 | 791 | 486 | 267 | 139 | 69 | 33 |
| 1000 | 955 | 715 | 443 | 247 | 130 | 65 | 32 |
| 791 | 715 | 533 | 341 | 197 | 106 | 55 | 27 |
| 486 | 443 | 341 | 229 | 139 | 78 | 41 | 21 |
| 267 | 247 | 197 | 139 | 88 | 52 | 29 | 15 |
| 139 | 130 | 106 | 78 | 52 | 32 | 18 | 10 |
| 69 | 65 | 55 | 41 | 29 | 18 | 11 | 6 |
| 33 | 32 | 27 | 21 | 15 | 10 | 6 | 4 |
| *Detailed* areas | | | | | | | |
| 246 | 854 | 1000 | 935 | 791 | 631 | 486 | 364 |
| 854 | 952 | 997 | 915 | 771 | 616 | 475 | 356 |
| 1000 | 997 | 955 | 852 | 715 | 573 | 443 | 334 |
| 935 | 915 | 852 | 752 | 631 | 509 | 397 | 302 |
| 791 | 771 | 715 | 631 | 533 | 433 | 341 | 262 |
| 631 | 616 | 573 | 509 | 433 | 356 | 284 | 221 |
| 486 | 475 | 443 | 397 | 341 | 284 | 229 | 180 |
| 364 | 356 | 334 | 302 | 262 | 221 | 180 | 143 |

from point A one can obtain point A$'$, using thresholding, which has a large PSNR than point A$''$, which is obtained by changing the quantizer table until matching the rate of A$'$ (matching points are marked by triangles). Thus, there is a short interval wherein there are gains in thresholding. If one can fine tune thresholding in a way that it would always work near its best performance (points marked by circles), the equivalent RD curve would be curve II, which shows a small but consistent gain in performance. The backdraw of the general thresholding method is that it may require buffering the image and several passes in each block to find the optimal operating point, however it typically increases the PSNR by about 1.0 dB. We employ a simplifying variant [9] in which we analyze each nonzero quantized AC coefficient independently. The method in [9] improves speed and reduces memory buffering compared to [8]. It does not require more memory than regular JPEG and the computational overhead is less than 10%, while providing a typical improvement in PSNR around 0.6–0.7 dB (for a constant MSE distortion measure).

### *B. Adaptive Distortion*

In general, for RD optimized transform coding, the signal is divided into units $x_i$, each contributing to the overall bit-rate $R$ by $R_i$ bits, i.e., $R = \sum_i R_i$. Distortion is some function of the quantization error $\hat{x}_i - x_i$, where $\hat{x}_i$ is the reconstructed unit. The global distortion is given by

$$D = f(\{\hat{x}_i - x_i, \forall\, i\}) \left( \text{e.g., } = \sum_i (\hat{x}_i - x_i)^2 \right). \tag{1}$$

By using a well-behaved distortion function such as MSE, any processing can be accounted in the RD balance by minimizing a cost function $J$ which combines rate and distortion through a Lagrangian multiplier [8]: $J = R + \lambda D$. We do not challenge the optimization principle,


(a)

(b)

Fig. 3. (a) Wine image and (b) its classification map (white = smooth, black = edge/text, gray = detailed).

rather we aim to improve the subjective coder performance by defining a space varying meaning for distortion as opposed to adapting the algorithm, i.e.

$$D = \sum_i f_i(\hat{x}_i - x_i) \left( \text{e.g., } = \sum_i (\hat{x}_i - x_i)^2 u_i \right) \tag{2}$$

where $u_i$ is a distortion weighting factor specific for the $i$th unit. This algorithm clearly demands *a priori* classification of the signal in order to identify units and assign proper weights. In JPEG and similar transform coders, the block with pixels $x(m, n; i, j)$, i.e., position $(i, j)$ within block $(m, n)$, is transformed into block $y(m, n; i, j)$ through an orthonormal transformation. Hence, the example above becomes

$$D = \sum_{i,j;\, m,n} (\hat{y}(m, n; i, j) - y(m, n; i, j))^2 \cdot u(m, n; i, j). \tag{3}$$

In conventional human visual system (HVS) weighted error measures, $u(m, n; i, j) = w(i, j)$, i.e., a fixed frequency-based weighting system is used [13]. If $w(p; i, j)$ is a weighting matrix whose entries are function of block $(m, n)$, i.e., $p = g(x(m, n))$, for some $g$, then weighting can be made block adaptive. The HVS response is not completely understood and cannot be easily modeled. Thus, we decided to classify the image blocks into a discrete number of representative

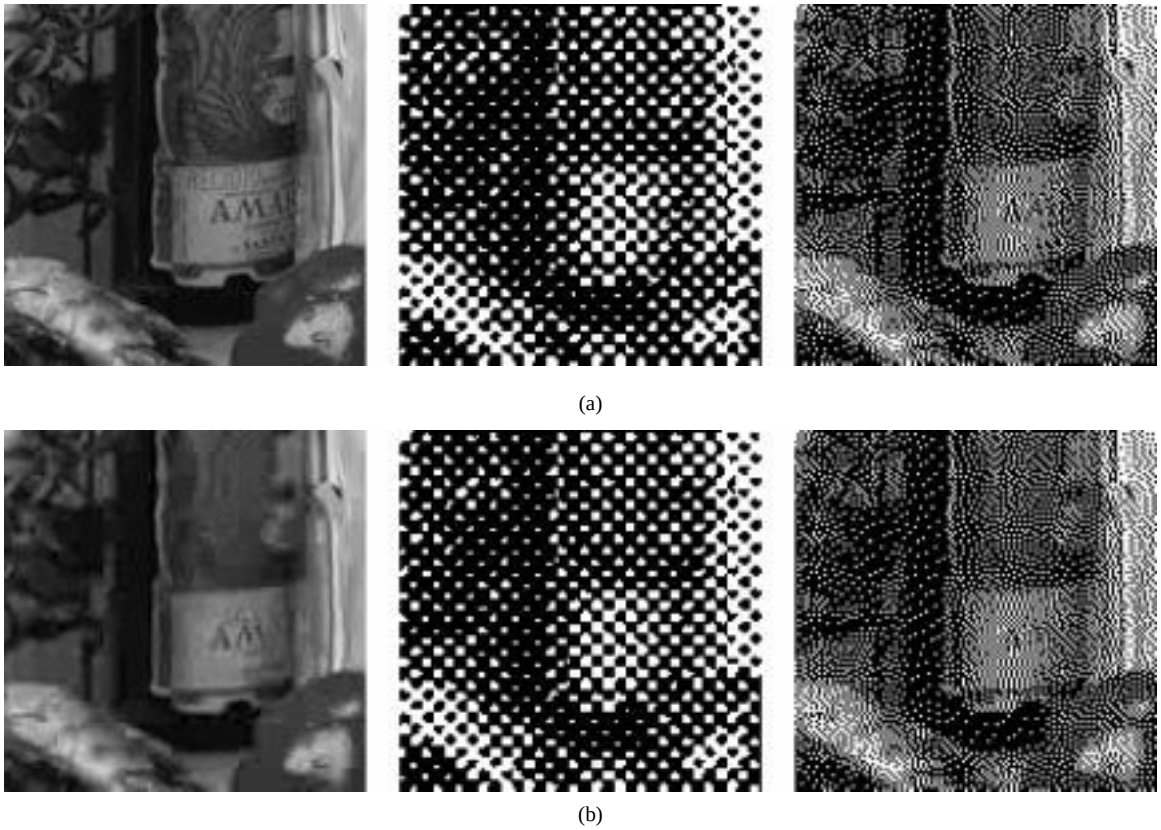

Fig. 4. Nonedge areas and their halftones (clustering and error diffusion). (a) JPEG compressed without using CT. (b) Overblurred by using CT. Although the continuous tone images are very different, the corresponding halftones are very similar.

classes and devise HVS weights $w_k(i, j)$ for each of the classes so that

$$x(m, n) \in \text{ class } C \rightarrow u(m, n; i, j) = w_C(i, j) \tag{4}$$

i.e., distortion is measured as block-classified perceptually weighted MSE.

### C. Classified Thresholding

In this paper, we incorporate perceptual classification into the thresholding decision. The goal of this perceptual weighting is to guarantee that most of the high frequency coefficients to be thresholded are from smooth and detailed blocks, while preserving the text and edge regions of the image. As in [9], for each nonzero AC coefficient in a given block, a cost-benefit ratio is computed where the cost is defined as the number of bits required to encode the coefficient and the benefit is defined as the decrease in distortion achieved when the coefficient is kept. The cost/benefit ratio is then compared to a threshold to decide whether or not the coefficient will be discarded.

First, the image pixels in the block undergo a fast block classification algorithm to determine to which of the predetermined classes the input block belongs. Without loss of generality, let us assume it belongs to class $K$. The block is then transformed using the DCT and quantized. In JPEG, quantized DCT coefficients of a block are mapped into a vector $zz(n)$ by scanning the block in a zigzag path. For a nonzero quantized coefficient $zz(n)$, assume the next nonzero quantized coefficient in the vector order is $zz(l)$ at index $l$. Without getting into the details of the JPEG variable length coding algorithm [1], it suffices to say that a nonzero sample $zz(i)$ produces two bit quantities: 1) $b$ bits are used to encode a composite symbol, where this symbol tells the decoder how many zero samples are there before $zz(n)$ (and after the last nonzero coefficient in the path) and also provides information on the magnitude of $zz(i)$ and 2) $SSSS$ bits are used to encode the sign of $zz(n)$ and the remaining information relative to its magnitude. Let $b = b_1$ and $SSSS = SSSS_1$ for $zz(n)$ and $b = b_2$ and $SSSS = SSSS_2$ for $zz(l)$. If $zz(n)$ is discarded, the run of zeroes before $zz(l)$ increases and it then spends $b_3 + SSSS_2$ bits to be encoded. The cost $R(n)$ of keeping the coefficient is then the cost of sending $zz(n)$ and $zz(l)$ minus the cost of sending $zz(l)$ if $zz(n) = 0$, that is

$$R(n|zz(n) \neq 0) = b_1 + b_2 - b_3 + SSSS_1. \tag{5}$$

The benefit achieved with keeping the coefficient is then the decrease in distortion given by the information conveyed in $zz(n)$. Let the original nonquantized coefficient be $d(n)$, the quantizer step size be $q(n)$, and the perceptual weight associated with that coefficient be $w_K(n)$, where $K$ is the class associated with the block. The distortion resulting from quantizing and keeping the coefficient is then $|d(n) - zz(n)q(n)|^2 w_K(n)$, while the distortion resulting from thresholding the coefficient is simply $|d(n)|^2 w_K(n)$. The decrease in distortion given by keeping $zz(n)$ is then

$$D(n|zz(n) \neq 0) = w_K(n) zz(n) q(n) (2d(n) - zz(n)q(n)). \tag{6}$$

Finally, the cost-benefit, i.e., RD ratio is given by

$$\nu(n) = \frac{b_1 + b_2 - b_3 + SSSS_1}{w_K(n) zz(n) q(n) (2d(n) - zz(n)q(n))}. \tag{7}$$

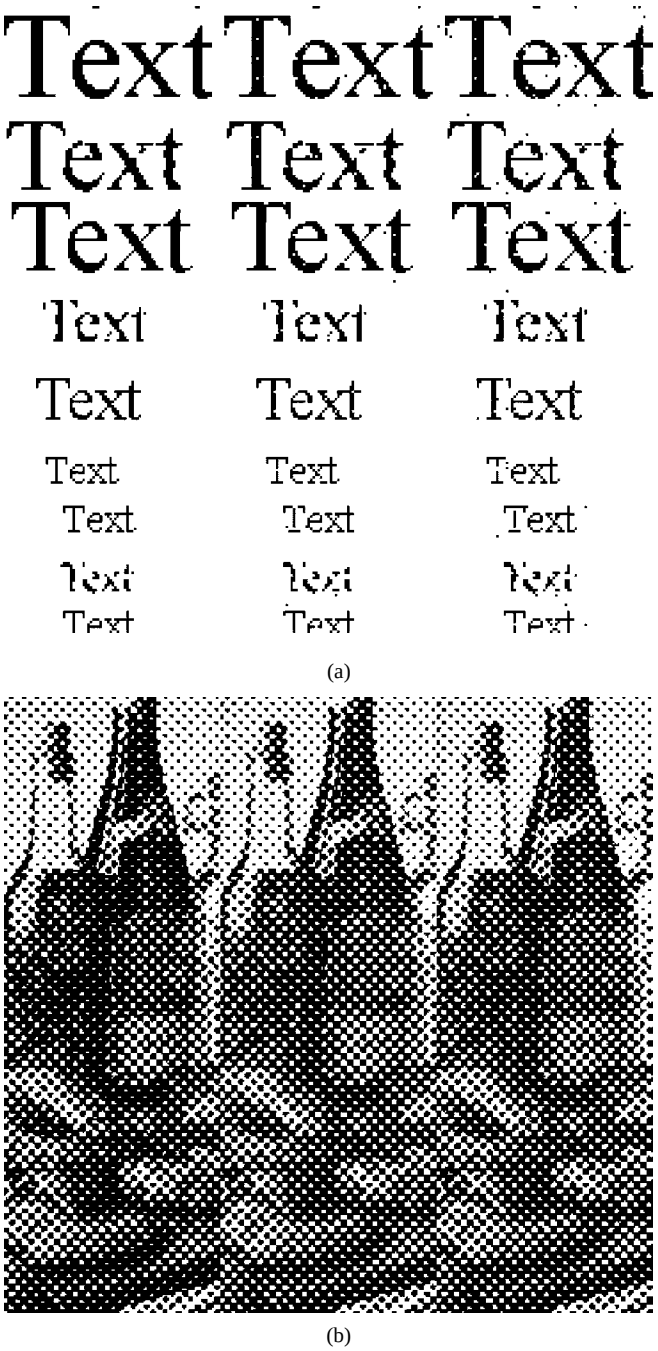

Fig. 5. Zoom of halftones of original and reconstructed images: (a) edge regions and (b) mixed regions. In each, the left image is the halftone of the original image, the center corresponds to using CT the right one corresponds to JPEG without CT.

$\nu(n)$ is compared to a threshold $\tau$ and $zz(n)$ is set to zero whenever $\nu(n) > \tau$. The threshold can be found experimentally (e.g., trained) or by using bisection. A block diagram of the encoder is shown in Fig. 2 and the technique is referred to here as classified thresholding (CT).

## III. Perceptual Classification

We assume the image blocks are to be classified into edge, smooth, or detailed blocks. Any classification algorithm can be employed, nevertheless we developed our own fast algorithm which was shown to be more efficient than other more sophisticated methods [14]. The algorithm classifies blocks through a simple analysis of the luminance difference values inside a block. Let $x(i, j)$ $(0 \le (i, j) \le 7)$ be the pixels in an 8 × 8 block and let $x'(k, l)$ $(0 \le (k, l) \le 3)$ be the pixels in a 4 × 4 block found by subsampling the 8 × 8 block through averaging of 2 × 2 neighboring pixels. The activity measures computed are the maximum differences among neighboring pixels in a block

$$\mu_1 = \max\{|x(i, j) - x(i-1, j)|, |x(i, j) - x(i, j-1)|\} \quad \text{for } 1 \le (i, j) \le 7 \tag{8}$$

$$\mu_2 = \max\{|x'(k, l) - x'(k-1, l)|, |x'(k, l) - x'(k, l-1)|\} \quad \text{for } 1 \le (k, l) \le 3. \tag{9}$$

Two thresholds $T_{lo}$ and $T_{hi}$ $(T_{hi} > T_{lo})$ are required for classification so that

$$\begin{aligned} \mu_1 > T_{hi} &\rightarrow \text{edge block} \\ \mu_1 < T_{lo} \text{ and } \mu_2 < T_{lo} &\rightarrow \text{smooth block} \\ \text{else} &\rightarrow \text{detailed block.} \end{aligned}$$

Note that computation can be greatly simplified if the algorithm is bypassed when a pixel difference is found larger than $T_{hi}$ while computing $\mu_1$. Also, if $\mu_1 \ge T_{lo}$, it is not necessary to compute $x'(i, j)$ or $\mu_2$. Experimentally, the thresholds $T_{lo} = 30$ and $T_{hi} = 60$ achieved good results for a wide range of mixed documents. These thresholds, however, are not absolute and the user may select them as desired according to the image and application. The user may also choose to use a two-way classification into purely text and nontext blocks, in which case strong image edges belonging to nontext regions are given a lower priority than the text regions.

## IV. Compression Results

The weights used to threshold the DCT coefficients were found by calculating DCT-domain energy of a linear HVS transfer function assuming maximum viewing frequency of 56 and 28 cycles/degree for smooth and detailed regions, respectively. The weights used are shown in Table I and can be changed depending on the application. For edges, uniform weighting was used with $\alpha = 2$.

A monochrome test image "wine" of 512 × 512 pixels (8 bits/pel) is shown along with its segmentation map in Fig. 3. This image was compressed with JPEG at 1.11 bpp using default quantizer tables, with and without CT. The image compressed without CT yielded a PSNR of 35.69 dB, while using CT caused the PSNR to drop by 4 dB. The drop in PSNR is irrelevant since it reflects the distortion measure in (1), while we are interested in the one in (3) and (4). The idea behind CT is to overblur the nonedge areas while saving bits to spend in the edge areas. Starting from a modestly compressed image, CT decreases PSNR for nonedge areas. Compared to not using CT at the same compression ratio, PSNR is slightly superior in text areas and substantially inferior otherwise. Overblurring is not a large problem if one wants to halftone the image, as shown in Fig. 4 since most of the details are lost in the halftoning process. However, ringing on text areas are shown as annoying sparse dots around letters. Enlarged portions of the halftoned reconstructed images are shown in Fig.

5 for $\tau = 0.01$.[1] Note that by using CT ringing around text is diminished while the details lost in textured areas are not captured by the halftoning process. In other words, after halftoning, the image compressed using CT is superior in all aspects. This is so because the adaptive distortion measure (weights) intentionally took into account masking properties. So, overblurring smooth or detailed regions was not as penalized as blurring text areas. Similar results were obtained for other mixed images as well. Overall, the method's compression improvement depends largely on image contents. The process of matching the visual quality between two images for a compression ratio comparison is imprecise to say the least. Nevertheless, for images with edge-texture mixtures similar to that of the example image, typical rate savings are in the order of 10% to 40%. Furthermore, parameter adjustment needs to be tied to imaging process and viewing conditions. However, the method seems to provide a performance improvement for most cases.

## V. Conclusions

This paper presented a new method to incorporate an adaptive distortion measure to a baseline-compliant JPEG coder. The adaptation was possible by adapting HVS weights for an RD-optimized thresholding technique with the guidance of a space-domain block classification technique. The perceptual classification was performed using a pixel-based classification algorithm which was shown to be visually accurate. The goal of our approach is to allow perceptually-adaptive baseline JPEG compression in order to preserve the visual quality of the most significant regions in a mixed document. The concept of thresholding with adaptive distortion measure can also be used in other coders, e.g., JPEG 2000.

## References

[1] W. B. Pennebaker and J. L. Mitchell, *JPEG—Still Image Data Compression Standard*, New York: Van Nostrand Reinhold, 1993.
[2] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.
[3] ISO/IEC, CD 10 918-3, "Information Technology—Digital compression and coding of continuous tone still images—Pt. 3: Extensions,", Nov. 13, 1994.
[4] K. Konstantinides and D. Tretter, "A method for variable quantization in JPEG for improved text quality in compound documents," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Chicago, IL, Oct. 1998, pp. 565–568.
[5] N. Chaddha, A. Agrawall, A. Gupta, and T. H.-Y. Meng, "Variable compression using JPEG," in *Proc. Int. Conf. Multimedia Computing Systems*, Boston, MA, May 1994, pp. 562–569.
[6] R. Rosenholtz and A. B. Watson, "Perceptual adaptive JPEG coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Lausanne, Switzerland, Sept. 1996, pp. 901–904.
[7] M. G. Ramos and S. S. Hemami, "Edge-adaptive JPEG image compression," in *Proc. SPIE Visual Communications Image Processing*, vol. 2727, Mar. 1996, pp. 1082–1093.
[8] K. Ramachandran and M. Vetterli, "Rate-distortion fast thresholding with complete JPEG-MPEG decoder compatibility," *IEEE Trans. Image Processing*, vol. 3, pp. 700–704, Sept. 1994.
[9] R. L. de Queiroz, "Processing JPEG-compressed images and documents," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 334–337.
[10] R. Safranek, "A comparison of the coding efficiency of perceptual models," in *Proc. SPIE Human Vision, Visual Processing, Digital Display IV*, vol. 2411, Feb. 1995, pp. 83–91.
[11] X. Ran and N. Farvardin, "A percepually motivated three-component image model—Part I: Description of the model," *IEEE Trans. Image Processing*, vol. 4, pp. 401–415, Apr. 1995.
[12] M. G. Ramos and S. S. Hemami, "Perceptually-based scalable image coding for packet networks," *J. Electron. Imag.*, vol. 7, pp. 453–463, July 1998.
[13] R. L. de Queiroz and K. R. Rao, "Transform coding," in *Handbook of Visual Communications*, H.-M. Hang and J. W. Woods, Eds, New York: Academic Press, 1995, ch. 7, pp. 223–263.
[14] M. G. Ramos, S. S. Hemami, and M. A. Tamburro, "Psychovisually-based multiresolution image segmentation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Santa Barbara, CA, Oct. 1997, pp. 66–69.

[1]The images were highly magnified to avoid detail losses when rendering this paper and to minimize the effect of an eventual printer halftone on them. The digital halftone images were created using popular error diffusion (Floyd and Steinberg weights) and clustered dots (4 × 4 Versatec) methods.

# Fast and High Performance Image Subsampling Using Feedforward Neural Networks

Adriana Dumitraş and Faouzi Kossentini

***Abstract*—We introduce a fast and high-performance image subsampling method using feedforward artificial neural networks (FANN's). Our method employs a pattern matching technique to extract local edge information within the image, in order to select the FANN desired output values during the supervised training stage. Subjective and objective evaluations of experimental results using still images and video frames show that our method, while less computationally intensive, outperforms the standard lowpass filtering and subsampling method.**

***Index Terms*—Feedforward neural network, pattern matching, subsampling, training algorithm.**

## I. Introduction

Image subsampling is important in many applications, such as lossy compression [1], [2], sub-band and pyramidal image decomposition [3], sampling structure conversions of the video signal in digital television [4], [5], and video motion estimation and compensation by hierarchical search methods [2]. Most of the existing subsampling methods are based on pixel neighborhood operations, such as the computation of a statistical measure of the local intensity values (e.g., the mean) within each image block. The reduced images may contain significant distortion, which can be eliminated by applying post-processing techniques [6]. However, the associated processing cost is often quite high. Moreover, although the image may be modeled as a bandlimited signal, image conditioning, which commonly involves lowpass filtering, is usually performed before subsampling [7], [8]. Of course, much of high frequency information is consequently lost. To reduce the distortion introduced by averaging and to minimize the information loss introduced by lowpass filtering, we here employ feedforward artificial neural networks (FANN's). Such networks can perform high speed parallel pro-

Manuscript received September 12, 1997; revised October 29, 1999. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Contract 06P-0187668. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jan P. Allebach.

A. Dumitras was with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver BC V6T 1Z4, Canada. She is now with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ont. M55 3G4, Canada.

F. Kossentini is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver BC V6T 1Z4, Canada.

Publisher Item Identifier S 1057-7149(00)02670-1.