A reprint from

JOURNAL OF

Electronic Imaging

April 1995

ISSN 1017-9909

IMPROVED CHEN-SMITH IMAGE CODER

Eduardo M. Rubino

Universidade de Brasília
Departamento de Engenharia Elétrica
CP 04591
70919-970 Brasília, DF, Brazil

Ricardo L. de Queiroz

Xerox Webster Research Center
800 Phillips Road
Building 128-27E
Webster, New York 14580

Henrique S. Malvar

PictureTel Corporation
222 Rosewood Drive
M/S 635
Danvers, Massachusetts 01923
E-mail: malvar@pictel.com

Journal of Electronic Imaging 4(2), 151–160 (April 1995).

Improved Chen-Smith image coder

Eduardo M. Rubino
Universidade de Brasília
Departamento de Engenharia Elétrica
CP 04591
70919-970 Brasília, DF, Brazil

Ricardo L. de Queiroz
Xerox Webster Research Center
800 Phillips Road
Building 128-27E
Webster, New York 14580

Henrique S. Malvar
PictureTel Corporation
222 Rosewood Drive
M/S 635
Danvers, Massachusetts 01923
E-mail: malvar@pictel.com

Abstract. *A new transform coder based on the zonal sampling strategy, which outperforms the JPEG baseline coder with comparable computational complexity, is presented. The primary transform used is the 8-× 8-pixel-block discrete cosine transform, although it can be replaced by other transforms, such as the lapped orthogonal transform, without any change in the algorithm. This coder is originally based on the Chen-Smith coder, therefore, we call it an improved Chen-Smith (ICS) coder. However, because many new features were incorporated in this improved version, it largely outperforms its predecessor. Key approaches in the ICS coder, such as a new quantizer design, arithmetic coders, noninteger bit-rate allocation, decimated variance maps, distance-based block classification, and human visual sensitivity weighting, are essential for its high performance. Image compression programs were developed and applied to several test images. The results show that the ICS performs substantially better than the JPEG coder.*

1 Introduction

Transform coders[1–5] have been widely used for image compression. In particular, the Joint Photographic Experts Group (JPEG)[6,7] baseline algorithm is now widely used for lossy compression of gray-scale and color images. In transform image coding, the transform is only a part of the overall compression scheme, because the coding process may involve more processing to implement quantizers, buffers, variable-length coders, etc.[1–7] The transform by itself does not imply any compression, but it makes easier the task of discarding signal information in the transform domain without affecting much of the subjective quality of the reconstructed image. Although various discrete transforms have been investigated for application to image coding, only the discrete cosine transform (DCT) has emerged as the most practical and efficient transform.[8,9] Recently, the lapped orthogonal transform has been extensively simulated and has proven to be advantageous over the DCT in terms of reducing blocking artifacts, mainly at low bit rates.[10,11] Details of transform coding of images can be found in Refs. 1 through 13. The spectrum of applications of image compression includes different disciplines such as medical imaging, remote sensing, consumer electronics, printing and publishing, defense, television, sports, communications, storage, etc.

Commonly, in the block transform coding approach, the image is decomposed into blocks of $M \times M$ samples (picture elements or pixels), and each block is transformed using 2-D transforms obtained from a separable transform applied to rows and columns of the block. The image samples are denoted as $x(n_1,n_2)$ for $0 \leq n_1 \leq N_1 - 1$ and $0 \leq n_2 \leq N_2 - 1$, where $N_1 \times N_2$ are the dimensions of the image. Assume M

Paper 94-001 received Jan. 20, 1994; revised manuscript received Nov. 28, 1994; accepted for publication Nov. 29, 1994.
1017-9909/95/$6.00.

divides both N_1 and N_2. Each image block is thus represented by the samples $x_{ij}(m,n)=x(mM+i,\ nM+j)$ for $0\leq(i,j)\leq M-1$, with $i,j,m,$ and n integers. The block is also referred to as having position (m,n), and the sample position inside the block is given by (i,j). Similarly, the notation for the sample in position (i,j) of a transformed block is $X_{ij}(m,n)$, for $0\leq(i,j)\leq M-1$, $0\leq m\leq N_{B1}-1$, and $0\leq n\leq N_{B2}-1$, where N_{B1} and N_{B2} are the number of blocks in the vertical and horizontal directions, respectively. We also denote $N_B=N_{B1}N_{B2}$ as the total number of blocks in the image. The transformed samples are referred to here as coefficients, and the image samples are referred to as pixels. The samples $X_{00}(m,n)$ are called the dc coefficients, and the remaining M^2-1 elements of each block are called the ac coefficients. Thus, the ac indices are those in the set $\Psi=\{ij\,|0\leq i,j\leq M-1,(i,j)\neq(0,0)\}$. The process of block segmentation is illustrated in Fig. 1.

Figure 2 illustrates the method for image compression through transforms. This method transforms the image, quantizes coefficients, and encodes the quantizer output to form a bit stream that will be transmitted or stored. The inverse operation is carried out to reconstruct the image. The compression is achieved by coding more efficiently (with more bits) the coefficients that are more important, i.e., that carry more energy of the input block. In adaptive transform coders, blocks with more ac energy are allocated more bits than blocks with low ac content.

The main approaches used to encode the coefficients are known as thresholding and zonal sampling.[5] Thresholding is the main philosophy behind JPEG.[5,6,12] In it, all coefficients are quantized, and all quantized coefficients are input to a binary encoding procedure to reduce the redundancy of the data through run-length and variable-length coders. The zonal sampling strategy is based on the selection of some coefficients for transmission, while discarding the remaining coefficients.[5,13] A first step for achieving signal compression is to allocate a different number of bits to each region. Regions with less concentration of energy receive fewer bits, whereas regions with greater concentration of energy receive more bits. The Chen-Smith (CS) coder[13] and the improved Chen-Smith (ICS) coder are based on an adaptive zonal sampling scheme.

This paper is organized as follows. Section 2 is devoted to the explanation of the basic concepts of the CS and ICS coders. Section 3 is concerned with the features incorporated into the ICS coder, which will help explain its better performance. Section 4 presents results of tests using the ICS coder and relating it to both CS and JPEG coders. Finally, Sec. 5 contains our conclusions.

2 Coder Outline

In this section we review the basic concepts common to the CS and ICS coders and also give an overview of the specific improvements that characterize the ICS coder.

2.1 *Chen-Smith Coder*

Chen and Smith[13] devised a technique to distribute different bit allocations to different blocks of the signal, according to the ac energy of the blocks. The brief description of the algorithm follows:

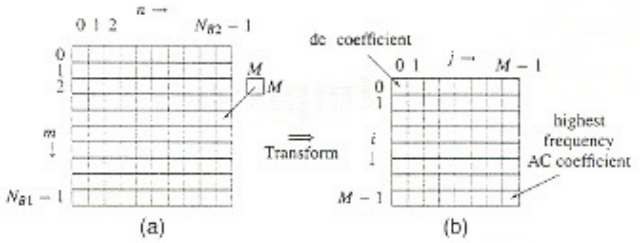


Fig. 1 Illustration of the position of the transformed coefficients: (a) An $N_1\times N_2$ image is divided into $N_{B1}\times N_{B2}$ blocks, each of the size $M\times M$. A 2-D transform is applied to each block, resulting in M^2 transform coefficients. A transformed block is illustrated in (b), indicating the position of the dc coefficient and of the highest frequency ac coefficient.

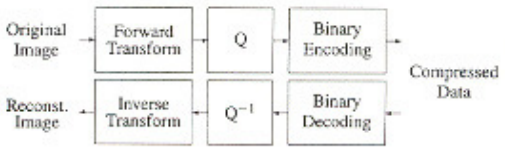


Fig. 2 Basic concepts of transform coding.

1. Transform the image using blocks of $M\times M$ pixels.
2. Quantize and code separately the dc coefficients using uniform quantizers.
3. Compute the ac energy of each block as

$$E(m,n)=\sum_{(i,j)\in\Psi}X_{ij}^2(m,n)\ . \tag{1}$$

Sort the energies, and classify the blocks (in sorted order) into N_C equally populated classes. Thus, there will be N_B/N_C blocks in each class. Construct the class map $C(m,n)$ with the classification of each block, for example, $C(m,n)=k$ if block (m,n) belongs to the class k $(k=1,\ ...,\ N_C)$.

4. For all blocks belonging to the same class, compute the average variance of the transform coefficients and then their standard deviations. Construct N_C standard deviation maps with the standard deviation of the coefficients, found from

$$\sigma_k^2(i,j)=\sum_{m,n\in\Phi_k}\sum X_{ij}^2(m,n)\quad\text{for }(i,j)\in\Psi\ , \tag{2}$$

where $\Phi_k=\{m,n\,|\,C(m,n)=k\}$ is the set of blocks that belong to class k.

5. Merge all N_C ac standard deviation maps and decide the bit allocation, using the standard log-variance rule.[4,5] From all $\sigma_k^2(i,j)$, find an equivalent set of rates $R_k(i,j)$ to minimize the distortion for a given bit budget. A constraint is imposed that each $R_k(i,j)$ is an integer lying between 0 and $B_{\max}$, where $B_{\max}$ is the maximum number of bits for which there is a quantization table available. Create N_C bit-allocation maps with a one-to-one correspondence with the elements of the standard deviation maps.

6. Reestimate the standard deviations using the bit-allocation maps:

$$\hat{\sigma}_k(i,j) = c\ 2^{R_k(i,j)-1} \quad \text{for } 1 \le k \le N_C \quad (i,j) \in \Psi \ , \tag{3}$$

where c is a normalization factor. We suggest choosing c as the maximum $\sigma_k(i,j)$, for which $R_k(i,j) = 1$, to avoid excessive clipping.

7. Send the class map, the normalization coefficient c, and the bit-allocation maps as side information. For example, if we chose $N_C = 8$ and $B_{\max} = 7$ we can encode the maps with 3 bits/sample, and c can be efficiently quantized with 16 bits or less.

8. Quantize, encode, and send all the coefficients, using the reestimated variances. A coefficient $X_{i,j}(m,n)$ [block (m,n)], which belongs to class k [$C(m,n) = k$], is scaled [divided by $\hat{\sigma}_k(i,j)$], applied to a quantizer with $2^{R_k(i,j)}$ levels, and encoded with $R_k(m,n)$ bits. If $R_k(i,j) = 0$, the particular coefficient is not transmitted.

The decoder may first decode the side information and the dc coefficients, the class map, the bit-allocation maps, and the normalization factor c. From them, the decoder can reconstruct the standard deviations used to scale the quantizers as in Eq. (3). With the maps reconstructed, and with the knowledge of the transmission order, the decoder can exactly determine the position of the incoming coefficient, the class of its block, how many bits were assigned to it, and the variance used for quantization. Therefore, the receiver can decode the coefficients, apply an inverse transform, and obtain the reconstructed image.

If we use b_1 bits to encode each bit-allocation map sample, b_2 bits to encode each class map sample, b_3 bits to encode c, and b_{dc} bits to encode each dc coefficient, then the total number of bits used to encode the overhead, the dc coefficients, and the ac coefficients (B_{ov}, B_{dc}, and B_{ac}, respectively) would be

$$B_{\text{ov}} = N_C(M^2 - 1)b_1 + N_B b_2 + b_3 \ , \tag{4}$$

$$B_{\text{dc}} = N_B b_{\text{dc}} \ , \tag{5}$$

$$B_{\text{ac}} = R_{\text{bpp}} N_B M^2 - B_{\text{dc}} - B_{\text{ov}} \ , \tag{6}$$

where R_{bpp} is the overall bit rate to encode the whole image in bits per pixel. The budget of bits available for the rate-allocation procedure is given by

$$\sum_{ij \in \Psi} \sum_{k=1}^{N_C} R_k(i,j) = B_{\text{ac}} \frac{N_C}{N_B}(M^2 - 1) \ . \tag{7}$$

For example, a 256-$\times$256-pixel image is compressed using $N_C = 8$ classes to a rate of 1 bit/pixel. There are $N_B = 1024$ blocks in the image, and we assume $B_{\max} = 7$ and $b_{\text{dc}} = 7$. Hence, $b_1 = 3$, $b_2 = 3$, $b_3 = 16$, and the overhead is responsible for 7% of the total bit rate, the dc coefficients for 11%, and the remaining 82% (0.82 bits/pixel) is spent with the ac coefficients. One disadvantage of this coder arises from the reestimation procedure, which could be avoided with the transmission of the standard deviation maps instead of the bit allocation maps. However, if we code each standard deviation with 16 bits, the part allocated for overhead would increase to 17%.

2.2 *Improved Chen-Smith Coder*

The ICS coder incorporates key features to overcome most of the problems present on the original CS coder. These features, which are explained in the next section, are the following.

- Distance-based block classification is an option.
- The standard deviation maps are sent to the receiver instead of the bit-allocation maps. This allows the receiver to have precise estimates of the variances, largely reducing quantization mismatches. However, we do transmit the maps, expending an amount of bits comparable to or lower than the amount needed to encode the bit allocation maps.
- Because the receiver knows the standard deviations, we can weight the standard deviation maps used for bit allocation, leaving the estimates used for quantization intact, and without increasing overhead for this step.
- Quantizers are designed following the Gaussian probability density function (PDF), for reasons we will explain later. The quantizers are optimized constrained to their output entropy, and the Lloyd-Max algorithm is not used. Arbitrary noninteger entropy values can be used.
- A bit rate allocation method is applied to allocate fractional and non-negative entropic bit rates to each coefficient in a particular class. Overall distortion is minimized because of the quantizer design algorithm.
- Arithmetic coding is applied to all quantized coefficients.

As for the ICS coder, a brief description of the algorithm follows:

1. Transform the image using blocks of $M \times M$ pixels.
2. Quantize and code separately the dc coefficients using uniform quantizers.
3. Classify blocks either using ac-energy-based or distance-based methods.
4. For each class, compute the standard deviation maps.
5. Reestimate the standard deviations using decimation/interpolation of the original map.
6. Send the class map, the standard deviation maps, and the human visual sensitivity (HVS) model sampling frequency f_s.[14,15] The standard deviation maps are decimated in two dimensions, as we will discuss later.
7. Weight the standard deviation maps using an HVS model and decide the bit allocation. Allow noninteger non-negative bit rates according to the quantizers available.
8. Quantize, encode, and send all the coefficients using the reestimated variances.

At the receiver side we first decode the side information and the dc coefficients. Then, a bit-allocation routine identical to the one used in the transmitter side is used. With the maps reconstructed, and with the knowledge of the transmission order, the receiver can decode the coefficients, apply an inverse transform, and obtain the reconstructed image.

The amount of bits spent with overhead and with the coefficients is similar to the number found for the CS coder, except that the standard deviation maps are encoded in place of the bit-allocation ones. However, as we discuss later, this is done without increasing the overhead.

3 Features

We now explain in more detail the key features introduced in the ICS coder and how these features enhance coder performance.

3.1 *Distance-Based Classification*

Define the distance between two blocks as

$$\beta(m_1,n_1,m_2,n_2)=\sum_{ij\in\Psi}[\,|X_{ij}(m_1,n_1)|-|X_{ij}(m_2,n_2)|\,]^2\ . \tag{8}$$

In the ICS, we seek to minimize the distance among all blocks that belong to a particular class. Thus, a vector quantization (VQ) codebook design procedure is carried.[4,16]

In Table 1 we have computed the signal-to-noise ratio (SNR) in decibels using both classification procedures (ac-energy-based and distance-based) for the test image "Lena," with several values of N_C and using the DCT over blocks of 8×8 pixels. The SNR was computed as

$$\mathrm{SNR}=10\ \log_{10}\left\{\frac{\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x^2(n_1,\,n_2)}{\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}[x(n_1,n_2)-\hat{x}(n_1,n_2)]^2}\right\}, \tag{9}$$

where $x(n_1,n_2)$ and $\hat{x}(n_1,n_2)$ represent the original and reconstructed images, respectively. We can see from Table 1 that distance-based classification works better than ac energy classification as the number of classes increases. Note that the overhead necessary to transmit the classes has been taken into account in the bit rates. Therefore, it does pay off to use a higher number of classes. The price to be paid is in the increased computational complexity of the classification procedure.

3.2 *Side Information*

The transmission of the standard deviation maps instead of the bit allocation maps allows the receiver to have accurate standard deviation maps, so that different maps can be used for bit allocation and for quantization. The map used for bit allocation is weighted by an HVS model, so that more bits can be given to more important coefficients, in a subjective sense. This cannot be done in the original CS coder. Because the bit allocation maps are transmitted as side information, if they are weighted beforehand, the standard deviation used to dequantize the coefficient would also be affected, causing severe quantizer mismatch in the reconstruction process. However, we cannot waste much information to send accurate maps, as we discussed earlier. The map of standard deviations in a block is a reasonably smooth function. Therefore, one can decimate this map by discarding some samples and, in most cases, the deleted samples can be very well approximated by interpolation. We retain the samples indicated in Fig. 3, and the interpolation can be a simple average of the nearest neighbors. The remaining samples are uniformly quantized and encoded using an adaptive method. Samples are scanned in a reversed zigzag path (from higher frequency coefficients to lower frequency ones), and it is expected to have high-frequency coefficients with less amplitude than coefficients with lower frequency. Then, the number of bits given to them is increased as we go along the path. The average bit rate to transmit these samples is smaller than 3 bits/sample. Thus, for $B_{\max}=7$ we can have some savings compared to the original CS method. Note that the bit allocation map is no longer transmitted. The decoder runs its own rate-allocation procedure based on the standard deviation maps available, and both encoder and decoder remain synchronized.

Table 1 SNR (in dB) results using image "Lena," DCT, and 8-×8-pixel blocks. B is the bit rate in bits per pixel achieved.

Image Lena, 256 × 256 pixels										
	AC Energy classification					Distance-based classification				
	Number of classes					Number of classes				
B	2	4	8	16	32	2	4	8	16	32
0.4	22.8	23.0	22.8	22.2	21.2	22.3	22.6	22.8	22.6	21.5
0.6	25.1	25.5	25.4	25.1	24.5	24.2	25.0	25.3	25.6	25.2
0.8	27.0	27.4	27.3	27.1	26.7	25.9	26.7	27.4	27.8	27.6
1.0	28.4	29.0	29.0	28.9	28.6	27.2	28.1	28.9	29.6	29.5
1.5	31.4	32.2	32.3	32.2	32.0	29.8	31.0	31.9	32.9	33.0

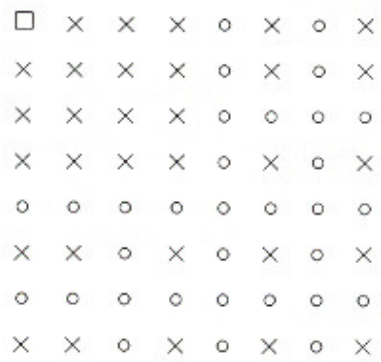


Fig. 3 Transmission of standard deviation maps for each block in a class. ⊠ represents samples retained in an 8×8 block, ○ represents discarded samples, and □ represents the dc coefficient, for which no deviation is computed.

3.3 *Human Visual Sensitivity Weighting*

The human eye has discriminative sensitivity to different spatial frequencies. The HVS weighting array is meant to devise the relative importance of each transformed coefficient for the reconstructed image, in the subjective viewpoint of a

human observer.[14–21] If the 1-D weighting model is given by $w(f)$, where f is the radial frequency in cycles per degree of the visual angle subtended, then the HVS array composed by elements η_{ij} for $0 \leq (i,j) \leq M-1$ is found by[15]

$$\eta_{ij} = \frac{w(f_{ij})}{\alpha^2(i)\alpha^2(j)} , \tag{10}$$

where $\alpha(0) = 1/\sqrt{2}$, $\alpha(n) = 1$ $(n > 0)$,

$$f_{ij} = \frac{f_s(i^2+j^2)^{1/2}}{2M} , \tag{11}$$

and f_s is a sampling frequency parameter, which can be varied according to the ratio of the distance of the viewer to the screen width and according to the number of pixels displayed per line[14,15] and is passed to the receiver as side information. If we set $f_s = 0$, we are actually turning off the HVS weighting process, because all coefficients would receive equal weights. Furthermore, the model is relative, such that the maximum value in the array can be set to unity. We used the following 1-D model[15]:

$$w(f) = 2.46(0.1 + 0.25f)\exp(-0.25f) . \tag{12}$$

Then we use $\hat{\sigma}_{ij}\eta_{ij}$ as input to the bit-rate-allocation process instead of using only the standard deviations $\hat{\sigma}_{ij}$.[14,15,21] In this way, the rate-allocation process will save bits from less important coefficients to give to subjectively more important ones, and, as we discussed before, $\hat{\sigma}_{ij}$ would still be available for scaling the coefficients in the quantization process.

3.4 *Design of the Quantizers*

The CS coder uses quantizers optimized by the Lloyd-Max method[4,5] assuming a Laplacian PDF. Although this is a common practice for quantizing DCT coefficients, it is not a good technique in coders such as CS or ICS.

It has been reported that the distribution of the DCT coefficients follows approximately a Laplacian PDF.[2,9,13] However, this fact has little relevance to us, because the meaningful PDF (of the sample input to the quantizers) is the conditional PDF given the class index. Suppose we use a large number of classes, so that fewer blocks would be included in one class, and suppose we were using a distance-based classification, and examining the PDF of all samples $X_{ij}(m,n)$ for a given (i,j) and $(m,n) \in \Phi_k$. These samples will have an estimated standard deviation $\hat{\sigma}_k(i,j)$. If $\hat{\sigma}_k(i,j)$ is large, we expect the coefficients to have large values, due to the classification procedure, and not values close to zero as in the Laplacian model. The resulting conditional PDF approximates a sum of two Gaussian PDFs, as shown in Fig. 4. The reason for the two lobes in Fig. 4 is due to the uncertainty about the coefficient sign. As the number of classes decreases, the two Gaussian functions are merged and the resulting PDF becomes approximately Gaussian. Similar reasoning applies to the use of ac energy classification; however, as many energy distribution patterns can be found in different blocks with similar ac energy, the conditional PDF does not present the two lobes, but the Gaussian PDF fits well to most input distributions tested. Note that low-frequency coefficients are generally larger, and the high-frequency components are generally small and do not influence the classification process much. Thus, the conditional PDF is not as important to them, and they are expected to follow more closely the Laplacian PDF. However, they also have lower energy concentration and lower importance to the reconstruction of the image. Furthermore, the use of a Gaussian PDF leads to quantizers that are more robust against PDF mismatches, compared to the Laplacian PDF.[5] These facts led us to choose the Gaussian PDF as the model for quantizer design.

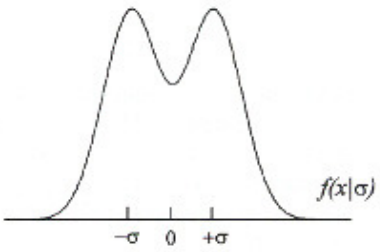


Fig. 4 Typical conditional PDF for a high number of classes and the estimated standard deviation σ.

Assume the range of real numbers is divided into L non-overlapping segments I_k $(1 \leq k \leq L)$, with each segment representing one quantization level. So a sample with amplitude $X = x$ and estimated standard deviation σ will be reconstructed as $\hat{X} = x_i$ if $(x/\sigma) \in I_i$, where x_i is the reconstruction value of the i'th quantizer level, and i is the symbol associated with the quantizer output. If the input sample (X/σ) has a PDF of $f(x)$, the probability of occurrence of the i'th symbol is

$$p_i \equiv \Pr[(X/\sigma) \in I_k] = \int_{I_k} f(\lambda)\, d\lambda , \tag{13}$$

and the entropy of the quantizer output is

$$H = -\sum_{i=1}^{L} p_i \log_2 p_i . \tag{14}$$

The average distortion D allowed by the quantizer is defined as the standard deviation of the quantizer error σ_q as

$$D = \sigma_q, \text{ with } \sigma_q^2 = \sum_{i=1}^{L} \frac{\int_{\lambda \in I_i} (\lambda - x_i)^2 f(\lambda)\, d\lambda}{p_i} . \tag{15}$$

The Lloyd-Max quantizer design technique[4,5] is optimal in a sense that it will minimize D for a given PDF and L, thus finding

$$\min_{I_i} D|L, f(x) . \tag{16}$$

Because we will apply efficient entropy coding procedures to the quantizer output, we are interested in minimizing the distortion subject to a particular output entropy, independent of the number of levels achieved. This is the entropy-constrained version of the Lloyd-Max algorithm,[22] and it is used in the design of the quantizers in the ICS coder, where the algorithm searches for

$$\min_{l_i} D|H, f(x) \quad . \tag{17}$$

The main advantage of using entropy-constrained scalar quantizers (ECSQs) is that we can obtain any desired entropy. This is important mainly at low bit rates, where the available entropies for Max quantizers are too few. For example, below 2 bits/sample there exist only three Max quantizers, with entropies of 1.0, 1.54, and 1.91 bits/sample. Therefore, with ECSQs we can get bit allocations closer to the true optimum.

We have designed 23 quantizers with entropy rates ranging from $H=0$ to $H=5.75$ in steps of 0.25 bits, i.e., 0.25, 0.50, 0.75, ..., 5.75 bits. The number of levels L ranges from 3 to 75. As is usual with ECSQs, the reconstruction levels of the quantizers are approximately uniformly distributed, except near the origin, where they are further spaced [that is basically how the entropy H gets to be much lower than $\log_2 L$ (see Ref. 22)]. The plot of D as a function of H for the set of quantizers used in this paper is shown in Fig. 5.

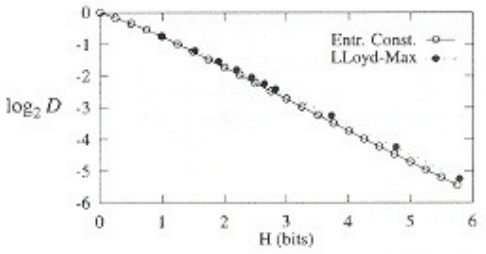


Fig. 5 Distortion $D\times$entropy (H) function for the optimal ECSQs used in this paper and for Lloyd-Max quantizers. With ECSQs, entropies below one bit per sample can easily be obtained.

3.5 *Quantizer Allocation*

We have available a set of N_Q quantizers, each one associated with an average normalized distortion (due to unit variance input), and a quantizer entropy value forming a pair (H_i, D_i) for $0\leq i\leq N_Q$. There are actually N_Q+1 quantizers, because for the first quantizer it is assumed $(H_0=0, D_0=1)$, and this quantizer is included just to simplify the presentation. This means that a particular coefficient assigned to it is not transmitted and is reconstructed as zero.

Let $QN[x]$ be a function returning the quantizer number i such that D_i is the distortion (in the set of available quantizers) closest to the real number x. There are $K=N_C(M^2-1)$ different coefficient classes and ac frequency bands to which we have to assign quantizers. If all classes are merged and coefficients are displaced lexicographically, we can say that we have variables $(X_1, ..., X_K)$ with respective standard deviations given by $(\sigma_1, ..., \sigma_K)$ to which we have to assign quantizers $(q_1, ..., q_K)$ for $0\leq q_i\leq N_Q$. An optimal allocation will distribute the same distortion to all X_i, unless the required distortion is greater than σ_i, in which case the total distortion is σ_i.[4,5]

Constrained distortion. If X_i is allocated* to a quantizer with distortion $d\leq 1$, the total distortion suffered by X_i is $D=d\sigma_i$. So for a given distortion parameter θ, we have

$$q_i=QN(\theta/\sigma_i) \qquad \text{for } i=1, ..., K \quad . \tag{18}$$

Constrained bit rate. We can interact θ until the bit rate is close enough to the total budget bit rate B available. Let (r_i, d_i) be the entropy/distortion pair of the quantizer assigned to X_i. So we have a recursion as

1. $r_i=H[QN(\theta/\sigma_i)] \qquad$ for $i=1, ..., K$
2. if $|(\sum_{i=1}^{K} r_i)-B|<\varepsilon$, then return, or else adjust θ and continue, where ε is a small control number and $H(i)\equiv H_i$.

The solution is optimal provided that $N_Q\to\infty$, the estimated standard deviations are accurate, the distribution of all X_i follows the assumed PDF, and the quantizer output is encoded by a perfect entropic coder. However, the above algorithm is our "best shot" given the set of quantizers available. We are using quantizers whose output entropy is spaced by quarter-bit intervals, in order to approximate ideal conditions.

3.6 *Arithmetic Coding*

Arithmetic coders[23,24] are applied to the quantizer output to provide near perfect entropic coding. To do that more efficiently, we applied one arithmetic coder for all coefficients, instead of using one arithmetic coder for each quantizer. As each coefficient is quantized, new information regarding L and $p_i(1\leq i\leq L)$ for the particular quantizer in use is loaded. The output bit stream will convey information from all quantizers together, but, because we know precisely the order in which each quantizer was used, we can recover all coefficients. This ensures that each quantizer output is encoded to a rate near to its entropy, and, compared to a set of N_Q distinct arithmetic coders operating in parallel, it simplifies implementation.

3.7 *Other Transforms*

Another point favoring the use of the ICS coder is its independence of the transform used. No parameter was designed for the DCT,† and most of the parameters are either computed or assumed. Therefore, we can easily replace the DCT by transforms with better performance such as the LOT or the extended lapped transforms[11] without any algorithm change.

4 Image Compression Performance

We have devised an image compressing prototype based on the ICS coder, written in C and using a PC 486 DX-33 as the platform. The prototype accepts input parameters, which allow us to change number of classes, image size, transform used, classification method, target bit rate or distortion, and the HVS parameter f_s. For a fair performance comparison, we also developed a JPEG baseline coder using similar routines. We used the quantization matrices suggested as an example in the JPEG baseline recommendation[6]; this is usual in many JPEG implementations. In our bit-rate computations

*If X_i is allocated to quantizer 0, $d=D_0=1$ and total distortion is σ_i.

†An exception is the HVS function, which would slightly change. However, this function can be easily changed to accommodate other transforms.[14]

for JPEG we have not included the overhead necessary to transmit the quantization matrices. A truly JPEG-compliant encoder must include those matrices in the encoded bit stream, and so its performance would be somewhat worse than that of our JPEG encoder.

Although the ICS coder incorporated many sophisticated techniques, such as arithmetic coding, optimal bit allocation, etc., JPEG is still not much faster than the ICS coder. For example, decoding of 256×256 pixels takes 2 s with the JPEG algorithm and 3.2 s with the ICS (using the DCT and ac energy classification). The reason for this relatively small difference in speed is because the most complex operations in the ICS coder, such as the quantizer allocation, etc., are carried over a small set of data. The time-consuming operations are those applied over the whole image, such as the DCT, classification, and arithmetic coding. Also, for low bit rates, only a reduced set of coefficients is actually quantized and coded.

In our test, we have comparisons of classification methods for the ICS coder in Table 1. The best entries in that table (for each bit rate and for both classification methods) were selected and used in Table 2 to compare the performance of the ICS coder with the CS coder and the JPEG coder. From these results, we can clearly see the best performance of the ICS coder. We also carried other objective comparisons (using several test images) between the ICS (using ac-energy-based or distance-based classification), and the JPEG coder, as shown in Fig. 6. In these simulations, we have turned off the HVS weighting by setting $f_v = 0$. Note the greater performance difference for smaller images (256×256 pixels). In this size range, for DCT-based coders using scalar quantization, and for a detailed monochrome image, a reasonable compromise between image quality and compression generally lies between 0.6 and 1.2 bit/pixel. In this range, the ICS coder is up to 2.5 dB better than the JPEG.

Figure 7 shows the resulting reconstructed images using the ICS coder for images "Lena" and "building" at 0.8 bit/pixel, with $f_v = 32$, which is well suited for someone observing the image at a distance four to six times longer than its width, assuming 256×256-pixel images. This situation is typical of someone working in front of a computer where the image is displayed on a medium-to-high resolution monitor. Also, in Fig. 7, a comparison is made between ICS and JPEG coders, for the same bit rates.

5 Conclusions

A new transform coder based on the well-known Chen-Smith coder is developed, which outperforms the JPEG baseline coder. Our tests have shown that this compression gain comes with only a small expense of compression speed. Essential factors for its higher performance (compared to the CS coder) are the new quantizer design, the use of arithmetic coders, noninteger bit-rate allocation, decimated variance maps, distance-based block classification, and HVS weighting. For low bit rates, blocking effects are present in both JPEG and ICS (with HVS weighting switched off) as is common with DCT-based coders. However, as we turn on the HVS weighting, these blocking effects are largely reduced or eliminated. In fact, all images coded with the ICS coder presented noticeably better subjective quality when compared with correspondent images coded with the JPEG coder.

Table 2 Objective comparison of performance (SNR in decibels) among different image coders, using 256-$\times$256-pixel image "Lena," and 8×8 DCT. ICS entries are for the best results in previous tables for distance-based and ac energy classifications (ICSdb and ICSac, respectively), and bit rate is given in bit per pixel.

Bit-rate	JPEG	CS	ICSdb	ICSac
0.4	21.3	21.3	22.8	23.0
0.6	23.7	23.1	25.6	25.5
0.8	25.4	24.5	27.8	27.4
1.0	26.9	25.8	29.6	29.0

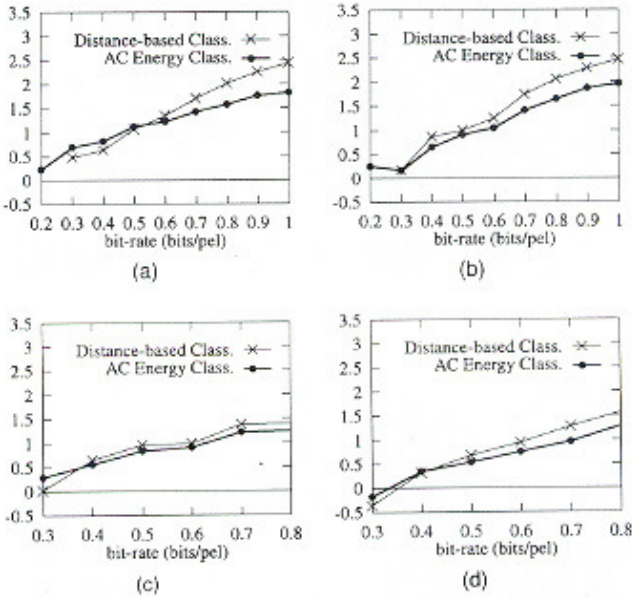


Fig. 6 Difference in SNR_{dB} between ICS coder and JPEG coder, using different images, DCT, and both classification methods. For the distance-based classification method we selected $N_C = 16$, while for ac energy classification we used $N_C = 4$. The values plotted correspond to $SNR_{dB}(ICS) - SNR_{dB}(JPEG)$ for different bit rates without HVS weighting. (a) 256-$\times$256-pixel image "Lena." (b) 256-$\times$256-pixel image "building." (c) 512-$\times$512-pixel image "jet." (d) 512-$\times$512-pixel image "locomotive."

Although the ICS coder presents significant performance gains over the JPEG, there are some points that could be added to our current prototype in order to improve its performance, such as better coding of the dc coefficients, which now are coded using unifirom quantization and a fixed-length code. The dc coefficients are coded first with the same quantizer, and this procedure is independent of the desired bit rate, which explains the performance of the ICS coder against the JPEG coder for very low bit rates in Fig. 6. Techniques such as differential pulse code modulation (with a quantizer whose step sizes can be increased as we decrease the bit rate) and arithmetic coding can be used to efficiently reduce the budget of bits spent for the transmission of the dc coefficients. This point will receive attention in the future.

Acknowledgment

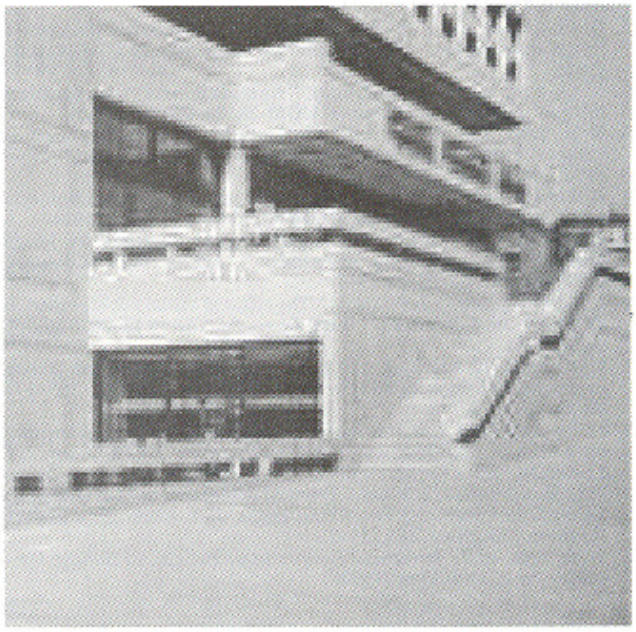
This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, under grant 200.804-90.

Fig. 7 Reconstructed images using ICS coder, DCT, 256-×256-pixel images, $f_s = 32$, and distance-based classification for 16 classes. (a) Original image "Lena" and (b) its reconstructed version at 0.8 bit/pixel. (c) Zoom of (b). (d) Same result as in (c) with the JPEG coder. (Continued on next page)

(e)

(f)

(g)



(h)

Fig. 7 (Continued from previous page.) (e) Original image "building" and (f) its reconstructed version at 0.8 bit/pixel. (g) Zoom of (f). (h) Same result as in (g) with the JPEG coder.

References

1. A. N. Netravali and B. G. Haskell, *Digital Pictures, Representation and Compression,* Plenum Press, New York (1988).
2. R. J. Clarke, *Transform Coding of Images,* Academic Press, Orlando, FL (1985).
3. M. Rabbani and P. W. Jones, *Digital Image Compression Techniques,* SPIE Optical Engineering Press, Bellingham, WA (1991).
4. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression,* Kluwer Academic, Hingham, MA (1992).
5. N. S. Jayant and P. Noll, *Digital Coding of Waveforms,* Prentice-Hall, Englewood Cliffs, NJ (1984).
6. W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard,* Van Nostrand Reinhold, New York (1993).
7. Final test for ISO/IEC DIS 10918-1, Info. Technology *"Digital compression and coding of continuous tone still images,"* Part 1: Requirements and guidelines (Jan. 1992); Part 2: Compliance testing, CD 10918-2, (Dec. 1991).
8. K. R. Rao, Ed., *Discrete Transforms and their Applications,* Van Nostrand Reinhold, New York (1985).
9. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications,* Academic Press, San Diego, CA (1990).
10. H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-37,** 553–559 (Apr. 1989).
11. H. S. Malvar, *Signal Processing with Lapped Transforms,* Artech House, Norwood, MA (1992).
12. W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.* **COM-32,** 225–232 (March 1984).
13. W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.* **COM-25,** 1285–1292 (Nov. 1977).
14. R. L. de Queiroz and K. R. Rao, "HVS weighted progressive transmission of images using the LOT," *J. Electron. Imaging* **1**(3), 328–338 (July 1992).
15. B. Chitprasert and K. R. Rao, "Human visual weighted progressive image transmission," *IEEE Trans. Commun.* **38,** 1040–1044 (July 1990).
16. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **COM-28,** 84–95 (Jan. 1980).
17. K. H. Tzou, T. R. Hsing, and J. G. Dunham, "Applications of physiological human visual system model to image compression," *Proc. SPIE* **504,** 419–424 (1984).
18. H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.* **COM-32,** 1316–1322 (Dec. 1984).
19. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.* **COM-33,** 551–557 (June 1985).
20. J. L. Mannos and D. J. Sakrison, "The effect of visual fidelity criterion on the encoding of images," *IEEE Trans. Inf. Theory* **IT-20,** 525–536 (July 1974).
21. K. N. Ngan, K. S. Leong, and H. Singh, "Cosine transform coding incorporating human visual system model," *Proc. SPIE* **707,** 165–171 (Sep. 1986).
22. N. Farvardin and J. W. Modestino, "Optimal quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inf. Theory* **30,** 485–497 (May 1984).
23. P. G. Howard and J. S. Vitter, "Practical implementations of arithmetic coding," in *Image and Text Compression,* J. A. Storer, Ed., Kluwer Academic, Hingham, MA (1992).
24. W. H. Press et al., *Numerical Recipes in C,* 2nd ed., Cambridge University Press, New York (1992).

Eduardo M. Rubino received his BSc from Universidade de Brasilia, Brazil, in 1986, and the MSc from Kyushu Institute of Technology, Japan, in 1989, both in electrical engineering. From 1989 to 1990 he was a consultant for Unitor in Japan. From 1990 to 1993 he was with the Department of Electrical Engineering of Universidade de Brasilia as a visiting professor. In 1993 he became a research associate at Universidade de Brasilia and also a consultant in computer systems and networks. His main interests are image compression, data networks, and symbolic computation.

Ricardo L. de Queiroz received the BS degree from Universidade de Brasilia, Brazil, in 1987, the MS degree from Universidade Estadual de Campinas, Brazil, in 1990, and the PhD degree from University of Texas at Arlington, in 1994, all in electrical engineering. From 1990 to 1991, he was with the DSP research group at Universidade de Brasilia as a research associate. In 1993 he received the Academic Excellence Award from the Electrical Engineering Department of the University of Texas at Arlington and in 1994 he was a teaching assistant at the same university. He joined Xerox Corporation in August 1994, where he is currently a member of the research staff at the Advanced Color Imaging group. His research interests are multirate signal processing, filter banks, image and signal compression, and image databases.

Henrique S. Malvar holds a PhD degree in electrical engineering from the Massachusetts Institute of Technology (1986). From 1979 to 1983 he was with the faculty of the Universidade de Brasilia, Brazil, where he was the head of the Digital Signal Processing Group. From 1985 to 1987 he was a consultant for PictureTel Corporation, Danvers, Massachusetts; he has now rejoined PictureTel as director of research. His main areas of interest are image and audio compression and enhancement. He has several publications in these areas, including the books *Signal Processing with Lapped Transforms* (Artech House, 1992) and *Digital Signal Compression.* Dr. Malvar is a member of the Sigma Xi. He was the recipient of the Young Scientist award from the Marconi International Fellowship in 1981, and the Senior Award in Image Processing from the IEEE Signal Processing Society in 1992.