Prioritized DCT for Compression and Progressive Transmission of Images

Yunming Huang, *Member, IEEE,* Howard M. Dreizen, *Member, IEEE,* and
Nikolas P. Galatsanos, *Member, IEEE*

***Abstract*—In this paper a new approach based on the block discrete cosine transform (DCT) for compression and progressive transmission of images is presented. The novelty of this approach is that the transform coefficients of all image blocks are coded and transmitted in absolute magnitude order. The resulting ordered-by-magnitude transmission is accomplished without sacrificing coding efficiency by using partition priority coding (PPC) a new source coding method that allows the transmission of an ordered data source without coding overhead due to prioritization. Using this approach, coding and transmission are adaptive to the characteristics of each individual image, and therefore, very efficient. Another advantage of this approach is its high progression effectiveness. Since the largest transform coefficients that capture the most important characteristics of images are coded and transmitted first, this method is well suited for progressive image transmission (PIT). Further compression of the image data is achieved by utilizing multiple distribution entropy coding (MDEC). MDEC is a new coding technique based on arithmetic coding. Experiments are presented where the new DCT approach was tested. It is shown that it compares favorably with previously reported DCT and subband image codecs and is also very effective for PIT.**

I. Introduction

Transform coding denotes a procedure in which the image is subjected prior to transmission to an invertible transform with aim to convert the statistically dependent image elements to independent coefficients. Thus through this transform the data are compressed by redundancy reduction. However, in many cases, in order to achieve further compression, the transformed data are also subjected to *irrelevancy reduction* [1]. Data that are considered irrelevant and do not convey a lot of information about the image are discarded. Thus the redundancy reduction is reversible, whereas the irrelevancy reduction is not.

Transform coding using the 2-D block discrete cosine transform (DCT) is a proven method for image compression and widely used by both the academic and industrial image processing communities [2]–[15]. For this approach the image is divided into blocks and the 2-D DCT transform is applied separately to each block. Irrelevancy reduction is then applied to the resulting transform coefficients of each block such that the most relevant information is retained for transmission or storage while the rest is eliminated. A considerable amount of research has been devoted to adapting the irrelevancy reduction to the nonstationary nature of real-world images. Because of this nonstationarity, the pattern of most relevant information varies from block-to-block and from image-to-image.

A number of solutions have been suggested to the irrelevancy reduction problem. Chen and Smith in [3] proposed the partitioning of the blocks into classes based on the activity of the ac components. For each class a bit allocation map is assigned, which captures the characteristics of the class; more bits are assigned to larger coefficients and for classes that have higher ac activity. Another method suggested by Ngan [12] and Chen [4], adopted for many applications [12], uses different types of scans of the normalized DCT coefficients; priority of the coefficient in each block is decided by its order in the scan of the block (see, for example, [12, fig. 4] for the zig-zag scan). A philosophically similar solution to this problem was suggested in [7] and [10]; the transform coefficients in each block are arranged in vectors and a clustering algorithm is used to encode them using vector quantization methods.

All the above irrelevancy reduction methods yield, in general, satisfactory results. However, they all suffer from the same problem, their adaptivity is limited by the number of classes, bit allocation maps, or vectors allowed. Although more bits are assigned to groups of "busy" blocks and fewer bits to the groups of "quiet" blocks, they cannot exactly match each individual block's characteristics. These methods are optimally adapted to a class of images in a statistical sense, but not to each individual image that is coded. Inevitably, some of the low frequency coefficients are assigned extra bits, or some high frequency coefficients are assigned fewer bits than required. This leads to redundancy or loss of information, respectively. Another problem with these methods is the loss of high frequency coefficients. The coefficients of the DCT block (i.e., as organized with dc in the upper left corner) in the lower right corner are typically discarded to meet the bit quota. This results in low quality reconstructed images, especially in the case of images with

Manuscript received May 15, 1991; revised February 25, 1992.

Y. Huang was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL. He is now with Audio DigitalImaging, Inc., Arlington Heights, IL 60005.

H. M. Dreizen was with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL. He is now with EPIX Inc., Northbrook, IL 60062.

N. P. Galatsanos is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, 60616.

IEEE Log Number 9202156.

large high frequency content; for example, images with text overlay. To avoid some of these problems variable length Huffman coding and uniform quantization of the DCT coefficients have been proposed in [2], [4].

Progressive image transmission (PIT) is another application where the block DCT has been used extensively, see [14, p. 312] and references within. PIT involves an approximate reconstruction of the image whose fidelity is built up gradually until either the viewer decides to abort the transmission or perfect reconstruction is achieved [5]. The applications of PIT are in transmitting images over low bandwidth channels such as telephone lines. Examples of such applications are teleconferencing, electronic shopping in mail order companies, security systems, and access to remote image databases.

For these applications it is important that the image develops detail rapidly over the entire image so the viewer can recognize the image as early as possible in the transmission. Therefore, information that describes important features of the image must be transmitted first. As mentioned previously, because of the nonstationarity of real-world images, the pattern of most significant information varies from image-to-image. Therefore, coding and transmitting information about the transform coefficients based on relevance is also a very important problem for PIT applications.

In this paper, a new block DCT approach called priority DCT coding (PDCT) is introduced. For this approach the transform coefficients are simply rounded to C bit precision. Then, the coefficients of the *entire* image are sorted, the largest ones combined with their position indicators are encoded and transmitted first, followed by the smaller ones, till either the desired bit rate or image quality is achieved. Determining the relevance of transform coefficients based on their magnitude has many advantages. It is mathematically justified because the larger coefficients of an orthonormal transform carry more of the signal energy. Furthermore, for a given bit rate, this selection of coefficients adapts to the characteristics of each image that is compressed and is also very suitable for PIT because larger coefficients, with a proportionally larger impact on the reconstructed image, are sent first. However, transmitting an ordered data source typically yields a higher coding cost because the position information must also be sent. In this paper we introduce partition priority coding (PPC) a new method for transmitting a data source in priority order without any additional coding cost due to the prioritization of the data.

After taking the block DCT of the image, the selected by PPC coefficients for transmission are arithmetically coded [16], [18]. In this paper a new technique called multiple distribution entropy coding (MDEC) is introduced. It involves partitioning the transmitted data into subsources and then using the multiple probability distributions of the subsources to entropy encode them. It is shown that this partitioning yields further compression of the image data.

The rest of this paper is organized as follows: in Section II PPC is introduced and its theoretical properties are examined. It is shown that the entropy of the original source is equal to the average entropy of the ordered source plus the average entropy of the position information necessary to reconstruct the source. Section III contains the implementation details of the PDCT coder. In Section IV the MDEC technique is explained and its properties are examined. Section V contains experimental results. Finally, Section VI contains a discussion, conclusions, and plans for future research.

II. Partition Priority Coding (PPC)

The advantages of irrelevancy reduction based on magnitude are well known [4]. However, prioritizing a source by magnitude typically leads to a decrease of the coding efficiency of the transmission since the coding order is unknown to the receiver, and therefore, must be transmitted. It has generally been assumed that informing the receiver of the coding order imposed on a particular sequence of samples (or coefficients) requires significant coding overhead, against which the application's benefit from the prioritized coding must be weighed [5]. In the rest of this section, a prioritized coding method called partition priority coding (PPC), is presented. The advantage of this method is that its average coding cost of transmitting the data (ordered source + position index) is identical to that of the data in index order.

Transmission of a sequence of samples is commonly performed in the natural order defined by the underlying time or spatial index. Only the sample values need be coded as the index of each received sample is known *a priori*. Prioritized transmission of the samples, according to value, would be a more desirable method to transmit a source. In this way, the most "relevant" samples with larger values are coded and sent first followed by less "relevant" ones. In order to reduce the position order coding, overhead partitioning of the data is performed first. In what follows PPC, a new data transmission method is presented. The interesting result is that the total coding cost of PPC is identical to non-PPC coding without prioritization.

Let $R = \{S_1, \cdots, S_t, \cdots, S_M\}$ be a partition on the range of the data X_i, where $1 \leq X_i \leq R$. That is, S_t are disjoint, nonempty, and $\cup S_t = R$. Each S_t represents a coding pass of PPC; for each S_t, position codes $\phi_t(I_i)$ indicating which X_i are within pass S_t, together with $\theta_t(X_i)$, the value of all samples $X_i \in S_t$, are transmitted. For example consider the source X_i

$$X_i = 10\ 40\ 30\ 20\ 50\ 40\ 20\ 45\ 10\ 55$$

using the partition

$$R = \{S_1 = \{48, \cdots, 55\},\quad S_2 = \{32, \cdots, 47\},$$
$$S_3 = \{16, \cdots, 31\},\quad S_4 = \{0, \cdots, 15\}\}$$

this source would be coded as (left to right, top to bottom):

$\phi_1(I_i)$	0	0	0	0	1	0	0	0	0	1
$\theta_1(X_i)$	—	—	—	—	50	—	—	—	—	55
$\phi_2(I_i)$	0	1	0	0	—	1	0	1	0	—
$\theta_2(X_i)$	—	40	—	—	—	40	—	45	—	—
$\phi_3(I_i)$	0	—	1	1	—	—	1	—	0	—
$\theta_3(X_i)$	—	—	30	20	—	—	20	—	—	—
$\phi_4(I_i)$	—	—	—	—	—	—	—	—	—	—
$\theta_4(X_i)$	10	—	—	—	—	—	—	—	10	—

where "—" represents no symbol coded. In practice, the $\phi_t(I_i)$ and $\theta_t(X_i)$ of pass S_t can be interleaved (for S_1 above, coding 0, 0, 0, 0, 1, 50, 0, $\cdots$) without affecting the coding cost. Sample X_i coded in pass S_t will obviously never be coded again, thus $\phi_u(I_i)$ = "—," for $u > t$. The position of the last pass $\phi_M(I_i)$ can be inferred; all X_i not coded in previous passes are coded in the last pass.

Selection of the passes controls the degree of priority coding as within each pass the eligible samples are coded without priority. If $M = 1$, PPC reduces to nonpriority coding. For $M = R$, PPC provides complete priority coding, all $X_i = r$ are transmitted before any $X_i < r$.

Theorem: $H(X_i) = \overline{H}_\theta(X_i) + \overline{H}_\gamma(I_i)$, where $H(X_i)$ is the entropy of the original unordered data, and $\overline{H}_\theta(X_i)$, $\overline{H}_\gamma(I_i)$, the average entropy of the ordered data and the position indicators for all the PPC passes respectively.

For a proof of this theorem see Appendix A. This theorem shows that the average entropy of position indicators and samples values using the PPC method is identical to that of ordinary coding sample values in index order. The most interesting case of PPC is $M = R$ in which each S_t contains a single value of R; $\overline{H}_\theta(X_i) = 0$, and therefore, $\overline{H}_\phi(I_i) = H(X_i)$. Thus coding position indicators for each value in R provides complete priority coding with entropy identical to $H(X_i)$.

A similar priority coding method, based on Huffman and runlength coding, was successfully used for nonhomogeneous PIT in [5]. However, in contrast with PPC prioritization in [5], it generates coding overhead.

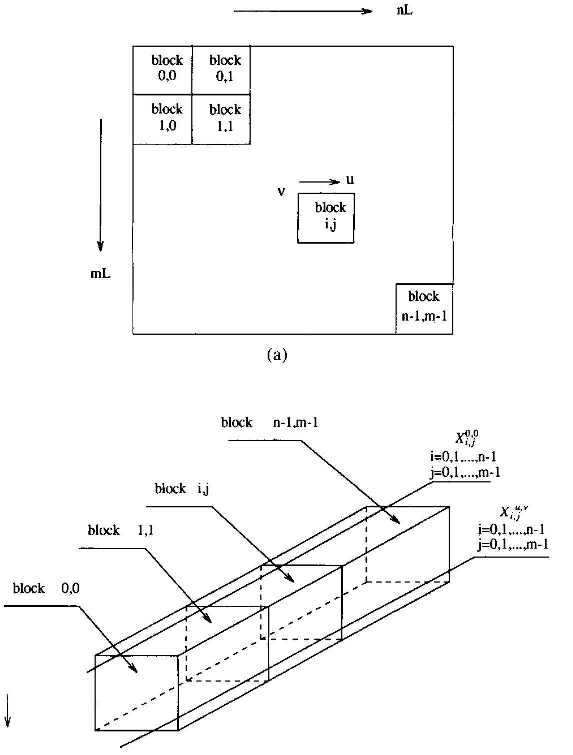
III. Description of the Implementation of the PDCT Codec

For simplicity of discussion, the images are assumed to have $n \cdot L \times m \cdot L$ pixels, where L is the length of the square blocks and n, m are the number of blocks in horizontal and vertical directions, respectively. An image can then be represented by the 4-tuple $Y(k, l, i, j)$:

$$Y(k, l, i, j) = \{Y_{i,j}^{k,l},\ k, l = 0, 1, \cdots, L-1\},$$
$$i = 0, 1, \cdots, n-1,$$
$$j = 0, 1, \cdots, m-1$$

where k, l are the indexes of the pixels within the block and i, j are the indexes of the block as shown in Fig. 1(a). The blocks, $Y_{i,j}$ then undergo an orthogonal and unitary block DCT transform $A(A^t = A^{-1}$, t is the transpose):

$$Q_{i,j} = AY_{i,j}A^t \tag{1}$$



(a)

(b)

Fig. 1. (a) The image blocks in the DCT domain. (b) The resulting data sources for $Z = 1$.

where $Y_{i,j}$ and $Q_{i,j}$ are $L \times L$ blocks of pixel data and DCT coefficients, respectively, and A is the $L \times L$ DCT coefficient matrix [14]. The resulting coefficients, $Q_{i,j}^{u,v}$, are rounded to C bit precision before arithmetic coding and transmission. The quantized coefficients are denoted by $X_{i,j}^{u,v}$.

After the DCT transform, the original image $Y(k, l, i, j)$ is converted into the transformed image $X(u, v, i, j)$ with coefficient elements, $X_{i,j}^{u,v}$. At the receiver, the $X_{i,j}^{u,v}$ coefficients are used to reconstruct an approximation of the coded image, $\overline{Y}$. By using the notation in (1) the reconstructed image is given by

$$\overline{Y}_{i,j} = A^t X_{i,j} A, \qquad i = 0, 1, \cdots, n-1,$$
$$j = 0, 1, \cdots, m-1. \tag{2}$$

The DCT coefficients $X_{i,j}^{u,v}$ $(-2^{c-1} \leq X_{i,j}^{u,v} < +2^{c-1})$ are coded using PPC. For each pass $S_t = [S_t^{\min}, S_t^{\max}]$ of a given partition $R = \{S_1, \cdots, S_t, \cdots, S_M\}$, where $S_t^{\min}$ is the low bound of the pass and $S_t^{\max}$ is the high bound of the pass, the entire DCT transformed image $X(i, j, u, v)$ is sequentially scanned from the upper left corner to the lower right corner. As described in Section III, coefficients with values in the range of the present pass and their position indicators are coded. The values of the position indicators $\phi_t(I_{i,j}^{u,v})$ in pass S_t are computed by

$$\phi_t(I_{i,j}^{u,v}) = \begin{cases} 1, & S_t^{\max} \geq |X_{i,j}^{u,v}| \geq S_t^{\min} \\ 0, & |X_{i,j}^{u,v}| < S_t, \qquad \text{for all } i, j, u, v. \\ \text{``}-\text{''}, & \text{otherwise} \end{cases} \tag{3}$$

The coefficients of this pass are given by

$$\theta_t(X_{i,j}^{u,v}) = X_{i,j}^{u,v}, \qquad \text{when } \phi_t(I_{i,j}^{u,v}) = 1. \tag{4}$$

The indicators and the coefficients of the pass are coded alternatively, if $\phi_t(I_{i,j}^{u,v}) = 1$, then $\theta_t(X_{i,j}^{u,v})$ follows.

IV. Multiple Distribution Entropy Coding (MDEC)

In this section a new technique for efficient lossless data compression using arithmetic coding is presented. Arithmetic coding is an entorpy source coding method which recently has become very popular. In principle, when a message is coded using this method, the number of bits in the coded string is the same as the entropy of that message with respect to the model (probability distribution of data) used for coding [18]. This method is superior to Huffman coding because it dispenses with the restriction that each symbol must translate to an integer number of bits, thus coding more efficiently.

The transform coefficients $X_{i,j}^{u,v}$ with $i = 1, 2, \cdots n - 1$ and $j = 1, 2, \cdots m - 1$ and fixed u, v are considered as one data source for entropy coding purposes. Thus for each image $L \times L$ data sources are available as shown in Fig. 1(b).

The new technique is called multiple distribution entropy coding (MDEC) and it involves partitioning each of the $L \times L$ data source into subsources. The partitioning for MDEC is different from the partitioning for PPC. Therefore, in order to avoid confusion, the word "splitting" will be used to indicate the partitioning of a source into subsources for MDEC purposes. Thus MDEC involves splitting the $L \times L$ sources with the DCT coefficients into subsources. The multiple probability distributions of the subsources are then utilized to arithmetically code the data. If multiple distributions are used, the available *a priori* knowledge about the original source is higher than that before the partition. Thus the average entropy of the data is now smaller, (see Appendix B for a proof), and this results in a lower coding cost. From the proof in Appendix B it is clear that no coding benefit is produced if the subsources have identical distributions.

The idea of splitting a source into subsources in coding applications is not new. In [3], Chen and Smith used ac energy classification as the criterion to split the DCT coefficients into subsources and then designed quantizers for each subsource. Similarly, Wornell and Staelin used a maximum-likelihood approach for this splitting [20]. From the findings in [3] and [17], it is clear that if ac energy classification is used, the DCT coefficients of different classes have different variances and thus different distributions. Therefore, based on the proof in Appendix B, theoretically, at least, MDEC will always yield a coding gain.

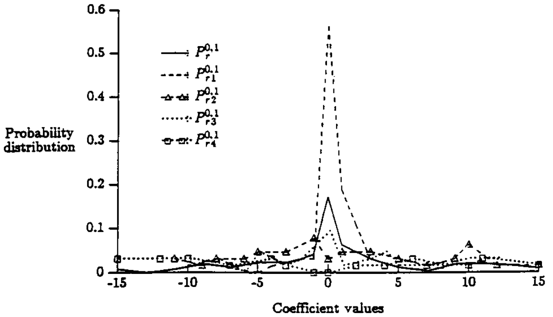


Fig. 2. The probability distributions functions (PDF's) of the (0, 1) DCT coefficient. $P_r^{0,1}$ is the PDF of the entire source. $P_{r1}^{0,1}$, $P_{r2}^{0,1}$, $P_{r3}^{0,1}$, and $P_{r4}^{0,1}$, are the PDF's of the $Z = 4$ subsources. Index splitting and the image "Karen" are used. The measured average entropy of the DCT coefficients using $Z = 1$ for each frequency is 1.060 b. The measured average entropy of the DCT coefficients using MDEC with $Z = 4$ subsources for each frequency is 0.970 b.

In practice, splitting based on ac energy for arithmetic coding purposes has the disadvantage that it requires the transmission of the classification map to the receiver. Let the DCT coefficients in each class be represented as $[X_{i,j}^{u,v}]_z$ for $z = 1, 2, \cdots Z$, with Z the number of subsources. In order to decode the transmitted data the receiver must know the subsource from which each piece of data came. Thus the classification map of all the blocks must be transmitted as overhead. This overhead is

$$B_{oh} = n \cdot m \log_2 Z \text{ b}. \tag{5}$$

Although in theory more subsources implies larger coding gains, for the ac energy splitting, in practice, this coding gain must be weighed against the increase in overhead required to transmit the block classification map.

In order to overcome the overhead problem of ac energy splitting an alternative method was used. This method is called index splitting and is based on the location of the block in the image. For example if $Z = 4$ subsources are used, the image is split into equal quadrants and the blocks of each quadrant form the subsource. Thus for this method the classification map is implicit and does not need to be transmitted. In Fig. 2 we show the probability distribution $P_r^{0,1}$ of the (0, 1) DCT coefficient before splitting along with the probability distributions $P_{r1}^{0,1}$, $P_{r2}^{0,1}$, $P_{r3}^{0,1}$, and $P_{r4}^{0,1}$ of the 4 subsources resulting from index splitting into $Z = 4$ subsources for the "Karen" image. Notice that all 4 distributions are different, thus it is expectd that this method will also yield coding gains. As an example the measured average entropy of the DCT coefficients of the image "Karen" (see Fig. 7(a)) was computed. Using $Z = 1$ (i.e., no MDEC) the entropy was 1.060 b. Using MDEC with $Z = 4$ subsources, it was 0.970 b.

The previously described PDCT codec combines the advantages of both PPC and MDEC and can be used both

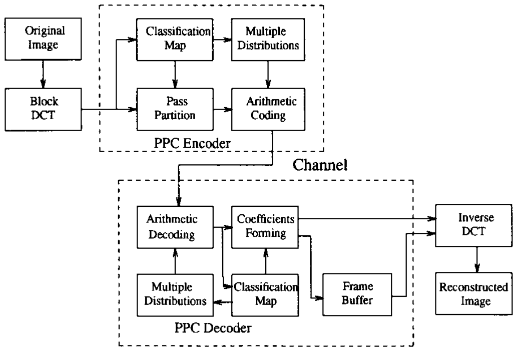


Fig. 3. Block diagram of the PDCT codec.

for image conpression and for PIT. A block diagram for this codec is shown in Fig. 3.

V. Experiments

Experiments were used to test the PDCT image coder. The original images were 8 b/pixel gray scale images, and 16 by 16 or 8 by 8 block DCT was used. Each DCT coefficient was rounded to 8-b precision, that is, the ac coefficients are integers in the range $[-128, 127]$. The dc coefficients are integers in the range $[0, 255]$. The decision to round-off using 8 b is arbitrary and is based on hardware considerations. Using 8-b precision does not yield errorless reconstruction. For this, about 12-b precision would be needed. The coefficients are then coded using PPC and MDEC with arithmetic coding [16] and [18]. For a single pass with the range $\{|128|, 0\}$, PPC degenerates to nonprogressive transmission coding and the output bit rate cannot be controlled, thus multiple passes were used.

The multiple probability distributions of the coefficients and the position indicators were estimated by averaging the corresponding distributions obtained from an image database. For the first (Model 1), 5 images were used with entropies close in value to the entropy of the coded image. For the second (Model 2), 32 arbitrary images were used.

The pass parameters for PPC were chosen to be

$$\{32, 16, 8, 4, 3, 2, 1\}.$$

for $M = 7$ passes. This choice of parameters provides a fairly even distribution of information in each pass. The dc coefficients were coded first followed by the classification map and the ac coefficients pass by pass. Using this approach, even in the very early stages of the transmission (for example, 0.05 b/pixel, or 160:1 compression ratio) a rough and meaningful image approximation can be reconstructed.

To evaluate system performance, the peak mean square error (PMSE) and the peak signal-to-noise ratio (PSNR) are used. PMSE is defined as the MSE normalized by the full range of the signal, i.e.:

$$\mathrm{PMSE} = \frac{\mathrm{MSE}}{255^2} \tag{6}$$

TABLE I
Compression Performance in Progressive Transmission of "Lena" (Picture Size: 512 by 480, Block Size: 16 by 16, Probability Distribution Model: Model 2, MDEC Splitting: Based on AC Energy)

			Bit Rate (bit/pixel)				
Stage	Pass	PSNR (dB)	$Z = 1$	$Z = 4$	$Z = 8$	$Z = 16$	$Z = 32$
0	DC	19.932	0.031	0.039	0.043	0.047	0.051
1	≥32	22.487	0.044	0.050	0.052	0.055	0.058
2	≥16	25.286	0.083	0.082	0.082	0.082	0.080
3	≥8	28.320	0.171	0.160	0.157	0.150	0.137
4	≥4	31.389	0.341	0.318	0.311	0.296	0.269
5	≥3	32.623	0.444	0.416	0.408	0.391	0.358
6	≥2	34.190	0.626	0.587	0.582	0.565	0.528
7	≥1	36.751	1.133	1.076	1.069	1.061	1.015

TABLE II
PSNR Performance in Progressive Transmission of "Lena" (Picture Size: 512 by 480, Block Size: 16 by 16, Probability Distribution Model: Model 2, MDEC Splitting: Based on AC Energy)

		PSNR (dB)				
Stage	Bit Rate (bit/pixel)	$Z = 1$	$Z = 4$	$Z = 8$	$Z = 16$	$Z = 32$
0	0.05	22.786	22.476	21.914	20.752	19.932
1	0.10	25.659	26.017	26.014	26.067	26.253
2	0.15	26.650	27.265	27.303	27.516	28.075
3	0.20	28.661	29.149	29.218	29.359	29.703
4	0.25	29.401	30.118	30.177	30.392	30.925
5	0.30	30.355	31.100	31.221	31.482	31.841
6	0.40	32.029	32.482	32.553	32.775	33.047
7	0.50	33.012	33.458	33.519	33.687	33.985
8	0.75	34.655	35.082	35.173	35.173	35.369
9	final*	36.751	36.751	36.751	36.758	36.751

*: final bit rates are shown in Table I (Stage = 7).

where

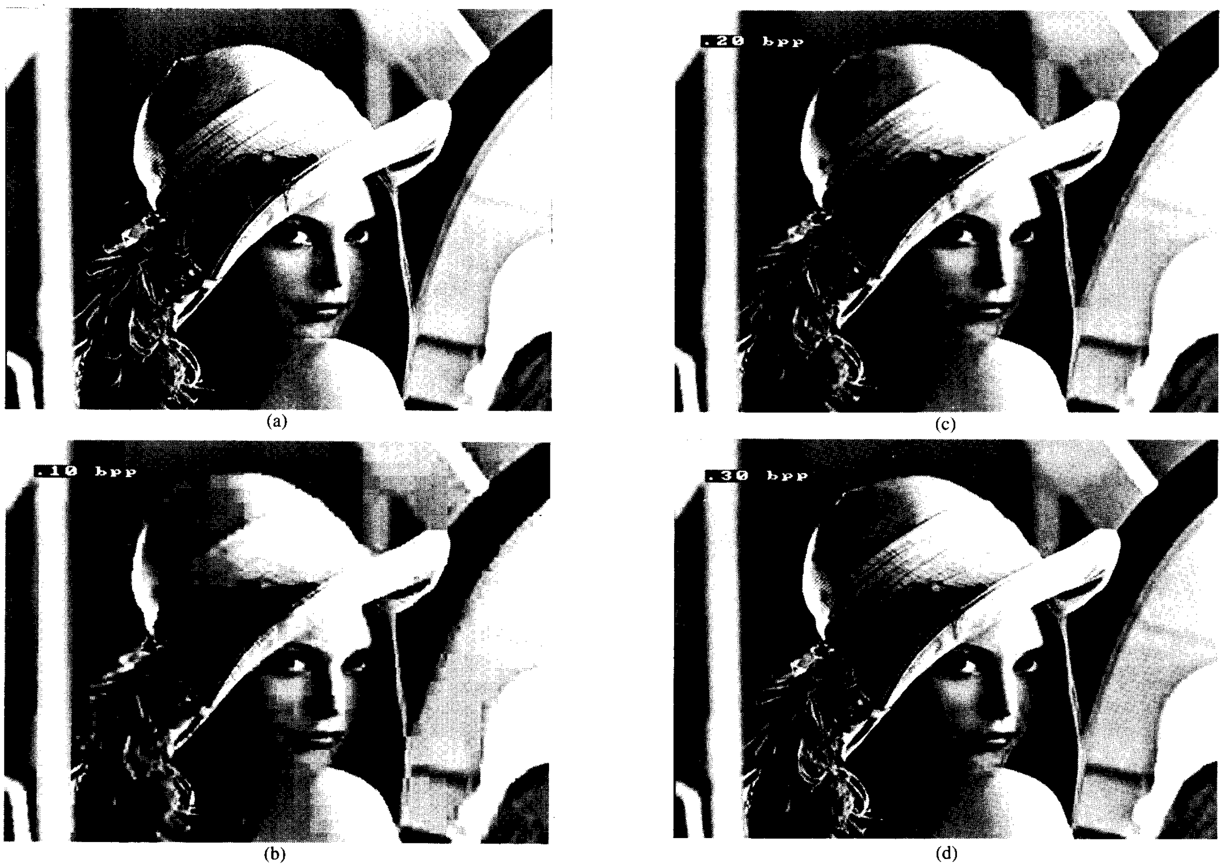
$$\mathrm{MSE} = \frac{1}{n \cdot L \cdot m \cdot L} \sum_{i=0}^{m} \sum_{j=0}^{n} \sum_{u=0}^{L} \sum_{v=0}^{L} [Y_{i,j}^{u,v} - \hat{Y}_{i,j}^{u,v}]^2 \tag{7}$$

$\hat{Y}_{i,j}^{u,v}$ are the reconstructed image pixel values. The PSNR is defined as

$$\mathrm{PSNR} = 10 \log \left(\frac{255^2}{\mathrm{PMSE}} \right). \tag{8}$$

A criterion based on the SNR might have been preferable. However, for purposes of direct comparison with previous coders, PSNR was considered.

In a first experiment, "Lena" with a size of (512 by 480) pixels and 8 b/pixel was used. The image was divided into 16 by 16 blocks and the blocks split into Z classes based on ac energy [3]. Experimental results for different number of Z classes are shown in Tables I and II. In Table I the coding performance of each pass is shown. The PSNR improvement is about 1-3 dB for each pass. When all the image data are transmitted, the final bit rate is 1.015 b/pixel (Model 2) and the PSNR = 36.751 dB. Table II shows PSNR results at different bit



(a) (c)

(b) (d)

Fig. 4. Progressively reconstructed images "Lena" (512 by 480). (a) Original image. (b) 0.10 b/pixel. (c) 0.20 b/pixel. (d) 0.30 b/pixel. (e) 0.40 b/pixel. (f) 0.50 b/pixel. (g) 0.75 b/pixel. (h) 1.00 b/pixel.

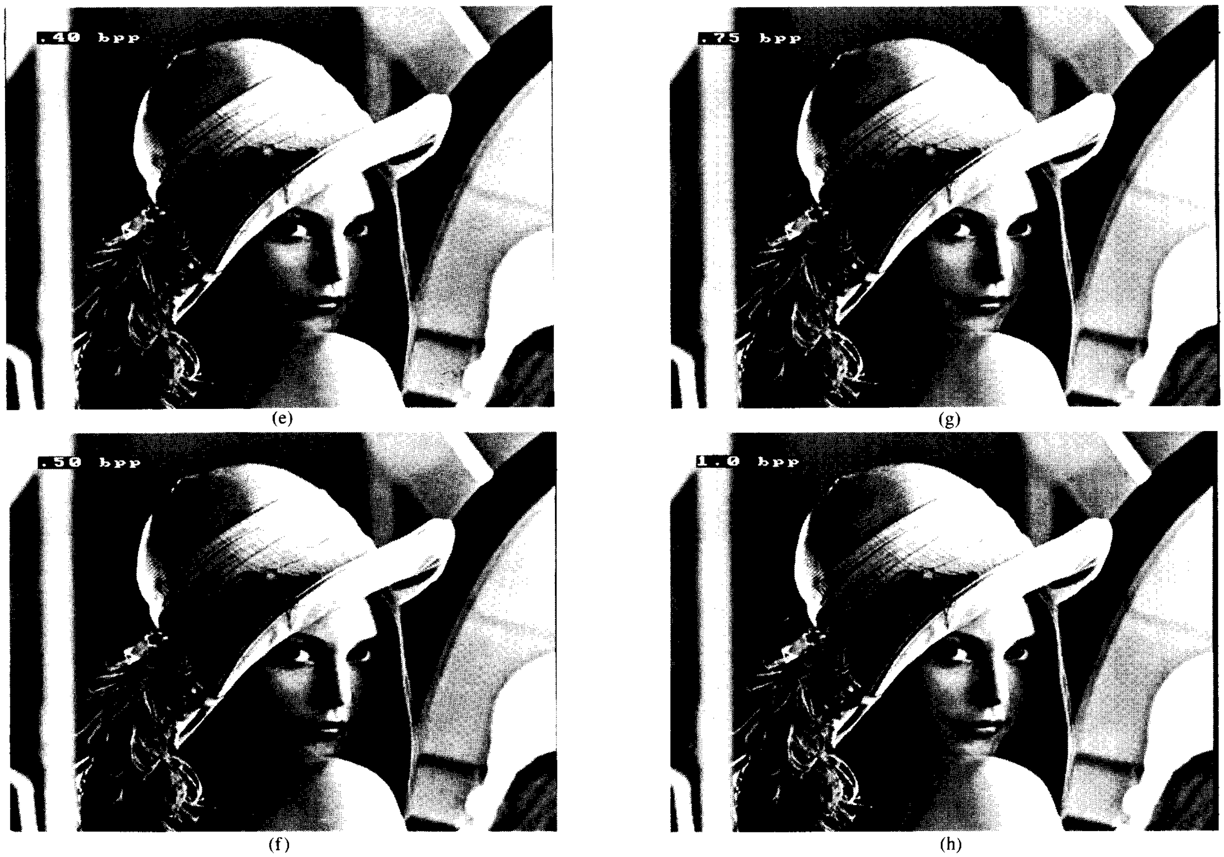
rates. From this table, it is clear that for very low bit rates (0.05 b/pixel) no coding advantage is gained by splitting the data into subsources based on ac energy because of the additional overhead required for the transmission of the classification map.

Tables I and II contain the results of an evaluation of the same coder using two different ways. In Table I the value of the smallest DCT coefficient transmitted is used to define the data utilized to reconstruct the image. Therefore, the resulting PSNR is fixed. However, the resulting bit rate after arithmetic coding will vary depending on the number of subsources used. In Table II the final bit rate resulting after arithmetic coding is fixed. Therefore, for different numbers of subsources the compression ratio will vary, and thus the PSNR will also change.

Fig. 4(b) to (i) shows a set of reconstructed images, using Model 2 with $Z = 32$, at different bit rates. From these figures it is clear that even at the very early stages of transmission, 0.10 b/pixel (compression ratio = 80:1), the reconstructed image exhibits the same basic features as the original to a degree sufficient to recognize it. During the following stages more details are added to the image. After reception of about 0.50 b/pixel, the reconstructed image is of a good quality (SNR $\geq$ 34.0 dB). This progressive transmission experiment shows the effectiveness of the PDCT coder for this application. The curves in Fig. 5 show the PSNR performance of this transmission. These curves show that the PSNR slope is very high during early transmission (bits/pixel < 0.25). By comparing the two curves in Fig. 5, it is also clear that statistical errors in probability distribution models did not seriously degrade coding performance.

For comparison purposes with previously reported codecs, a second experiment was performed using the 256 by 256 "Lena" image with 8 by 8 blocks. For this experiment both ac energy and index splitting were used to divide in Z subsources. The PSNR results for different bit rates with (Z = 1, 4, 8, 16, 32) for the ac energy and index splitting methods are shown in Tables III and IV, respectively. From these tables it is clear that at low bit rates (especially for large Z), the index is superior to ac energy splitting. However, for higher bit rates, ac energy splitting yields better results. The results in Table IV are compared with results reported in [19]. This comparison (Fig. 6) shows that the PDCT codec outperformed both previous DCT based codec by 3–4.5 dB and the adaptive subband codec reported in [19] by 1–3.5 dB.

In a third experiment, the 256 by 256 "Karen" image with text overlay was used. The block size was chosen to be 16 by 16. This experiment was designed to show the



(e)

(g)

(f)

(h)

Fig. 4. (*Continued.*)

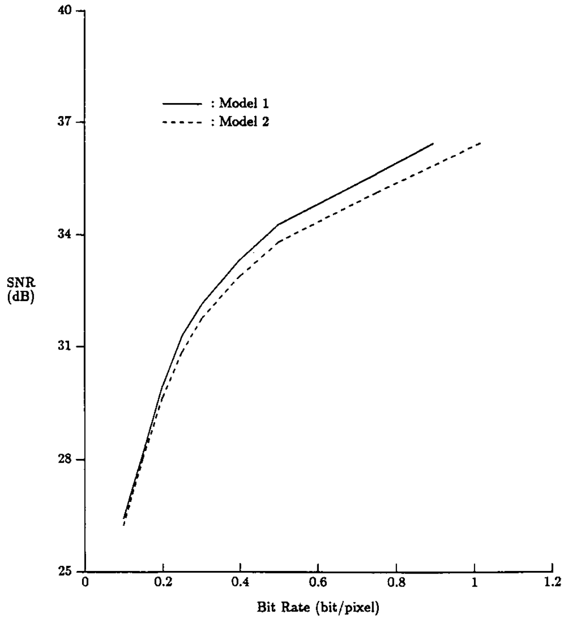


Fig. 5. Progressive image transmission performance for "Lena" (512 by 480) using Model 1 and Model 2.

TABLE III

PSNR Performance in Progressive Transmission of "Lena" (Picture Size: 256 by 256, Block Size: 8 by 8, Probability Distribution Model: Model 2, MDEC Splitting: Based on AC Energy)

Stage	Bit Rate (bit/pixel)	PSNR (dB)				
		$Z = 1$	$Z = 4$	$Z = 8$	$Z = 16$	$Z = 32$
0	0.20	23.632	23.212	23.634	21.071	18.749
1	0.30	25.922	25.940	25.827	25.839	25.905
2	0.40	27.073	27.231	27.235	27.306	27.623
3	0.50	28.268	28.584	28.618	28.854	29.763
4	0.75	30.838	31.317	31.354	31.481	31.992
5	1.00	33.143	33.950	34.041	34.283	34.762
6	1.25	35.297	35.966	36.021	36.206	36.582
7	1.50	36.773	37.517	37.621	37.761	38.228
8	1.75	38.341	38.917	39.962	39.044	39.295
9	2.00	39.284	39.927	40.024	40.127	40.470

advantages of our method for integrated text, gray scale image transmissions. The reconstructed image using the method in [3] at 1.0 b/pixel is shown in Fig. 7(b). This image is blurred and the text is not recognizable. The reconstructed images using the PDCT coder, at 0.50 and 1.0 b/pixel are shown in Fig. 7(c) and (d), respectively. Using the PDCT coder the text is recognizable even at a bit rate of 0.50 b/pixel.

TABLE IV
PSNR Performance in Progressive Transmission of "Lena" (Picture Size: 256 by 256, Block Size: 8 by 8, Probability Distribution Model: Model 2, MDEC Splitting: Based on Index)

		PSNR (dB)				
Stage	Bit Rate (bit/pixel)	$Z = 1$	$Z = 4$	$Z = 8$	$Z = 16$	$Z = 32$
0	0.20	23.632	23.806	23.977	24.261	24.594
1	0.30	25.922	26.378	26.568	26.899	27.243
2	0.40	27.073	27.331	27.641	28.175	29.099
3	0.50	28.268	28.621	29.131	29.984	30.621
4	0.75	30.838	31.144	31.384	31.837	32.725
5	1.00	33.143	33.557	33.854	34.508	35.101
6	1.25	35.297	35.508	35.780	36.234	36.757
7	1.50	36.773	37.034	37.283	37.660	38.334
8	1.75	38.341	38.623	38.730	38.956	39.327
9	2.00	39.284	39.487	39.661	39.949	40.479

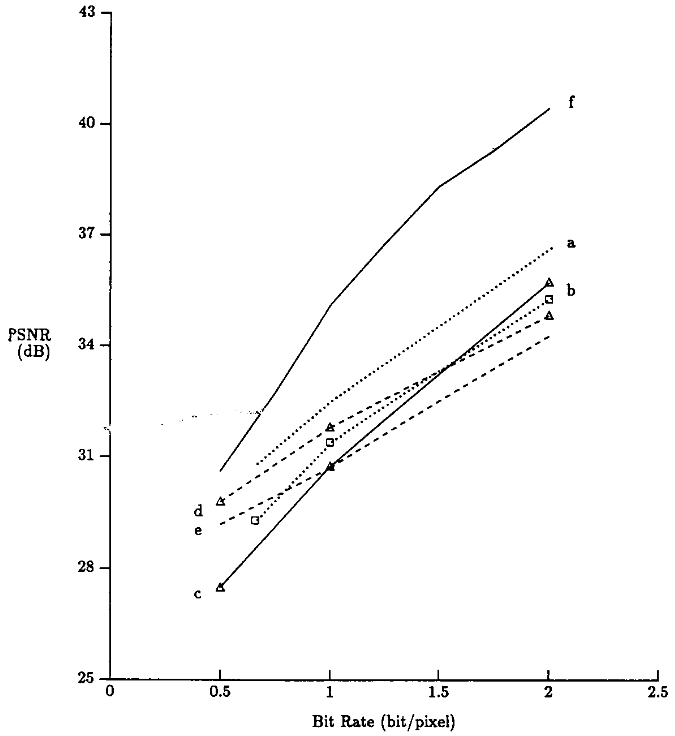


Fig. 6. PSNR versus bit rate for "Lena" (256 by 256) using: (a) adaptive SBC, (b) nonadaptive SBC, (c) adaptive DCT, (d) differential VQ, (e) VQ, and (f) PDCT (plots (a)-(e) are taken from [19]).

VI. Discussion, Conclusions, and Future Research

In this paper a new block DCT based coder was presented. The novelties of this coder are the PPC and MDEC ideas. PPC is a new source coding and transmission method that allows transmission of the transform coefficients based on magnitude without a prohibitive amount of overhead due to prioritization. At this point it must be made clear that the entropy equivalence of the unordered data and the prioritized data using PPC holds only if the *entire* source is transmitted. This can be the case in PIT applications. However, for image compression applications this may not always be true. The rate distortion properties of partial PPC is still an open question that is currently under investigation. The computational overhead that PPC adds is equivalent to the computational load required to order a source by magnitude. This load increases as the number of passes increase.

MDEC is a lossless arithmetic coding technique based on the principal that the average entropy of a source can be reduced if it is split into subsources with different distributions. Two methods for splitting into subsources were tested: ac energy and index block classification. Experimental evidence in Tables III and IV indicates that at low bit rates the overhead of the classification map required by the ac energy method is significant, thus the index method yields superior results. However, at high bit rates

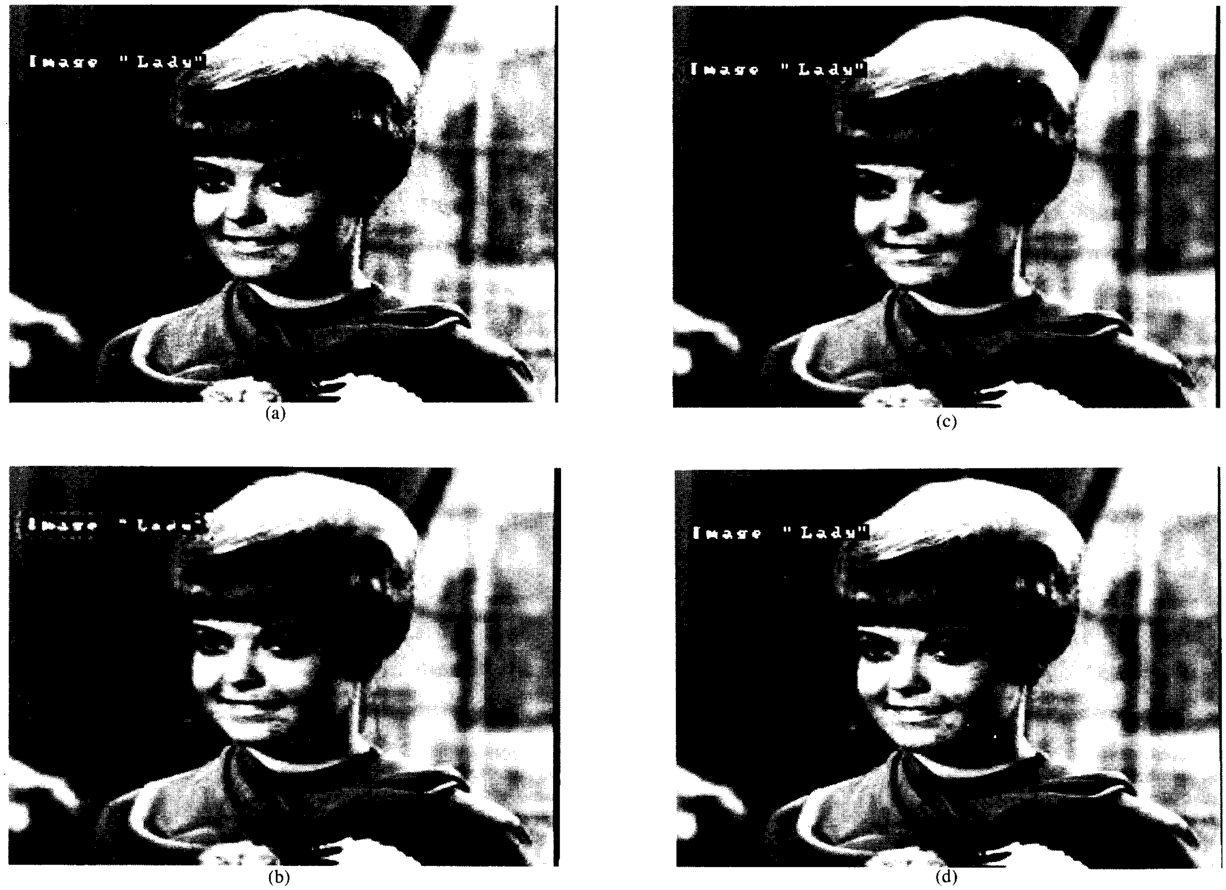


Fig. 7. Reconstructed images of "Karen with Text" using (b) adaptive DCT method [3], (c), and (d) prioritized DCT method. (a) Original image. (b) 1.00 b/pixel (ADCT). (c) 0.50 b/pixel (PDCT). (d) 1.00 b/pixel (PDCT).

the ac energy method is preferable because it yields better (from an entropy coding point of view) subsources. The curves in Fig. 6 indicate that the advantage over previous codecs of the PPC–MDEC combination becomes bigger at higher bit rates. When PPC is combined with MDEC, the bigger fraction of the data transmitted, the bigger the compression ratio due to MDEC. This behavior is explained by the fact that the transmitted data then matches the available statistical description better, which was obtained by using the entire data source.

Currently, research is under way where PPC is used for coding image sequences [6] and [8]. Initial experiments show that magnitude-based irrelevancy reduction using PPC is also very effective for this application. The main advantage of PPC in this application is that it adapts optimally to the nonstationary characteristics of the interframe error.

Appendix A

In this appendix it is shown that the entropy of an unordered data source is equal to the entropy of the ordered data source plus the entropy of the position information required to reconstruct the original data source using PPC. Let X_i be a sequence of $N > 1$ independent samples $X_1, X_2, \cdots, X_N$ with time or spatial indexing, with a discrete range of integer values $1 \leq X_i \leq R$, and with known $P\{X_i = r\}$ or P_r. As usual, the entorpy is given by

$$H(X_i) = \sum_{r=1}^{R} - P_r \log_2 P_r. \qquad \text{(A-1)}$$

Let $R = \{S_1, \cdots, S_t, \cdots, S_M\}$ be a partition on the range of X_i. That is, the S_t are disjoint, nonempty, and $\cup S_t = R$.

The entropy of samples in pass S_t is derived from $P\{X_i = r | X_i \in S_t\}$:

$$H_{\theta_t}(X_i) = \sum_{r \in S_t} - \frac{P_r}{P_{S_t}} \log_2 \frac{P_r}{P_{S_t}} \qquad \text{(A-2)}$$

where

$$P_{S_t} = P\{X_i \in S_t\} = \sum_{r \in S_t} P_r. \qquad \text{(A-3)}$$

The average entropy of the samples for all passes is

$$\overline{H}_\theta(X_i) = \sum_{t=1}^{M} P_{S_t} \cdot H_{\theta_t}(X_i). \qquad \text{(A-4)}$$

The entropy of the position indicators is computed using $P\{\phi_t(I_i) = 1\}$ which is P_{S_t}. Since an I_i of 1 in pass S_t

is always a "–" in later passes

$$P\{\phi_t(I_i) = 0\} = 1 - \sum_{u=1}^{t} P_{S_u} = \sum_{u=t+1}^{M} P_{S_u} \qquad \text{(A-5)}$$

and

$$\begin{aligned} P^*\{\phi_t(I_i)\} &= P\{\phi_t(I_i) = 1\} + P\{\phi_t(I_i) = 0\} \\ &= \sum_{u=t}^{M} P_{S_u}. \qquad \text{(A-6)} \end{aligned}$$

Position indicator entropy in pass S_t, therefore, is

$$\begin{aligned} H_{\phi_t}(I_i) = &-\frac{P\{\phi_t(I_i) = 1\}}{P^*\{\phi_t(I_i)\}} \log_2 \frac{P\{\phi_t(I_i) = 1\}}{P^*\{\phi_t(I_i)\}} \\ &- \frac{P\{\phi_t(I_i) = 0\}}{P^*\{\phi_t(I_i)\}} \log_2 \frac{P\{\phi_t(I_i) = 0\}}{P^*\{\phi_t(I_i)\}}. \qquad \text{(A-7)} \end{aligned}$$

The average entropy of the indicators for all passes is

$$\overline{H}_\phi(I_i) = \sum_{t=1}^{M} P^*\{\phi_t(I_i)\} H_{\phi_t}(I_i). \qquad \text{(A-8)}$$

Pass entropy is defined as

$$H_{\theta S} = \sum_{t=1}^{M} - P_{S_t} \log_2 P_{S_t} \qquad \text{(A-9)}$$

Lemma 1: $\overline{H}_\phi(I_i) = H_{\theta S}$, where $\overline{H}_\phi(I_i)$ is the entropy of the position indicators.

Proof: Expanding (A-8):

$$\begin{aligned} \overline{H}_\phi(I_i) = \sum_{t=1}^{M} \Bigg[&-P_{S_t} \log_2 \frac{P_{S_t}}{\sum_{u=t}^{M} P_{S_u}} \\ &- \left(\sum_{u=t+1}^{M} P_{S_u} \right) \log_2 \frac{\sum_{u=t+1}^{M} P_{S_u}}{\sum_{u=t}^{M} P_{S_u}} \Bigg] \qquad \text{(A-10)} \end{aligned}$$

using log $(A/B) = \log A - \log B$ and combining $(P_{S_t}) + (\Sigma_{u=t+1}^{M} P_{S_u})$ as $\Sigma_{u=t}^{M} P_{S_u}$ (A-10) yields

$$\begin{aligned} &= \sum_{t=1}^{M} \left[\left(\sum_{u=t}^{M} P_{S_u} \right) \log_2 \left(\sum_{u=t}^{M} P_{S_u} \right) \right] \\ &- \sum_{t=1}^{M} \left[\left(\sum_{u=t+1}^{M} P_{S_u} \right) \log_2 \left(\sum_{u=t+1}^{M} P_{S_u} \right) \right] \\ &+ \sum_{t=1}^{M} - P_{S_t} \log_2 P_{S_t} \qquad \text{(A-11)} \end{aligned}$$

canceling terms from the first bracket of (A-11) for $t = v + 1$ with terms of the second bracket for $t = v$, and taking into account that the first bracket is zero for $t = 1$ and the second bracket for $t = M$ $(\Sigma_{v=1}^{M} P_{S_v} = 1)$ yields

$$\overline{H}_\phi(I_i) = H_{\theta S}. \qquad \text{(A-12)}$$

Theorem: $\overline{H}_\theta(X_i) + \overline{H}_\phi(I_i) = H(X_i)$.

Proof: From (A-1), (A-2), and (A-4):

$$\begin{aligned} \overline{H}_\theta(X_i) &= \sum_{t=1}^{M} \sum_{r \in S_t} - P_r \log_2 \frac{P_r}{P_{S_t}} \\ &= \left[\sum_{t=1}^{M} \sum_{r \in S_t} - P_r \log_2 P_r \right] \\ &\quad - \sum_{t=1}^{M} \left[-\log_2 P_{S_t} \sum_{r \in S_t} P_r \right] \\ &= \sum_{r=1}^{R} - P_r \log_2 P_r - \sum_{t=1}^{M} - P_{S_t} \log_2 P_{S_t} \\ &= H(X_i) - H_{\theta S}. \qquad \text{(A-13)} \end{aligned}$$

Using Lemma 1 and rearranging the terms in (A-13) completes the proof.

Appendix B

In this Appendix it is shown that the entropy of a source is reduced if it is partitioned into subsources with different probability distributions. Let X_i be a source sequence of $N > 1$ independent samples $X_1, X_2, \cdots, X_N$ with time or spatial indexing, with a discrete range of integer values $1 \le X_i \le R$, and with known $P\{X_i = r\}$ or P_r. As usual, the source entropy $H(X_i) = \Sigma_{r=1}^{R} - P_r \log_2 P_r$.

Multiple distribution entropy coding divides the source X_i to M subsource sequences X_i^k, where $1 \le k \le M$. X_i^k is a sequence of L_k independent samples $X_1^k, X_2^k, \cdots, X_{L_k}^k$ with time or spatial indexing, and with known $P\{X_i^k = r\}$ or P_r^k. That is, the X_i^k are disjoint, nonempty, and $\cup_k \{X_i^k\} = \{X_i\}$. Because X_i^k are subsources of X_i, it is clear that

$$P_r = \sum_{k=1}^{M} \frac{L_k}{N} \cdot P_r^k, \qquad 1 \le r \le R. \qquad \text{(B-1)}$$

After dividing the source, the average entropy of the subsources based on their multiple probability distributions is

$$\begin{aligned} \overline{H}(X_i^k) &= \sum_{k=1}^{M} H(X_i^k) \\ &= \sum_{k=1}^{M} \frac{L_k}{N} \left\{ \sum_{r=1}^{R} - P_r^k \log_2 P_r^k \right\} \qquad \text{(B-2)} \end{aligned}$$

the mean of the entropy of the subsources.

Lemma 2: $\overline{H}(X_i^k) \le H(X_i)$ for known P_r^k.

Proof: Because $\overline{H}(X_i^k)$ is a function of P_r^k, the maximum value of $\overline{H}(X_i^k)$ can be found from

$$\frac{d\overline{H}(X_i^k)}{dP_r^k} = 0, \qquad \text{for } k = 1, 2, \cdots M \qquad \text{(B-3)}$$

with the constraint that

$$P_r^M = \frac{N}{L_M} P_r - \sum_{k=1}^{M-1} \frac{L_k}{L_M} P_r^k, \qquad 1 \le r \le R. \qquad \text{(B-4)}$$

Equations (B-3) and (B-4) yield

$$\frac{dH(X_i^k)}{dP_r^k} = \frac{d[-P_r^k \log_2 P_r^k - P_r^M \log_2 P_r^M]}{dP_r^k}$$

$$= -P_r^k\left(\frac{1}{P_r^k}\right)\log_2 e - \log P_r^k + P_r^M\left(\frac{1}{P_r^M}\right)$$

$$\cdot \log_2 e + \log P_r^M = \log_2 \frac{P_r^k}{P_r^M}. \qquad \text{(B-5)}$$

From (B-5) it is clear that the derivative is zero if $P_r^k = P_r^M$ for $k = 1, 2, \cdots M$. Then (B-1) yields

$$P_r = P_r^k, \qquad 1 \leq k \leq M. \qquad \text{(B-6)}$$

Combining (B-2) and (B-6) yields

$$[\overline{H}(X_i^k)]_{\max} = H(X_i). \qquad \text{(B-7)}$$

Lemma 2 verifies that the entropy of a source is reduced or remains constant if we divide it to $M > 1$ subsources. Also, from the proof it is clear that no coding advantage is gained if the subsources have the same probability distributions.

References

[1] H. Musmann, P. Pirsch, and H. Grallert, "Advances in picture coding," *Proc. IEEE*, pp. 523-542, Apr. 1982.

[2] "CCITT SG XV working party XV/4, Specialist group on coding of visual telephony, Description of reference model 8 (RM8)," Doc. 525, June 1989.

[3] W. H. Chen and C. H. Smith, "Adaptive coding of monochrome and color image," *IEEE Trans. Commun.*, vol. COM-25, pp. 1285-1292, Nov. 1977.

[4] W. H. Chen and W. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225-231, Mar. 1984.

[5] H. M. Dreizen, "Content-driven progressive transmission of gray level images," *IEEE Trans. Commun.*, vol. COM-35, pp. 289-296, Mar. 1987.

[6] S. N. Efstratiadis, Y. Huang, Z. Xiong, N. P. Galatsanos, and A. K. Katsaggelos, "Motion compensated priority discrete cosine transform coding for image sequences," in *Proc. SPIE VCIP-91*, vol. 1605, Boston, Nov. 1991, pp. 16-25.

[7] S. Farkash, D. Malah, and W. A. Pearlman, "Transform trellis coding of images at low bit rates," *IEEE Trans. Commun.*, vol. 38, pp. 1871-1878, Oct. 1990.

[8] Y. Huang, N. Galatsanos, and H. Dreizen, "Priority DCT coding for image sequences," in *Proc. IEEE Int. Conf. in Acoustics, Speech, and Signal Processing*, Toronto, Canada, 1991, pp. 2629-2632.

[9] M. Kaneko, Y. Hatori, and A. Koike, "Improvements of transform coding algorithm for motion-compensated interframe prediction errors," *IEEE J. Selected Areas Commun.*, vol. SAC-5, pp. 1068-1078, Aug. 1987.

[10] Y. Kato, N. Mukawa, and S. Okubo, "A motion picture coding algorithm using adaptive DCT encoding based on coefficient power distribution classification," *IEEE J. Selected Areas Commun.*, vol. SAC-5, pp. 1090-1099, Aug. 1987.

[11] M. L. Liou, "Visual telephony as an ISDN application," *IEEE Commun.*, vol. 28, pp. 30-38, Feb. 1990.

[12] K. N. Ngan, "Image display techniques using the cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 173-177, Feb. 1984.

[13] K. N. Ngan, K. S. Leong, and H. Singh, "Adaptive cosine transform coding of image in perceptual domain," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1743-1750, Nov. 1989.

[14] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications.* New York: Academic, 1990.

[15] K. Rose, A. Heiman, and I. Dinstein, "DCT/DST alternate-transform image coding," *IEEE Trans. Commun.*, vol. 38, pp. 94-101, Jan. 1990.

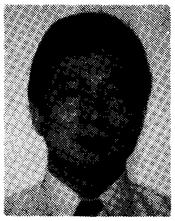
[16] F. Rubin, "Arithmetic stream coding using fixed precision registers," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 672-675, Nov. 1979.

[17] R. C. Reininger and J. D. Gibson, "Distribution of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. COM-31, pp. 835-839, June 1983.

[18] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520-540, June 1987.

[19] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278-1288, Oct. 1986.

[20] G W. Wornell and D. H. Staelin, "Transform image coding with a new family of models," in *Proc. IEEE ICASSP-88*, New York, 1988, pp. 777-780.



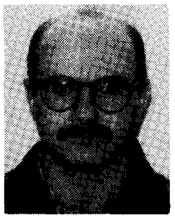
Yunming Huang (S'88-M'90) received the B.S. degree from Nanjing Institute of Technology, China, the M.S. degree from Huazhong University of Science and Technology, China, and the Ph.D. degree from the Illinois Institute of Technology, all in electrical engineering, in 1978, 1983, and 1990, respectively.

He joined Audio DigitalImaging, Inc., IL, in 1989, as a Software Design Engineer. Since then he has been involved in the development of imaging and video systems. He is currently an Engineering Manager, and leads projects of video conferencing and image IC card systems. His research interests include image compression, video processing, and VLSI circuit implementation for digital video communications.

Howard M. Dreizen (M'83) received the B.S. degree in mathematics and computer science in 1975, and the M.S. and Ph.D. degrees in electrical engineering and computer science in 1977 and 1983, respectively, from the University of Illinois, Chicago.

From 1983 to 1989 he was Assistant Professor with the Department of Electrical and Computer Engineering at the Illinois Institute of Technology. He is currently Vice President of EPIX, Inc., in Northbrook, IL.

Dr. Dreizen is a member of the Association for Computing Machinery.



Nikolas P. Galatsanos (M'89) was born in Athens, Greece in 1958. He received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1982, and the M.S. and the Ph.D. degrees in electrical engineering from the University of Wisconsin, Madison in 1984 and 1989, respectively.

Since 1989, he has been on the faculty of the Department of Electrical and Computer Engineering at the Illinois Institute of Technology, Chicago, IL, where he is currently an Assistant Professor. His current research interests include image processing and more specifically recovery, and compression of single and multichannel/frame images.

Dr. Galatsanos is a member of the OSA, and the Technical Chamber of Greece. He currently serves as an Associate Editor for the IEEE Transactons on Transactions on Image Processing.