Scene Adaptive Coder

WEN-HSIUNG CHEN, MEMBER, IEEE, AND WILLIAM K. PRATT, SENIOR MEMBER, IEEE

Abstract—**An efficient single-pass adaptive bandwidth compression technique using the discrete cosine transform is described. The coding process involves a simple thresholding and normalization operation on the transform coefficients. Adaptivity is achieved by using a rate buffer for channel rate equalization. The buffer status and input rate are monitored to generate a feedback normalization factor. Excellent results are demonstrated for coding of color images at 0.4 bits/pixel corresponding to real-time color television transmission over a 1.5 Mbit/s channel.**

I. INTRODUCTION

TRANSFORM image coding, developed about 15 years ago, has been proven to be an efficient means of image coding [1]-[6]. In the basic transform image coding concept, an image is divided into small blocks of pixels, and each block undergoes a two-dimensional transformation to produce an equal-sized array of transform coefficients. Among various transforms investigated for image coding applications, the cosine transform has emerged as the best candidate from the standpoint of compression factor and ease of implementation [7]-[11]. With the basic system, the array of transform coefficients is quantized and coded using a zonal coding strategy [3]; the lowest spatial frequency coefficients, which generally possess the greatest energy, are quantized most finely, and the highest spatial frequency coefficients are quantized coarsely. Binary codes are assigned to the quantization levels, and the code words are assembled in a buffer for transmission. At the receiver, inverse processes occur to decode the received bit stream, and to inverse transform the quantized transform coefficients to reconstruct a block of pixels.

The basic transform image coding concept, previously described, performs well on most natural scenes. A pixel coding rate of about 1.5 bits/pixel is achievable, with no apparent visual degradation. To achieve lower coding rates, without increasing coding error, it is necessary to adaptively quantize transform coefficients so that those blocks of coefficients containing large amounts of energy are allocated more quantization levels and code bits than low energy blocks. In almost all adaptive transform coding designs to date, transforms are computed, and transform energy is measured or estimated on a first pass through the image. This information is then utilized to determine the quantization levels and code words for a second pass [9]. With this scheme, compression factors can be reduced by a factor of two or more as compared to nonadaptive coding. The practical difficulties are the memory required for the second pass and the complexity of the quantization algorithm. Both these problems are eliminated in the scene adaptive coder described in this paper.

The scene adaptive coder is a single-pass adaptive coder of relative simplicity. The following sections describe the coding scheme and present subjective and quantitative performance evaluations.

Paper approved by the Editor for Communication Theory of the IEEE Communications Society for publication after presentation at the International Conference on Communications, Philadelphia, PA, June 1981. Manuscript received September 2, 1982; revised July 8, 1983.

W. Chen is with Compression Labs, Inc., San Jose, CA 95131.
W. K. Pratt is with VICOM Systems, Inc., San Jose, CA 95131.

II. COSINE TRANSFORM REPRESENTATION

The two-dimensional discrete cosine transform of a sequence $f(j, k)$ for $j, k = 0, 1, \cdots, N-1$, can be defined as [6]

$$F(u,v) = \frac{4C(u)C(v)}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \cdot \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (1)$$

for $u, v = 0, 1, \cdots, N-1$, where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } w = \phi \\ 1 & \text{for } w = 1, 2, \cdots, N-1. \end{cases}$$

The inverse transform is given by

$$f(j,k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cdot \cos\left[\frac{(2j+1)u\pi}{2N}\right] \cos\left[\frac{(2k+1)v\pi}{2N}\right] \quad (2)$$

for $j, k = 0, 1, \cdots, N-1$. Among the class of transform possessing fast computational algorithms, the cosine transform has a superior energy compaction property [6]-[9]. The following sections present some other properties of the cosine transform, which are useful to the subsequent discussion.

A. Statistical Description of DCT Coefficients

Let the pixel array $f(j, k)$ represent a sample of a random process with zero mean represented in two's complement format over an integer range $-M \leqslant f(j, k) \leqslant (M-1)$. The probability density of the cosine transform coefficients $F(u, v)$ has been modeled by a number of functions [3], [12]. Among them, the Laplacian density has been shown to provide the best fit [12]. This function can be written as

$$p(x; u, v) = \frac{1}{\sqrt{2}\,\sigma(u,v)} \exp\left\{\frac{\sqrt{2}|x|}{\sigma(u,v)}\right\} \quad (3)$$

where $\sigma(u, v)$ denotes the standard deviation of a coefficient.

B. Coefficient Bound

The maximum coefficient value for the cosine transform can be derived from (1) as

$$F_{\max}(0,0) = 2f_{\max} \quad (4a)$$

and

$$\frac{16}{\pi^2} f_{\max} \leqslant F_{\max}(u,v) \leqslant 2f_{\max} \quad (4b)$$

0090-6778/84/0300-0225$01.00 © 1984 IEEE

where f_{max} is the maximum value of the discrete array $f(j, k)$ and F_{max} is the maximum value for the coefficient $F(u, v)$. If the transform is performed in a vector length of $N = 16$, then

$$F_{max}(0, 0) = 2f_{max} \tag{5a}$$

$$F_{max}(u, v) = 1.628 f_{max}. \tag{5b}$$

C. Mean Square Error Representation

The mean square quantization error between an original image $f(j, k)$ and its reconstructed image $\hat{f}(j, k)$ can be written as

$$\text{MSE} = \frac{1}{N^2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} E\{[f(j, k) - \hat{f}(j, k)]^2\}. \tag{6}$$

The unitary property of the cosine transform allows one to express the MSE in the transform domain as

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} E\{[F(u, v) - \hat{F}(u, v)]^2\} \tag{7}$$

which reduces to

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{n=-\infty}^{\infty} \int_{D_{n-1}}^{D_n} (x - x_n)^2 \cdot p(x; u, v)\, dx \tag{8}$$

where $p(x; u, v)$ is the probability density function, D_n is a set of decision levels, and x_n is a set of reconstruction levels. With Laplacian modeling of the probability density function, as represented in (3), the MSE becomes

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \Bigg\{ \sigma^2(u, v) - \sum_{k=1}^{\infty} \Bigg[(2x_n D_{n-1} + \sqrt{2}\sigma(u, v) x_n - x_n^2) \exp\left(\frac{-\sqrt{2} D_{n-1}}{\sigma(u, v)}\right) - (2x_n D_n + \sqrt{2}\,\sigma(u, v) x_n - x_n^2) \cdot \exp\left(\frac{-\sqrt{2} D_n}{\sigma(u, v)}\right)\Bigg]\Bigg\}. \tag{9}$$

This result has been verified by computer simulation.

III. Scene Adaptive Coder

Fig. 1 contains a block diagram of the scene adaptive coder. In operation, the input image undergoes a cosine transform in 16 × 16 pixel blocks. An initial threshold is established, and those transform coefficients whose magnitudes are greater than the threshold are scaled according to a feedback parameter from the output rate buffer. The scaled coefficients are quantized, Huffman coded, and fed into the rate buffer. The rate buffer operates with a variable rate input, dependent upon the instantaneous image energy, and a constant channel output rate. The buffer status (fullness) and input rate are monitored to generate the coefficient scaling factor. At the receiver, the received fixed rate data are fed to a rate buffer that generates Huffman code words at a variable rate to the decoder. The decoded transform coefficients are then inverse normalized by the feedback parameter, added to the threshold, and inverse transformed to reconstruct the output pixel block. The following sections describe the coding algorithm in greater detail.

A. Cosine Transform

Referring to the block diagram in Fig. 1, the original image is first partitioned into 16 × 16 pixel blocks. Each block of data is then cosine transformed as defined by (1). The resultant transform coefficients $F(u, v)$ are stored in a register according to the zigzag scan of Fig. 2. Scanning the data in this fashion minimizes the usage of runlength codes during the subsequent coding process.

B. Thresholding

The transform coefficients in the register undergo a threshold process in which all the coefficients, except $F(0, 0)$, that are below the threshold are set to zero, and those coefficients above the threshold are subtracted by the threshold. This results in

$$F_T(u, v) = \begin{cases} F(u, v) - T & \text{if } F(u, v) > T \\ 0 & \text{if } F(u, v) \leq T \end{cases} \tag{10}$$

where T is the threshold. Fig. 3 shows a plot of the percentage of coefficients below a threshold versus the threshold for the original images exhibited in Figs. 8(a) and 9(a). As demonstrated, more than 90 percent of the coefficients have absolute magnitude of less than a value of 3, even though the maximum coefficient value could be as large as $1.628 f_{max}$ [see 5(b)]. Therefore, the thresholding process indicated in (10) will set a major portion of coefficients to zero, and thereby limit the number of coefficients to be quantized. The value of the threshold varies with respect to the globally desired bit rate. However, it can be adjusted locally on a block-to-block basis if desired.

C. Normalization and Quantization

The threshold subtracted transform coefficients $F_T(u, v)$ are scaled by a feedback normalization factor D from the output rate buffer according to the relation

$$F_{TN}(u, v) = \frac{F_T(u, v)}{D}. \tag{11}$$

The scaling process adjusts the range of the coefficients such that a desired number of code bits can be used during the coding process.

The quantization process is simply a floating point to integer roundoff conversion. No decision and reconstruction tables are required. Therefore, there is a significant simplification and saving for a hardware implementation. Because many of the threshold subtracted coefficients are of fractional value, the roundoff process will set some of the coefficients to zero and leave only a limited number of significant coefficients to be amplitude coded. The quantized coefficients can now be represented as

$$\hat{F}_{TN}(u, v) = \text{integer part of } [F_{TN}(u, v) + 0.5]. \tag{12}$$

It should be noted that a lower bound has to be set for the normalization factor in order to introduce meaningful transform coefficients to the coder. This lower bound is dependent upon how accurately the cosine transform is computed. Generally speaking, setting the minimum value of D to unity is sufficient for most of the compression applications. In this case, the worst-case quantization error can be obtained from (9) by letting $D_{n-1} = n - 0.5$, $D_n = n + 0.5$, and $x_n = n$.

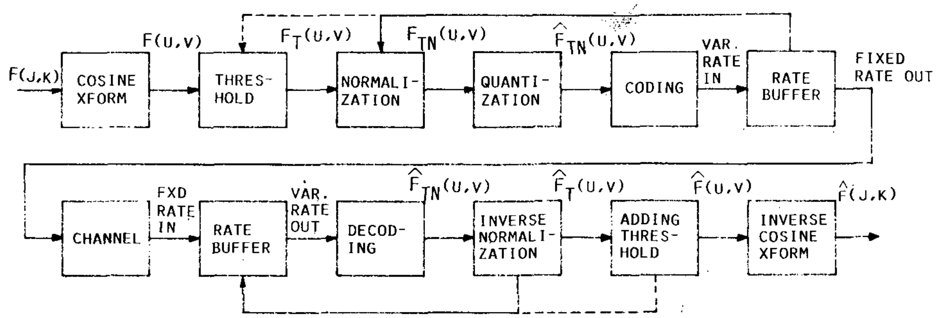


Fig. 1. Block diagram of scene adaptive coding/decoding system.

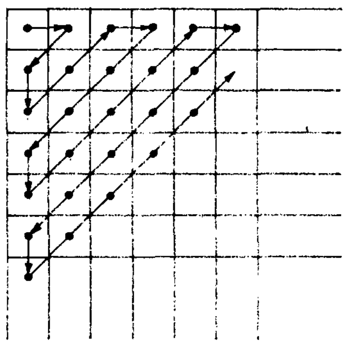


Fig. 2. Zigzag scan of cosine transform coefficients.

Thus

$$\text{MSE} = \frac{1}{4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} e(u, v) \tag{13}$$

where

$$e(u, v) = \sigma^2(u, v) - (1/2) \sum_{n=1}^{\infty} \Bigg\{ [n^2 + (\sqrt{2}\sigma(u, v) - 1)n] \cdot \exp\left[\frac{-\sqrt{2}(n - 0.5)}{\sigma(u, v)}\right] - [n^2 + (\sqrt{2}\sigma(u, v) + 1)n] \exp\left[\frac{-\sqrt{2}(n + 0.5)}{\sigma(u, v)}\right] \Bigg\}.$$

Fig. 4 shows the functional relationship between $e(u, v)$ and $\sigma(u, v)$. As can be seen, $e(u, v)$ is always less than 1/12. (Note: 1/12 is the quantization error for a uniform density.) Therefore, the MSE represented by (13) is always less than $N^2/48$. For $N = 16$, this MSE is less than 16/3, which corresponds to a normalized error of 0.0082 percent. This MSE also corresponds to a peak-to-peak signal-to-quantization-noise ratio of more than 40.9 dB, which is relatively insignificant.

D. Coding

The coefficient $F(0, 0)$ in the upper left-hand corner of each luminance transform block is proportional to the average luminance of that block. Because block-to-block luminance variations resulting from quantization of $F(0, 0)$ are easily discerned visually, $F(0, 0)$ is linearly quantized and coded with a 9 bit code. As for the other nonzero coefficients, their magnitudes are coded by an amplitude lookup table, and the addresses of the coefficients are coded using a runlength lookup table. The amplitude and runlength lookup tables are simply Huffman codes derived from the histograms of typical transform coefficients. As demonstrated by the histograms of Fig. 5, the domination of low amplitudes and short runs of zero-valued coefficients indicates that both Huffman tables are relatively insensitive to the type of input images and the de-

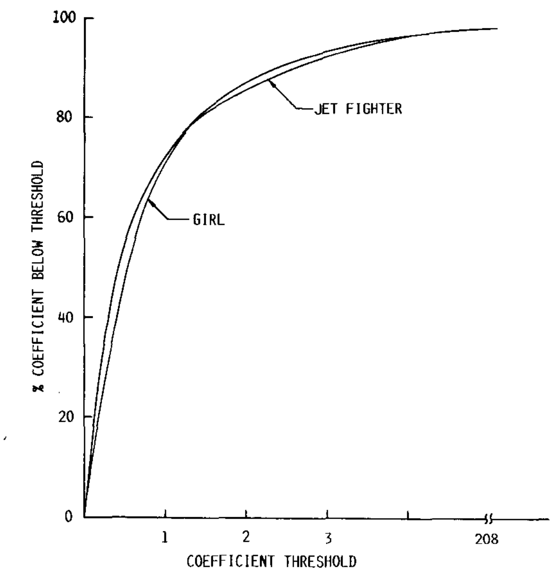


Fig. 3. Distribution of cosine transform coefficients.

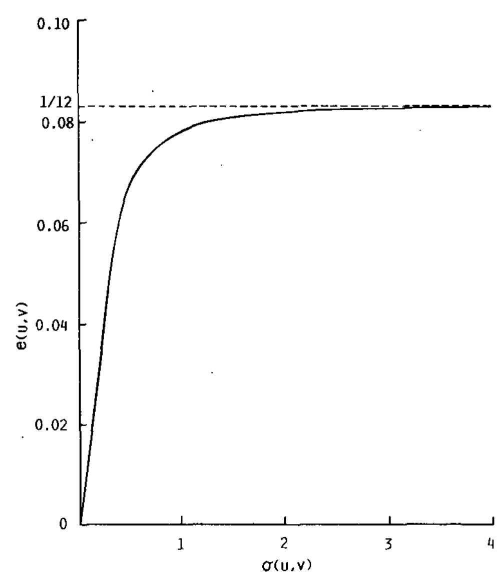


Fig. 4. Quantization error $e(u, v)$ versus coefficient standard deviation $\sigma(u, v)$ at normalization factor of unity and coding threshold of zero. (Probability density function of the coefficient is assumed to be Laplacian.)

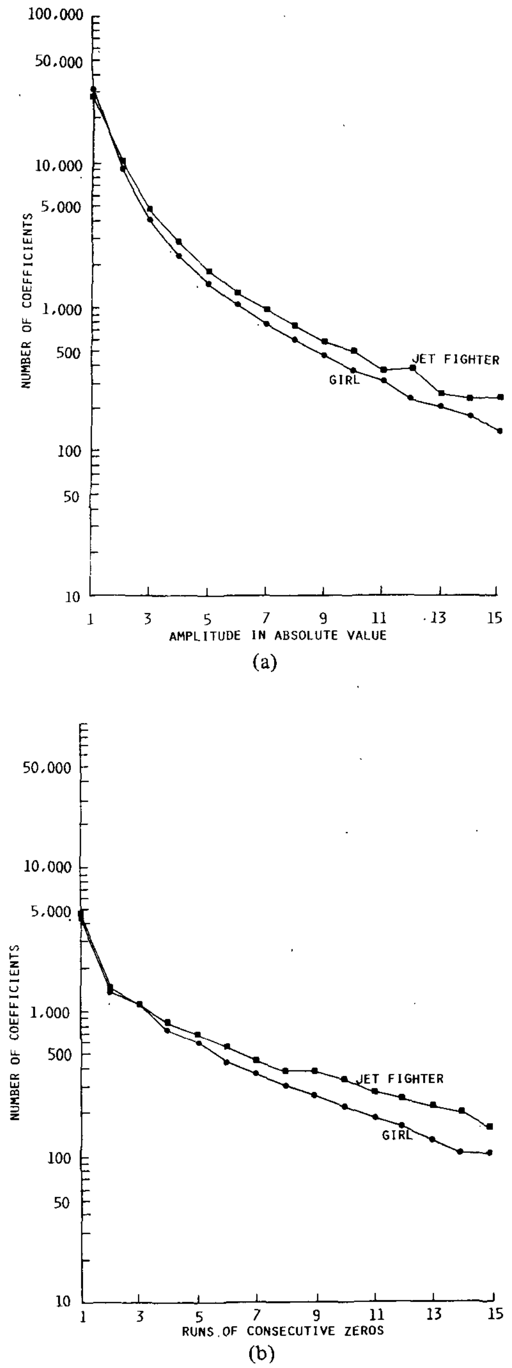


Fig. 5. (a) Histogram of cosine transform coefficients obtained with threshold of zero and normalization factor of one. (b) Histogram of runs of consecutive zero counts obtained with threshold of zero and normalization factor of one.

sired bit rate. This suggests that only two predetermined tables are needed for the coding process.

The length of the Huffman table for the amplitude codes is a function of the normalization factor and the transform bound represented by (5b). For a two-dimensional cosine transform of 16 X 16 pixels, the maximum length is

$$L_A = \frac{1.628\,|f_{\max}|}{D_{\min}} \tag{14}$$

where $D_{\min}$ is the minimum allowable normalization factor. For $|f_{\max}|$ and $D_{\min}$ equal to 128 and 1, respectively, the length will be 208. As for the runlength of zero counts, the

TABLE I
HUFFMAN CODE TABLE FOR COEFFICIENT AMPLITUDE IN ABSOLUTE VALUE

AMPLITUDE	NUMBER OF CODE BITS	HUFFMAN CODES
1	1	1
2	3	001
3	4	0111
4	5	00001
5	5	01101
6	6	011001
7	7	0000001
8	7	0110001
9	8	00000000
10	8	01100000
11	8	00000001
12	8	01100001
13	6+8	000001+8 BITS
EOB	4	0001
RL PREFIX	3	010

length of the table can be represented by

$$L_R = (N^2 - 1) \tag{15}$$

where N is the transform block size. The subtraction of unity is because the dc coefficient in individual blocks is separately encoded. For a transform block size of 16 X 16 pixels, the length is 255. In practice, the length of both tables can be shortened to less than 30 entries by assigning Huffman codes to only the low amplitude coefficients and short runlengths (using a fixed length code elsewhere). Again, due to the domination of the low amplitude coefficients and short runlengths, the loss of coding efficiency is insignificant.

Tables I and II show typical truncated Huffman code tables for the amplitude and runlength, respectively. It should be noted that the amplitude codes in Table I include an "end of block (EOB)" code and a "runlength prefix" code. The EOB code is used to terminate coding of the block as soon as the last significant coefficient of the block is coded. The runlength prefix code is required in order to distinguish the runlength code from the amplitude code.

It should be noted that there are many ways to improve the coding efficiency of the coder. One way is to cut down the number of runs in the runlength coder. This can be accomplished by skipping single isolated coefficients with absolute magnitude of one, or by introducing an amplitude code for the isolated coefficients with zero amplitude. The coding improvement is generally quite significant; this is especially true if the average coding rate is low.

E. Rate Buffer

The rate buffer in the SAC performs channel rate equalization. The buffer has a variable rate data input and a constant data output. The differentials are monitored from block to block, and the status is converted into a scaling factor that is fed back to the normalizer. The buffer always forces the coder to adjust to the local coding variations, while ensuring global performance at a desired level. The general operation of a rate buffer is well documented [13], [14]. The specific method used in this paper is described as follows.

Let $B(m)$ represent the number of bits into the rate buffer for the mth block and let $S(m)$ represent the normalized buffer status at the end of the mth block ($-0.5 < S(m) < 0.5$). Then, $B(m)$ and $S(m)$ can be written as

$$B(m) = \sum_{\substack{u=0 \\ (u,v)\neq(0,0)}}^{15} \sum_{v=0}^{15} H\{[\hat{F}_{TN}(u,v)]_m\} + 9 \tag{16}$$

TABLE II
HUFFMAN CODE TABLE FOR THE NUMBER OF CONSECUTIVE ZERO-VALUED COEFFICIENTS

RUN-LENGTH	NUMBER OF CODE BITS	HUFFMAN CODE
1	2	11
2	3	101
3	3	011
4	4	0101
5	4	0011
6	5	01000
7	5	10010
8	5	01001
9	5	10001
10	5	10011
11	6	001000
12	6	100000
13	6	001010
14	6	001001
15	6	100001
16	6	000011
17	6	001011
18	7	0000000
19	7	0000100
20	7	0000010
21	7	0001110
22	7	0000001
23	7	0000101
24	7	0000011
25	7	0001111
26	8	00011000
27	8	00011010
28	8	00011001
29	8	00011011
30	5+8	00010+8 BITS

$$S(m) = S(m-1) + \frac{[B(m) - 256R]}{L} \tag{17}$$

where

$[\hat{F}_{TN}(u, v)]_m$	quantized coefficients of the mth block, as defined in (12)
$H\{\cdot\}$	Huffman coding function
R	average coding rate
L	rate buffer size.

The buffer status $S(m)$ is used to select an instantaneous normalization factor $\hat{D}(m)$ according to an empirically determined "normalization factor versus status" curve. This relationship is described by

$$\hat{D}(m) = \Phi\{S(m)\}. \tag{18}$$

In order to smooth out this instantaneous normalization factor such that the desired normalization factor does not fluctuate too much, a recursive filtering process is applied to generate

$$D(m) = cD(m-1) + (1-c)\hat{D}(m) \tag{19}$$

where c is a constant with value less than unity.

The desired operating conditions for the rate buffer algorithm are: a) the feedback normalization factor is as stable as possible; b) the buffer status is able to converge rapidly and stay as close to the half full position ($S(m) = 0$) as possible. Both these conditions may be satisfied using the above set of equations. Fig. 6 shows typical values of the normalization factor and buffer status as a function of block indexes for the images shown in Figs. 8(a) and 9(a).

The rate buffer can be guaranteed not to overflow. This is because the normalization factor can get very large within a few blocks of operation, and effectively limit the data going into the buffer. However, there is no guarantee that the buffer will not underflow if a minimum allowable normalization factor is set to a fixed value. Therefore, the buffer status has to be constantly monitored and, if the status is closer to −0.5, fill bits must be introduced into the channel.

IV. SCENE ADAPTIVE CODING OF COLOR IMAGES

Fig. 7 contains a block diagram of a color image coding system based on the scene adaptive coder. In this system, a color image, represented by tristimulus signals $R(j, k)$, $G(j, k)$, $B(j, k)$, is first converted to a new three-dimensional space defined by

$$\begin{bmatrix} Y(j,k) \\ I(j,k) \\ Q(j,k) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.589 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.253 & 0.312 \end{bmatrix} \begin{bmatrix} R(j,k) \\ G(j,k) \\ B(j,k) \end{bmatrix} \tag{20}$$

where $Y(j, k)$ is the luminance signal and $I(j, k)$ and $Q(j, k)$ are chrominance signals. This conversion compacts most of the signal energy into the Y plane such that more efficient coding can be accomplished [15]. The I and Q chrominance planes are spatially averaged and subsampled by a factor of 4 to 1 in both the horizontal and vertical directions. The luminance and subsampled chrominance images are then partitioned into 16×16 pixel blocks and coded by the SAC in the order of 32 Y, two I, and two Q sequences. At the receiver, the received code bits are decoded. Inverse cosine transform and inverse coordinate conversions are then performed to reconstruct the source tristimulus signals. The inverse coordinate conversion is described by

$$\begin{bmatrix} R(j,k) \\ G(j,k) \\ B(j,k) \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y(j,k) \\ I(j,k) \\ Q(j,k) \end{bmatrix}. \tag{21}$$

V. SIMULATION RESULTS

Computer simulations have been conducted to evaluate the performance of the scene adaptive coder. The original test images shown in Figs. 8(a) and 9(a) are of size 512×512 pixels with each red, green, and blue tristimulus value uniformly quantized to 8 bits/pixel. Figs. 8(b) and 9(b) show the reconstructed images at a combined average bit rate of 0.4 bits/pixel. This rate corresponds to a channel bandwidth of 1.5 Mbits for a 15 frame/s intraframe coding system. The excellent reconstruction of the images is clearly demonstrated. Table III tabulates the average mean square error between the original and the reconstructed images. Also included in the table is the peak signal-to-noise ratio for the reconstructed image.

VI. SUMMARY

The scene adaptive coder described herein encodes cosine transform coefficients in a simple manner. The coding process involves only thresholding, normalization, roundoff, and rate buffer equalization. The performance of the coder is quite good in terms of mean square error and subjective evaluation. Because the coding process is dependent upon the instantaneous coefficient content inside the block and the accumulated rate buffer content, it is well suited for intraframe coding of moving images. At Compression Labs, Inc., the coder has been implemented with real-time hardware to code NTSC color video at a channel rate of 1.5 Mbits/s.

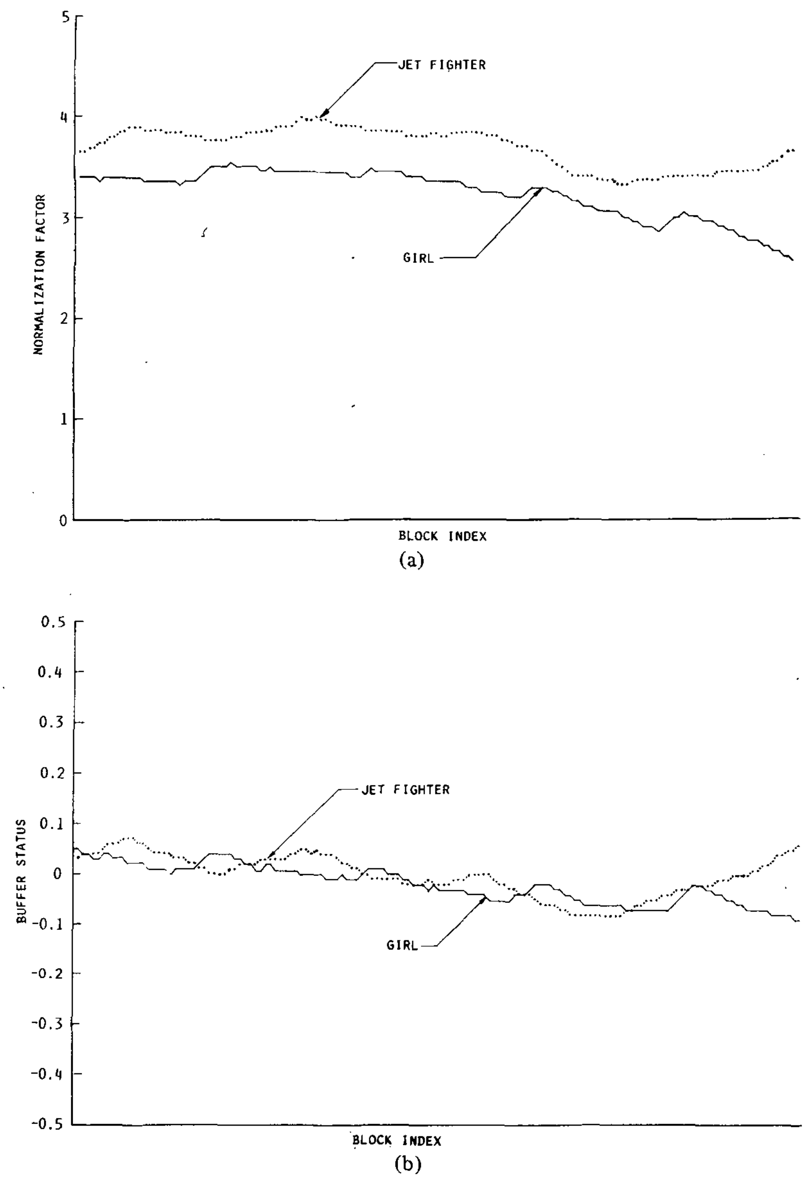


Fig. 6. Buffer status for the last 114 blocks of images shown in Figs. 8(a) and 9(a).

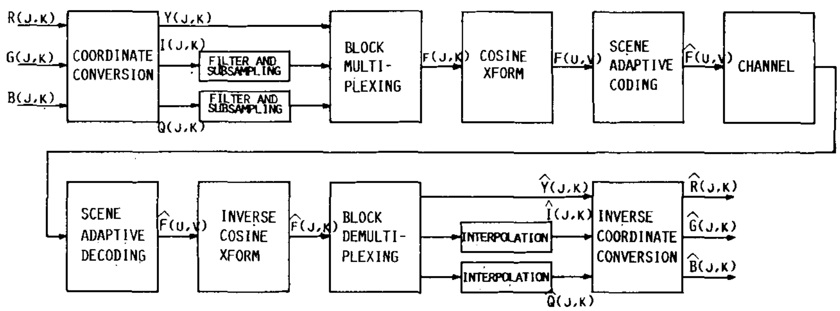


Fig. 7. Cosine transform color image coding/decoding system.

(a)

(b)

Fig. 8. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.

(a)

(b)

Fig. 9. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.

TABLE III
MEAN SQUARE ERROR BETWEEN ORIGINAL AND RECONSTRUCTED IMAGES AT 0.4 BITS/PIXEL; MEAN SQUARE ERROR IS COMPUTED AT THE INPUT OF CODER AND THE OUTPUT OF DECODER WITH $f_{max}(j, k)$ NORMALIZED TO 1

IMAGES	MEAN SQUARE ERROR	SNR
GIRL	0.0359%	34.45 DB
JET	0.0521%	32.84 DB

ACKNOWLEDGMENT

The authors wish to express their sincere gratitude to Dr. A. Tescher for his valuable discussions and suggestions in the area of rate buffer stabilizations.

REFERENCES

[1] H. C. Andrews and W. K. Pratt, "Fourier transform coding of images," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan. 1968, pp. 677–678.

[2] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, pp. 58–68, Jan. 1969.

[3] W. K. Pratt, W. H. Chen, and R. Welch, "Slant transform image coding," *IEEE Trans. Commun.*, vol. COM-22, pp. 1075–1093, Aug. 1974.

[4] A. Habibi and P. A. Wintz, "Image coding by linear transformation and block quantization," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 50–62, Feb. 1971.

[5] W. K. Pratt, *Digital Image Processing*. New York: Wiley-Interscience, 1978.

[6] N. Ahmed, T. Natarjan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.

[7] M. Hamidi and J. Pearl, "Comparison of the cosine and Fourier transforms of Markov-1 signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 428–429, Oct. 1976.

[8] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, Oct. 1979.

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

[10] W. H. Chen, C. H. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004–1009, Sept. 1977.

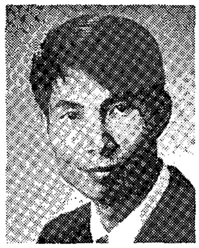
[11] R. M. Haralick, "A storage efficient way to implement the discrete cosine transform," *IEEE Trans. Comput.*, vol. C-25, pp. 764–765, July 1976.

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

[13] A. Tescher, "Rate adaptive communication," in *Proc. Nat. Telecommun. Conf.*, 1978.

[14] ——, "A dual transform coding algorithm," in *Proc. Nat. Telecommun. Conf.*, 1980.

[15] W. K. Pratt, "Spatial transform coding of color images," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 980–992, Dec. 1971.



Wen-hsiung Chen (M'82) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1962, the M.S. degree from Kansas State University, Manhattan, in 1966, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1973, all in electrical engineering.

From 1966 to 1968 he was employed with the Allis-Chalmers Company, Harvey, IL, responsible for solid-state circuitry development. He was a Teaching and Research Assistant and Research Associate at the Department of Electrical Engineering, University of Southern California, from 1968 to 1973. In 1973 he joined Ford Aerospace and Communications Corporation, Palo Alto, CA, to work on communication systems analysis and digital image processing. In 1977 he left to help form Compression Labs, Inc., San Jose, CA, where he developed the combined symbol matching coder for coding facsimile images, the intraframe coder for coding NTSC color video at 1.5 Mbits, and the combined intraframe and interframe coder for coding color video at less than 750 kbits. As Chief Scientist of the company, he is responsible for research and development work in digital image processing.

★

William K. Pratt (S'57–M'61–SM'75) received the B.S. degree in electrical engineering from Bradley University, Peoria, IL, in 1959 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1961 and 1965, respectively.

He received Master's and Doctoral Fellowships from Hughes Aircraft Company and was employed there from 1959 to 1965. He became an Assistant Professor of Electrical Engineering at the University of Southern California in 1965, an Associate Professor in 1969, and a Full Professor in 1975. At U.S.C. he was the Director of the Image Processing Institute and of the Engineering Computer Laboratory. He is now the President and Chairman of the Board of Vicom Systems, Inc., San Jose, CA.

In 1976 Dr. Pratt was awarded a Gugenheim Fellowship for research and image analysis techniques. He is a member of Sigma Tau, Omicron Delta Kappa, and Sigma Xi.

(a)

(b)

Fig. 8. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.

(a)

(b)

Fig. 9. Cosine transform scene adaptive coding. (a) Original. (b) Reconstructed image at 0.4 bits/pixel.