Constrained Transform Coding and Surface Fitting

LAYNE T. WATSON, ROBERT M. HARALICK, SENIOR MEMBER, IEEE, AND OSCAR A. ZUNIGA

Abstract—**A constrained transform coding procedure is developed which is a combination of transform coding with differential pulse code modulation. The algorithm avoids block boundary mismatch errors, yet retains the coding efficiency of transform coding. A general theory of constrained transform coding is developed which includes the discrete cosine transformation and tensor products of splines as special cases. Results using the cosines and splines are given for two images. A complete discussion of the necessary linear algebra background is also given.**

I. INTRODUCTION

The constrained transform coding technique grew out of a search for the correspondences between transform coding, surface fitting, and approximation theory. The viewpoint is that of function approximation and sophisticated numerical linear algebra is used. All theorems shown pertain to only one-band imagery. There is a natural and straightforward extension to multiband imagery. In this paper, we illustrate that transform coding is nothing more than least squares surface fitting, albeit on a piece-by-piece basis. When transform coding is put into this framework, the reason why errors occur at block bound-

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society for publication after presentation at the SIAM National Meeting, Alexandria, VA, June 1980. Manuscript received December 1, 1981; revised October 12, 1982.

L. T. Watson is with the Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

R. M. Haralick is with the Departments of Electrical Engineering and Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

O. A. Zuniga is with the Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

0090-6778/83/0500-0717$01.00

aries in the highly compressed transform coded images is apparent: there has never been any constraint in transform coding that the surfaces must match up at the block boundaries.

If a picture is transform coded block by block in a left-to-right, top-to-bottom scan, such a surface matching constraint requires that when the current block's surface is extended to the block on its left and to the block above, then these surfaces must match. We can, of course, extend the matching concept to making the surfaces match up to an nth derivative. In any case, requiring that the left and top edges match leaves us in a situation where, knowing nothing more than the left and top boundaries of the current block's surface, we may interpolate or estimate the remainder of the block's surface. As in differential pulse code modulation, the differences between the actual surface and the estimated surface can be transmitted. However, these differences do not have to be transmitted in a pixel-by-pixel manner as in differential pulse code modulation. Rather they are transmitted parametrically as surface differences. The parameters are exactly analogous to the coefficients transmitted in a transform coding procedure.

Thus, what has happened is this: we have defined a constrained transform coding procedure which by its nature is a combination of transform coding with differential pulse code modulation. The new procedure has the property that there will be no block boundary mismatch errors and it retains the coding efficiency advantage of transfrom coding.

The use of transforms for image data compression was introduced by Andrews, Kane, and Pratt [4]. They advocated the Hadamard transform applied to the entire image. Tasto and Wintz [19], Habibi and Wintz [11], and Wintz [21] suggested breaking the image up into blocks and transform coding each block. Various kinds of quantizations were applied to the transform coefficients. Various kinds of basis functions were used. Anderson and Huang [2], [3] suggested the discrete Fourier basis. Pratt, Chen, and Welch [16] suggested the slant transform. Rao, Narasimhan, and Revuluri [17] suggested the Haar transform. Ahmen, Natarajan, and Rao [1] suggested the discrete cosine transform. Habibi [10], [12] suggested combining a transform technique by line with a DPCM technique by column. Haralick and Shanmugan [14] experimented with a block transform technique followed by DPCM. Andrews and Patterson [5] illustrated the computationally expensive singular value decomposition technique for data compression.

More recent advances in image data compression have been achieved by adaptive quantization techniques. Chen and Smith [6] discuss adaptive quantization of monochrome and color images with a transform coding technique. Zschunke [22] discusses adaptive quantization in conjunction with a DPCM data compression technique. Habibi [13] gives a survey of adaptive image coding techniques. Haralick and Zuniga [15] illustrate how a rate buffer can be integrated into an adaptive quantizing for DPCM or transform coding data compression.

In order to describe this constrained transform coding procedure in detail we need to use some concepts and theorems about orthogonality and orthogonal projection operators. The Appendix contains a concise, yet complete, review of the necessary definitions and theorems from linear algebra. The following sections use the terminology and theorems of the Appendix. Section II shows how surface or function approximation in the discrete least squares sense corresponds to orthogonal projection. Section III applies the surface fitting and orthogonal projection ideas to constrained transform coding data compression and illustrates how the combination is a mixture of transform coding with differential pulse code modulation.

II. DISCRETE LEAST SQUARES AND ORTHOGONAL PROJECTION

In this section, we illustrate that discrete least squares fitting a set $\{(x_i, y_i); i = 1, \cdots, K\}$ of points, whose x coordinate is the independent variable and whose y coordinate is the dependent variable, with respect to a set of functions $\{f_n(x); n = 1, \cdots, N\}$ is exactly the same problem as taking the orthogonal projection of the vector

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_K \end{pmatrix}$$

onto the space spanned by the vectors

$$\begin{pmatrix} f_n(x_1) \\ \vdots \\ f_n(x_K) \end{pmatrix},$$

$n = 1, \cdots, N$. It is assumed that $N < K$, and that the functions $f_n(x)$ are independent with respect to the points x_i. Precisely,

$$\sum_{i=1}^{N} \alpha_i f_i(x_k) = 0$$

for $k = 1, \cdots, K$ implies $\alpha_1 = \cdots = \alpha_N = 0$.

Imagine the data points (x_i, y_i) as lying on the graph of some function $g(x)$, and let L be a vector space (of functions) containing g and $f_1, \cdots, f_n$. Define an inner product on L by

$$\langle h, k \rangle = \sum_{i=1}^{K} h(x_i)k(x_i). \tag{1}$$

(Actually, this may only be a positive *semidefinite* hermitian form, but this technical subtlety is irrelevant for our purposes.) The problem of finding the best approximation $\hat{g}$ to g by a linear combination of $f_1, \cdots, f_n$ with respect to this inner product is

$$\begin{aligned} \min_{\alpha} \left\| g - \sum_{i=1}^{N} \alpha_i f_i \right\|^2 &= \min_{\alpha} \left\langle g - \sum_{i=1}^{N} \alpha_i f_i, g - \sum_{i=1}^{N} \alpha_i f_i \right\rangle \\ &= \min_{\alpha} \sum_{k=1}^{K} \left(y_k - \sum_{i=1}^{N} \alpha_i f_i(x_k) \right)^2 \\ &= \min_{\alpha} \| y - B\alpha \|_2^2, \end{aligned} \tag{2}$$

where

$$\alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}, \quad B = \begin{pmatrix} f_1(x_1) \cdots f_N(x_1) \\ \vdots \qquad\quad \vdots \\ f_1(x_K) \cdots f_N(x_K) \end{pmatrix}$$

and $\|\cdot\|_2$ is the ordinary length in E^K. This shows that the function approximation problem with respect to the inner product (1) is equivalent to an ordinary least squares problem in E^K. By Proposition 3 the (unique) solution

$$\hat{g} = \sum_{i=1}^{N} \hat{\alpha}_i f_i$$

is the orthogonal projection of g onto the subspace M with basis $f_1, \cdots, f_N$. $\hat{g}$ is called the best discrete least squares approximation to g. It also follows from Propositions 3 and 4 that

$$\hat{\alpha} = (B^t B)^{-1} B^t y$$

and

$$\hat{y} = B\hat{\alpha} = B(B^t B)^{-1} B^t y$$

is the vector of fitted values at $x_1, \cdots, x_K$. Hence, $\hat{y}$ is the orthogonal projection of y onto the column space of B.

III. CONSTRAINED TRANSFORM CODING

Let T be a matrix whose columns are the desired basis vectors of some transform coding scheme. The columns of T need not be orthonormal. Let y be the vector representing the pixels in the block currently being transform coded and let α be the vector representing the transformed coefficients for the block y.

In accordance with the identity between discrete least squares function fitting and orthogonal projections, each column of T is a function sampled at the same specified values, which for the case of transform coding is just the spatial coordinates of the pixels in the block of y, and $T\alpha$ is the sampled values of the fitted surface at the pixel locations on the block. Now, these functions which are sampled to form the columns of T can be sampled at each of the spatial coordinates of the exterior top and left borders of the block y, thereby defining a matrix S having the same number of columns as T and having one row for each exterior top or left border pixel. The extrapolation of the estimated surface of the block y to the coordinates at the exterior top and left border pixels of y is then given as $S\alpha$.

Let z be the vector specifying the values of the surface at the exterior top and left border pixels of block y. Constraining the fitted surface to match at the top and left border of the block amounts to requiring that α minimize $\|z - S\alpha\|$. Therefore, the constrained transform coding problem is determining the coefficient vector α which minimizes $\|y - T\alpha\|$, subject to the constraint that α minimizes $\|z - S\alpha\|$ first.

The next proposition specifies that the α which achieves this constrained minimization can be represented as $\alpha_p + \alpha_h$ where α_p is any vector minimizing $\|z - S\alpha_p\|$ and $T\alpha_h$ is the orthogonal projection of $y - T\alpha_p$ onto $T \ker S$. This leads to, of course, the procedure for determining α.

Proposition 8: Let T be a $K \times N$ matrix, S an $m \times N$ matrix, $m < N < K$, rank $T = N$, rank $S = r \leq m$, y a K-vector, and z an m-vector. Then, the solution to

$$\min_{\alpha \in \Omega} \|y - T\alpha\|, \quad \Omega = \{\alpha \mid \alpha \text{ minimizes } \|z - S\alpha\|\},$$

has the form $\alpha = \alpha_p + \alpha_h$ where $\alpha_p \in \Omega$ and $T\alpha_h$ is the orthogonal projection of $y - T\alpha_p$ onto $T \ker S$.

Proof: Let $\alpha_p \in \Omega$ be fixed. Then, every $\alpha \in \Omega$ has the form $\alpha = \alpha_p + \alpha_h, \alpha_h \in \ker S$ ($\ker S \neq \phi$ since $m < N$). Now

$$\|y - T\alpha\| = \|y - T(\alpha_p + \alpha_h)\|$$

$$= \|(y - T\alpha_p) - T\alpha_h\| = \|(y - T\alpha_p) - TB\beta\|$$

where B is a matrix whose columns are an orthonormal basis for $\ker S$. Now, clearly $\|y - T\alpha\|$, $\alpha \in \Omega$, is minimized for some $\hat{\beta}$ such that $TB\hat{\beta} = T\alpha_h$ is the orthogonal projection of $y - T\alpha_p$ onto $T \ker S$. Q.E.D.

Using Proposition 4 an explicit formula for α_h is

$$\alpha_h = B[(TB)^t TB]^{-1}(TB)^t (y - T\alpha_p). \tag{3}$$

Corollary 9: If the columns of T are orthonormal, then α_h is the orthogonal projection of $T^t y - \alpha_p$ onto $\ker S$.

Proof: Using the fact that the columns of B and T are orthonormal in (3) gives

$$\alpha_h = BB^t(T^t y - \alpha_p),$$

from which the corollary follows by Proposition 6. Q.E.D.

As alluded to earlier, α_p and α_h should not actually be calculated by formulas such as (3), which are notoriously inaccurate. Numerically accurate and stable computations of α_p and α_h are based on the matrix decompositions given in the next two propositions.

Proposition 10: Let A be an $n \times k$ matrix, $k \leq n$. Then, there exists an $n \times n$ orthogonal matrix Q such that

$$QA = R$$

is upper triangular.

This is called the QR factorization of A, and is extremely important in the numerical calculation of eigenvalues and the numerical solution of least squares problems. The calculation of Q and R is numerically stable, and can done accurately and efficiently using Householder reflections. For the details see [20], the bible of numerical linear algebra. Examining the formula $A = Q^t R$ column by column shows that the first k columns of Q^t are an orthonormalization of the columns of A, and thus, Proposition 10 provides a proof for Proposition 5.

If A has less than full rank the appropriate decomposition is the following.

Theorem 11: Singular Value Decomposition. Let A be an $n \times k$ matrix, $k \leq n$. Then, there exist an $n \times n$ orthogonal matrix U, a $k \times k$ orthogonal matrix V, and an $n \times k$ diagonal matrix Σ with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k \geq 0$ such that

$$U^t A V = \Sigma.$$

The numbers $\sigma_1, \cdots, \sigma_k$ are called the singular values of A, and are uniquely determined (although U and V are not

unique). Observe that

$$\text{rank } A = \text{rank } \Sigma,$$

and, thus, the σ_i are good indicators of the "independence" of the columns of A. The numerical calculation of the singular value decomposition is also based on Householder reflections, and is numerically stable although relatively expensive [20].

Now these results will be applied to the transform coding scheme. Consider the problem of finding ker S. Then, the following equivalent statements:

$$x \in \ker S,$$

$$Sx = 0,$$

x is orthogonal to the row space of S,

$$x \in (\text{column space of } S^t)^{\perp},$$

show that ker S is the orthogonal complement of the subspace spanned by the columns of S^t. Using Proposition 10,

$$S^t = QR$$

where Q is an $N \times N$ orthogonal matrix and R is upper triangular (assume, without loss of generality, that the first $r = \text{rank } S^t$ columns of R are independent). As observed earlier, the first r columns of Q are an orthonormalization of the columns of S^t. Since the columns of Q are an orthonormal basis for E^N, the last $N - r$ columns of Q are an orthonormal basis for the orthogonal complement of the column space of S^t. Therefore, an orthonormal basis for ker S is the last $N - r$ columns of Q. By Proposition 6, this will be particularly nice for computing projections onto ker S.

Now let A be an $n \times k$ matrix, $n \geq k$, $b \in E^n$, and consider the least squares problem

$$\min_x \| Ax - b \|.$$

The computation of both α_p and $\hat{\beta}$ ($\alpha_h = B\hat{\beta}$) reduces to such a problem. By Proposition 3, this is equivalent to finding the projection of b onto the column space of A. There are two reasons for not using the explicit formula for the projection operator in Proposition 4. First, the formula requires A to have full rank. Secondly, A^tA is typically extremely ill conditioned, which may result in a serious loss of accuracy in the calculation of $(A^tA)^{-1}$ and $x = (A^tA)^{-1}A^tb$. By Theorem 11,

$$U^tAV = \Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_r & \\ & & & & 0 \\ & & & & & \ddots \\ & & & & & & 0 \end{pmatrix}$$

where $r = \text{rank } A$. Using Proposition 7, $\| Ax - b \| = \| U\Sigma V^tx - b \| = \| \Sigma V^tx - U^tb \| = \| \Sigma\hat{x} - \hat{b} \|$ where $\hat{x} = V^tx$ and $\hat{b} = U^tb$. Now, clearly $\| \Sigma\hat{x} - \hat{b} \|$ is minimized when $\hat{x}_i = \hat{b}_i/\sigma_i$, $i = 1, \cdots, r$, and $\hat{x}_i = 0$, $i = r + 1, \cdots, k$. It is also clear that $\hat{x}$ is unique if rank $A = r = k$, and that if rank $A < k$ the above $\hat{x}$ is the unique solution of minimum norm. These statements apply also to the solution $x = V\hat{x}$ to the original problem.

This completes the description of the numerical calculation of α_p and α_h.

To simplify the discussion, assume that the columns of T are orthonormal. The argument for the general case is analogous. By the representation of α, we have actually shown how to mix transform coding with the differential pulse coding scheme. However, our differential pulse coding will not be on a pixel-by-pixel basis. It will be by surface parameters of a block, making it, therefore, very efficient. To see how this can work, consider the situation at the transmitter and receiver at the time block y is being encoded. Its top and left exterior border pixels z are known to the receiver from the transmission of the previous blocks. The matrices T and S are also known to the transmitter and receiver. The vector α_p is any vector which minimizes $\| z - S\alpha_p \|$. Since the number of columns exceeds the number of rows in S, there are many such α_p vectors. Since z and S are known to the receiver, it can determine an α_p. As in differential pulse code modulation, the transmitter can mimic the receiver and construct this same α_p.

Now by Corollary 9 α_h is given by

$$\alpha_h = BB^t(T^ty - \alpha_p) = B[B^t(T^ty - \alpha_p)].$$

The transmitter can now send $B^t(T^ty - \alpha_p)$. Multiplying this by B the receiver obtains α_h. The receiver then adds α_p to α_h to obtain α and produces the reconstructed block at $T\alpha$, $T\alpha = T\alpha_p + T\alpha_h$.

$T\alpha_p$ is the receiver's interpolation estimate of the block using the top and left exterior border z. $TT^ty - T\alpha_p$ is the difference between the best fit sampled surface of y according to the column space of T and the sampled surface the receiver has estimated. $T^ty - \alpha_p$ is the same difference except it is in the parameter space of the surface rather than in the values of the surface points. The transmitter sends the orthogonal projection of $T^ty - \alpha_p$ onto ker S instead of the difference $T^ty - \alpha_p$ in order that the reconstructed surface at the receiver match its top and left borders.

To see how this works in an example problem with numbers, consider a block which is 16×16. The y is a 256×1 vector. The top and left border pixels number 33. Hence, z is a 33×1 vector. Suppose we approximate the surface of the extended 17×17 block by a 49-dimensional surface. Then α is a 49×1 vector. The matrix S takes α into z, so it must be a 33×49 matrix. Ker S has dimension $r \geq 49 - 33 = 16$. The matrix B, therefore, is $49 \times r$. The matrix T takes α into y, so it must be a 256×49 matrix. The transmitter and receiver both construct the same α_p. The transmitter sends $B^t(T^ty - \alpha_p)$. This is an $r \times 1$ vector. The receiver multiplies what the transmitter sends by B obtaining

$$\alpha_h = B[B^t(T^ty - \alpha_p)].$$

Then, it adds α_p and α_h to obtain α and finally reconstructs the block as $T\alpha$. The achieved component compression ratio is $256/r \cong 16{:}1$. (The matrices B, S, and T are fixed, and only two matrix multiplications are needed to reconstruct the block. With an array processor, this is feasible in real time.)

The above discussion assumes an ideal channel and no quantization error. Further compression can be achieved by quantization and Hufman encoding (which was done in the experiments), but the intent here is to study the effect of constrain-

ing the transform coding, and not the secondary effects of quantization error, interpolation errors, or noisy channels.

IV. SURFACE APPROXIMATION BY COSINES (DCT)

Consider an L-by-L block as lying in the unit square in the plane, with pixels at the x-y coordinates $((2i-1)/2L, (2j-1)/2L)$, $i, j = 1, \cdots, L$. The basis functions for the surface approximation discussed in Section II are taken to be

$$\cos r\pi x \cos s\pi y, \quad r, s = 0, 1, \cdots, L-1.$$

This gives $K = L^2$ independent functions, and the grey tone surface is to be approximated by some linear combination of $N < K$ of these functions. In some fashion, order the L^2 pixel coordinates, calling them $P_1, \cdots, P_k$, and similarly order the basis functions, calling them $f_1, \cdots, f_K$. These cosine functions have the extremely useful property of being discretely orthogonal with respect to the pixel points, i.e., the vectors

$$\begin{pmatrix} f_i(P_1) \\ \vdots \\ f_i(P_K) \end{pmatrix}$$

are mutually orthogonal. This can be shown by observing that the pixel coordinates in the x and y directions are related to the zeros of the Lth Chebyshev polynomial $T_L(x) = \cos(L \cos^{-1} x)$, and then using the fact that the first L Chebyshev polynomials are discretely orthogonal with respect to the zeros of the Lth Chebyshev polynomial. Let $g(P_i)$ be the grey tone at pixel point P_i, let

$$T = \begin{pmatrix} f_1(P_1) & \cdots & f_N(P_1) \\ \vdots & & \vdots \\ f_1(P_K) & \cdots & f_N(P_K) \end{pmatrix}$$

be the matrix of sampled basis functions, let

$$\alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{pmatrix}$$

be the (unknown) vector of coefficients of these basis functions, and let

$$y = \begin{pmatrix} g(P_1) \\ \vdots \\ g(P_K) \end{pmatrix}$$

be the vector of grey tone values. Then, the least squares surface approximation problem is

$$\min_{\alpha} \| T\alpha - y \|.$$

Since the columns of T are orthogonal, computing the projection of y onto the column space of T is trivial. *This projection calculation for α is precisely the discrete cosine transformation of y.*

V. SURFACE APPROXIMATION BY CUBIC SPLINES

Consider a block as lying in the unit square in the plane, with corner pixels having coordinates (0, 0), (1, 0), (1, 1), (0, 1). Let P be a positive integer, $x_i = i/P$, $y_i = i/P$, $i = -3, -2, \cdots, P+3$. Let $B_k(x)$, $k = -1, \cdots, P+1$ be the cubic B-splines defined at the knots x_i, and similarly for $B_k(y)$ at y_i. The B_k are defined by [8]

$$B_k(s) = \frac{4}{P}(x-s)_+^3[x_{k-2}, x_{k-1}, \cdots, x_{k+2}].$$

The $P+3$ B-splines $B_{-1}(x), \cdots, B_{P+1}(x)$ are a basis for the linear space of all cubic splines with knots at $x_0, \cdots, x_P$ [7].

The intent is to approximate (in a discrete least squares sense) the grey tone surface over the block by a tensor product of cubic splines [8]

$$p(x) \otimes q(y)$$

where $p(x)$, $q(y)$ are cubic splines with knots at x_i, y_i, respectively ($i = 0, \cdots, P$). By the preceding remarks, a basis for all such splines is

$$B_i(x) \otimes B_j(y), \quad -1 \leqslant i, j \leqslant P+1.$$

Denote the pixel coordinates (in the block under consideration) by (u_i, v_j) and the grey tone at (u_i, v_j) by $g(u_i, v_j)$, $i, j = 1, \cdots, L$ where $K = L^2$ and $N = (P+3)^2$ are the same as in Section II. Then, the least squares surface approximation problem is

$$\min_{\alpha} \sum_{i,j=1}^{L} \left(\sum_{k,l=-1}^{P+1} \alpha_{kl} B_k(u_i) \otimes B_l(v_j) - g(u_i, v_j) \right)^2$$

or

$$\min_{\alpha} \| T\alpha - y \|^2$$

where y is the vector of grey tones $g(u_i, v_j)$ and each column of the $K \times N$ matrix T is one of the basis functions $B_k(x) \otimes B_l(y)$ evaluated at all the pixel coordinates (u_i, v_j).

VI. RESULTS

Fig. 1 shows the original of an aerial image, an RADC picture. Fig. 2 [Fig. 2(b) is a blowup of a portion of Fig. 2(a)] shows the result of applying an unconstrained DCT to the RADC image where 16 coefficients are retained. Fig. 3(a) and 3(b) are analogous to Fig. 2(a) and (b), except 25 coefficients are retained. Both Figs. 2 and 3 have the same compression ratio, 30:1. Comparison of Figs. 2 and 3 shows that transmitting more coefficients with less accuracy on each is better than transmitting fewer coefficients with more accuracy on each. This can be explained as follows. The image in a block is given exactly by

$$\sum_{i=1}^{K} \alpha_i f_i$$

where $f_1, \cdots, f_K$ are basis vectors. For the DCT case, the latter f_i represent higher frequencies, and thus, typically, the α_i decrease in magnitude as i increases. The image is approxi-

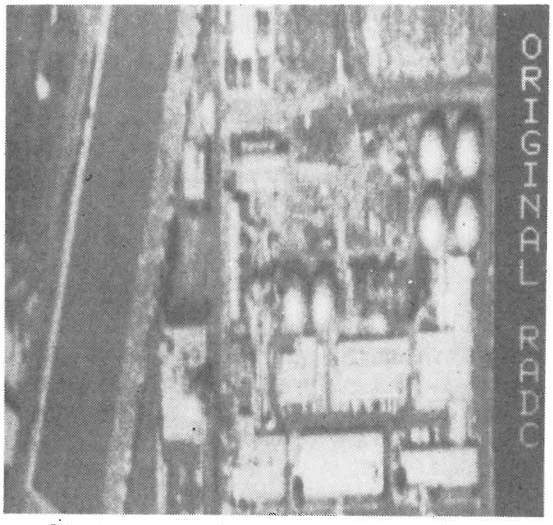
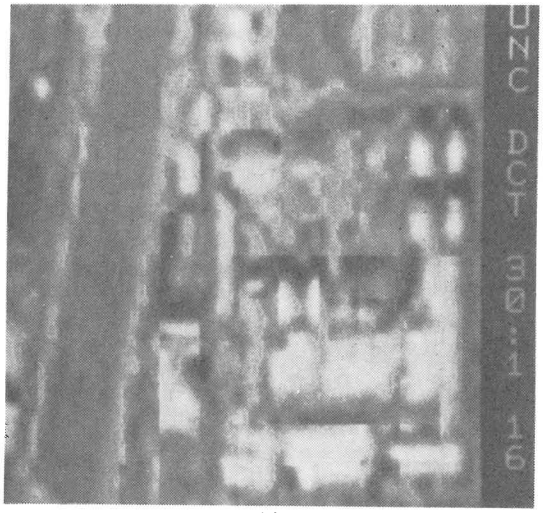
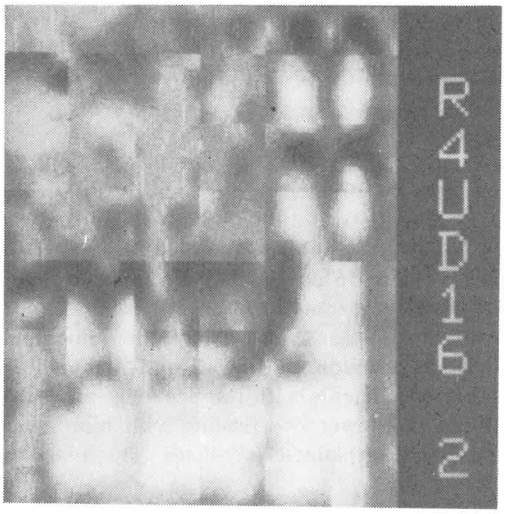


Fig. 1. Original "RADC" picture, 8 bits/pixel.

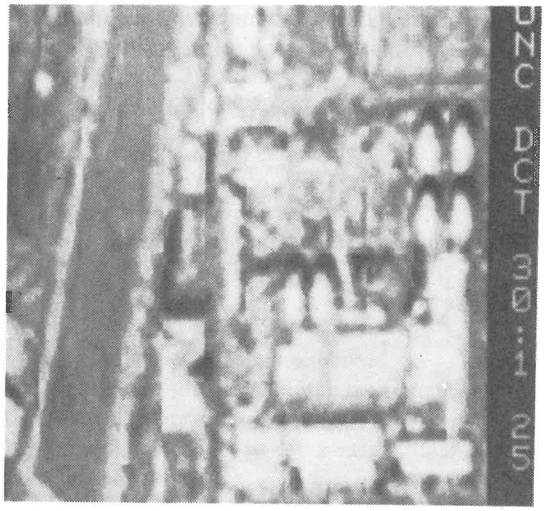


(a)

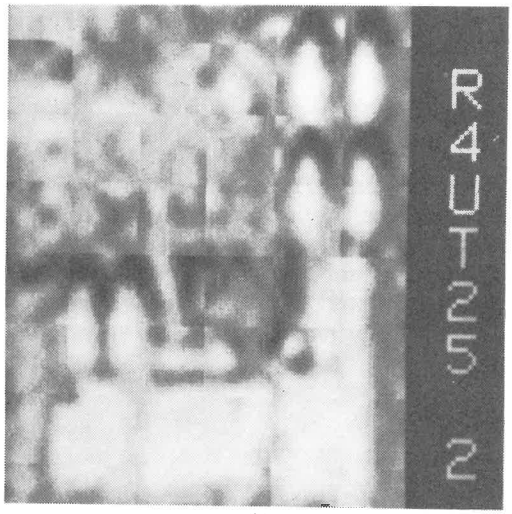


(b)

Fig. 2. (a) Reconstructed picture obtained by retaining 16 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel. (b) Reconstructed picture obtained by retaining 16 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel.



(a)



(b)

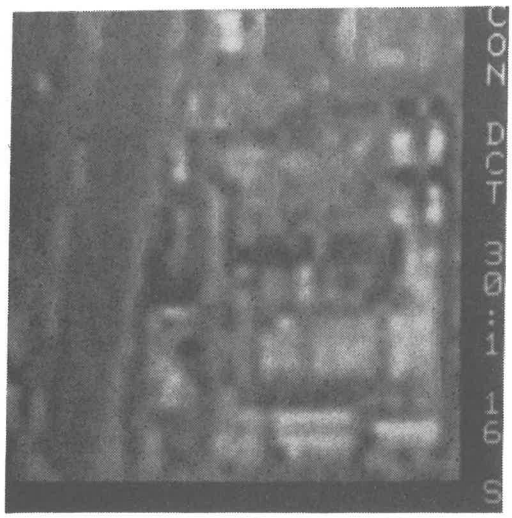
Fig. 3. (a) Reconstructed picture obtained by retaining 25 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel. (b) Reconstructed picture obtained by retaining 25 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel.

mated by

$$\sum_{i=1}^{N} \alpha_i f_i$$

and the error is roughly the order of magnitude of the first omitted term, $\alpha_{N+1} f_{N+1}$. If N is too small, α_{N+1} is larger than the quantization errors in $\alpha_1, \cdots, \alpha_N$, and their high accuracy is wasted. On the other hand, if N is too large, the quantization errors in the first few α_i overwhelm the last few α_i transmitted, and these latter coefficients are wasted. An interesting problem is to determine the optimal N for a given compression ratio.

Fig. 4(a) and (b) [Fig. 4(b) is a blowup of Fig. 4(a)] shows the constrained DCT applied to the RADC image, where 16 coefficients were retained and the compression ratio is 30:1. Comparing Fig. 2(b) to Fig. 4(b), note that the blocking in Fig. 2(b) is much worse. Fig. 4(b) is fuzzier, but since there



(a)

(b)

Fig. 4. (a) Reconstructed picture obtained by retaining 16 coefficients in the original space (nine coefficients in kernel space) of a constrained discrete cosine transformation. Equal interval quantization in the kernel space. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel. (b) Reconstructed picture obtained by retaining 16 coefficients in the original space (nine coefficients in kernel space) of a constrained discrete cosine transformation. Equal interval quantization in the kernel space. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel.

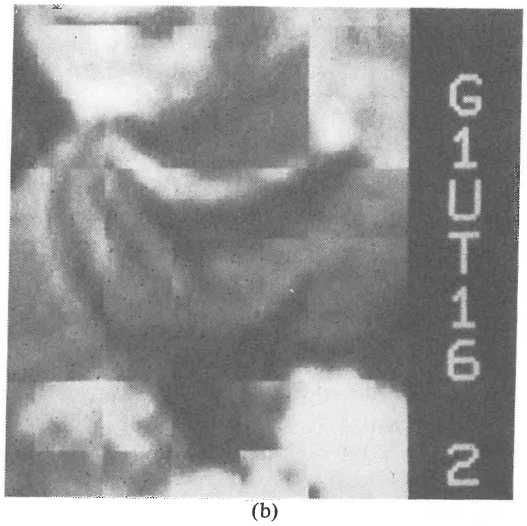
are obviously gross block boundary mismatches in Fig. 2(b), its sharpness is specious. The effect of the constrained transform coding algorithm is to remove the block boundary mismatches, at the expense of slightly defocusing the image within each block.

Figs. 5, 6, and 7 also illustrate the effect of constrained coding. Fig. 5 is the original of a girl picture. Fig. 6(a) and (b) [Fig. 6(b) is a blowup of 6(a)] shows the unconstrained DCT applied to the girl image, where 16 coefficients are retained and the compression ratio is 30:1. Fig. 7(a) and (b) [Fig. 7(b) is a blowup of 7(a)] is the constrained analog of Fig. 6(a) and (b). The blocking in Fig. 6 is particularly objectionable, whereas Fig. 7, although more blurred, has a much better overall quality.

Compared to the DCT, the performance of the cubic splines was disappointing. There were two noticeable differences between the images reconstructed from cosines and

Fig. 5. Original "girl" picture, 8 bits/pixel.

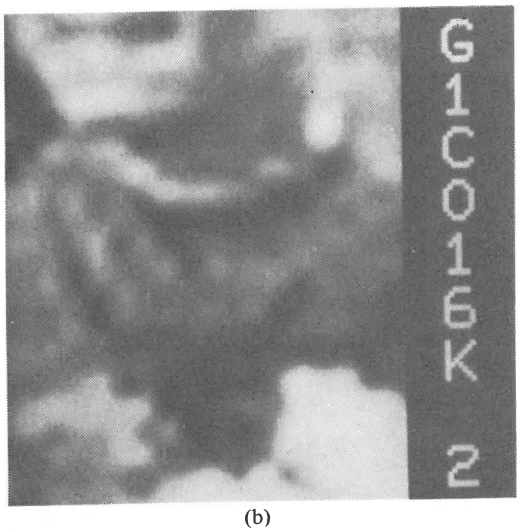
(a)



(b)

Fig. 6. (a) Reconstructed picture obtained by retaining 16 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel. (b) Reconstructed picture obtained by retaining 16 coefficients of an unconstrained discrete cosine transformation. Equal interval quantization. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel.

(a)



(b)

Fig. 7. (a) Reconstructed picture obtained by retaining 16 coefficients in the original space (nine coefficients in kernel space) of a constrained discrete cosine transformation. Equal interval quantization in the kernel space. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel. (b) Reconstructed picture obtained by retaining 16 coefficients in the original space (nine coefficients in kernel space) of a constrained discrete cosine transformation. Equal interval quantization in the kernel space. Hufman encoding. Compression ratio = 30:1, 0.27 bits/pixel.

cubic splines. For a given compression ratio, the cosine images are better. Also, the equal interval quantization had a more pronounced effect on the spline images than the cosine images. This is illustrated by Figs. 8 and 9. Fig. 8 is a constrained spline image based on 36 coefficients. Fig. 9 is the result of quantizing those coefficients used to construct Fig. 8. Note that some blocking has been introduced by the quantization. However, there are many ways to set up a spline approximation, and the particular scheme described in Section V (combined with equal interval quantization) may just be a poor choice. Another possible explanation may lie in the fact that for a Toeplitz image covariance matrix, the DCT is a good asymptotic approximation to the Karhunen-Loeve transform which is optimum in the two-norm.

Fig. 10 shows the unconstrained spline transformation applied to the girl image, using 36 coefficients with a compression ratio of 15:1. Fig. 11 is the counterpart to Fig. 10 for the constrained spline transformation. The effectiveness of the con-

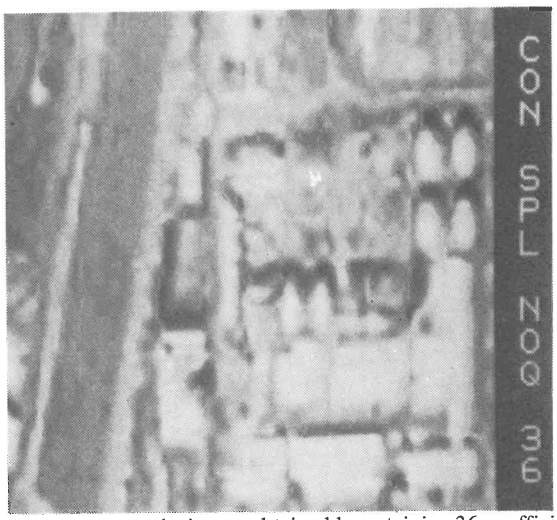


Fig. 8. Reconstructed picture obtained by retaining 36 coefficients in the original space (25 coefficients in kernel space) of a constrained splines transformation. No quantization.

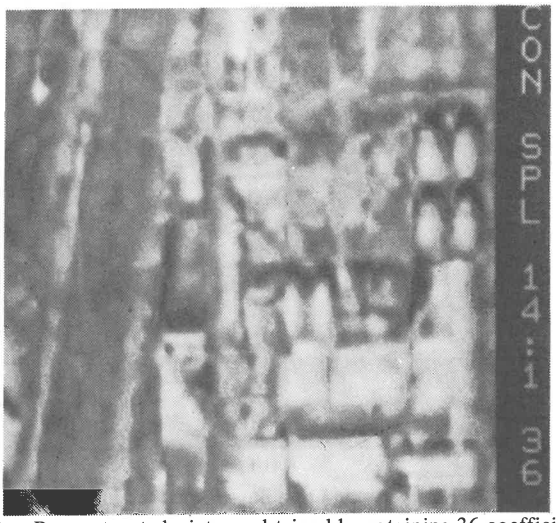


Fig. 9. Reconstructed picture obtained by retaining 36 coefficients in the original space (25 coefficients in kernel space) of a constrained splines transformation. Equal interval quantization in the kernel space. Hufman encoding, compression ratio = 14:1, 0.57 bits/pixel.

Fig. 10. Reconstructed picture obtained by retaining 36 coefficients of an unconstrained splines transformation. Equal interval quantization quantization. Hufman encoding. Compression ratio = 15:1, 0.53 bits/pixel.

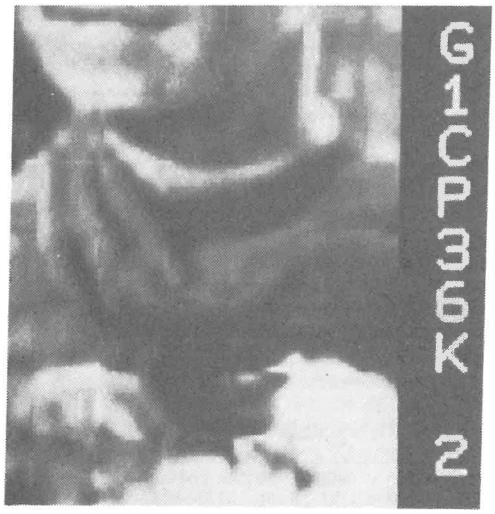


Fig. 11. Reconstructed picture obtained by retaining 36 coefficients in the original space (25 coefficients in kernel space) of a constrained splines transformation. Equal interval quantization in the kernel space. Hufman encoding, compression ratio = 15:1, 0.53 bits/pixel.

strained algorithm in alleviating the block boundary mismatches is not limited to the cosine basis functions (DCT), as Figs. 10 and 11 show.

Several issues remain to be resolved. From a function approximation theoretic viewpoint, some bases are better than others, e.g., cubic splines and Chebyshev expansions are usually better than cosines. Whether there is something special about image data which makes the cosine basis especially good, or whether other bases have not been systematically investigated, should be explored. Constrained transform coding is clearly superior to unconstrained transform coding with respect to block boundary errors and storage (everything can be reconstructed from the kernel coefficients, and there are fewer of them than transform coefficients). The effect of different kinds of quantization on the reconstructed constrained images is not well understood, and seeking an optimal quantization scheme for use with constrained transform coding appears worthwhile.

APPENDIX
LINEAR ALGEBRA

The facts reviewed here are standard in numerical analysis and approximation theory, and proofs can be found in the textbooks [18], [7], or [9], the latter being the most advanced.

Let E denote the real numbers, E^n be n-dimensional Euclidean space (all n-tuples of real numbers), L be a finite dimensional real vector space, and M a subspace of L. An *inner product* (also known as a *positive definite hermitian form*) on L is a function which assigns to every pair of vectors $x, y \in L$ a real number $\langle x, y\rangle$ such that

1) $\langle x, y\rangle = \langle y, x\rangle$ for all $x, y \in L$;
2) $\langle \alpha x, y\rangle = \alpha\langle x, y\rangle$ for all $x, y \in L$, $\alpha \in E$;
3) $\langle x + w, y\rangle = \langle x, y\rangle + \langle w, y\rangle$ for all $x, y, w \in L$;
4) $\langle x, x\rangle > 0$ for all $x \neq 0$ in L.

For $L = E^n$, the standard inner product is $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i$.

An inner product $\langle x, y\rangle$ always leads to a norm on L defined by

$$\| x \| = \sqrt{\langle x, x\rangle}.$$

A norm has the properties

1) $\| x \| \geqslant 0$ with equality if and only if $x = 0$;
2) $\| \alpha x \| = |\alpha| \, \| x \|$ for all $x \in L$, $\alpha \in E$;
3) $\| x + y \| \leqslant \| x \| + \| y \|$ for all $x, y \in L$.

Not every norm arises from some inner product, but those that do have nicer properties.

Vectors $u, v \in L$ are *orthogonal* if $\langle u, v\rangle = 0$. A set of vectors $\{u_1, \cdots, u_k\}$ is *orthonormal* if

$$\langle u_i, u_j\rangle = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$$

A vector $v \in L$ is orthogonal to the subspace M if $\langle v, x\rangle = 0$ for all $x \in M$. The orthogonal complement of M, denoted by $M^{\perp}$, is the set of all vectors orthogonal to M:

$$M^{\perp} = \{ y \in L \mid \langle y, w\rangle = 0 \text{ for all } w \in M \}.$$

Proposition 1: Every vector $x \in L$ has a unique representation of the form

$$x = u + v, \quad u \in M, \quad v \in M^{\perp}.$$

This is sometimes expressed by saying L is the direct sum of M and $M^{\perp}$, denoted $L = M \oplus M^{\perp}$. Note that u and v are unique, and $\langle u, v\rangle = 0$. u is called the *orthogonal projection* of x onto M. An elementary proof of Proposition 1 is by constructing orthogonal bases for $M^{\perp}$ and M, but an elegant proof which does not depend of the existence of orthogonal bases is known [9].

Proposition 2: Let $x = u + v \in L$, $u \in M$, $v \in M^{\perp}$. Then the map $x \to u$ defines an operator P which is linear

$$P(\alpha y + \beta z) = \alpha P y + \beta P z,$$

symmetric

$$\langle Py, z\rangle = \langle y, Pz\rangle,$$

and idempotent

$$PPy = Py.$$

Conversely, any linear, symmetric, idempotent operator P on L is a projection onto $M = \text{range } P$.

When $L = E^n$, a matrix P is a projection operator if and only if $P^t = P$ (symmetry) and $PP = P$ (idempotent). Projection operators are intimately related to least squares problems, as shown by the following.

Proposition 3: Let P be the projection operator onto the subspace M, and $f \in L$. Let $f = u + v$, $u \in M$, $v \in M^{\perp}$. Then there is a unique closest point in M to f, namely $u = Pf$, and the distance from f to M is $\| v \|$. In other words, the approximation problem

$$\min_{y \in M} \| y - f \|$$

has the unique solution $u = Pf$ and the minimum is $\| v \| = \| (I - P) f \|$.

For $L = E^n$ and $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i$, projection operators have an explicit representation.

Proposition 4. Let $L = E^n$, $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$, and B be a matrix whose columns are a basis for a subspace M. Then the projection operator P onto M is given by

$$P = B(B^tB)^{-1}B^t.$$

This explicit representation is convenient for theoretical purposes, but serious roundoff error due to the ill conditioning of B^tB makes it computationally impractical. However, if the columns of B are orthonormal, then $B^tB = I$ is perfectly conditioned and there are no numerical difficulties. The existence of orthonormal bases is shown by the following.

Proposition 5: Let $u_1, \cdots, u_n$ be independent vectors in L. Then there exist orthonormal vectors $\phi_1, \cdots, \phi_n$ such that the subspace spanned by $u_1, \cdots, u_k$ is equal to the subspace spanned by $\phi_1, \cdots, \phi_k$ for each $k = 1, \cdots, n$.

This is usually proved by constructing the ϕ_i with the Gram–Schmidt process. The Gram–Schmidt process will not be elaborated on, because it is numerically unstable, and there is a numerically stable construction of the ϕ_i (when $L = E^n$) based on the QR factorization.

Besides maintaining numerical stability, orthonormal bases make the calculation of projections trivial, as shown by the following.

Proposition 6: Let $L = E^n$, $\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i$, $x \in L$, $\{\phi_1, \cdots, \phi_k\}$ be an orthonormal basis for M, and B the matrix with columns $\phi_1, \cdots, \phi_k$. Then

$$u = Px = (BB^t)x = B(B^tx) = \sum_{i=1}^{k} \langle x, \phi_i \rangle \phi_i$$

is the projection of x onto M, where $P = BB^t$ is the projection operator onto M.

Note that the projection Px is completely specified by its Fourier coefficients $\langle x, \phi_i \rangle$, and it is these which are actually transmitted (since usually $k \ll n$).

An $n \times n$ matrix Q is orthogonal if $Q^tQ = I$, where I is the identity matrix. Note that Q is orthogonal if and only if Q^t is orthogonal, and $Q^{-1} = Q^t$. Orthogonal matrices are extremely important in matrix calculations, because multiplication by orthogonal matrices does not magnify roundoff errors.

Proposition 7: If Q is an $n \times n$ orthogonal matrix, then

$$\| Qx \| = \| x \| \quad \text{for all } x \in E^n$$

where

$$\| y \|^2 = \sum_{i=1}^{n} y_i^2.$$

The QR and SVD matrix factorizations used in Section III are based on orthogonal matrices, and thus because of Proposition 7 are very stable numerically.

REFERENCES

[1] N. Ahmen, T. Natarajan, and K. R. Rao, "On image processing and a discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, 1974.

[2] G. B. Anderson and T. S. Huang, "Piecewise Fourier transformation for picture bandwidth compression," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 133–140, 1971.

[3] ——, "Piecewise Fourier transformation for picture bandwidth compression," *IEEE Trans. Commun.*, vol. COM-20, pp. 448–491, 1972.

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

[5] H. C. Andrews and C. L. Patterson III, "Singular value decompositions and digital image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 26–53, 1976.

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

[7] G. Dahlquist, A. Bjorck, and N. Anderson, *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[8] C. de Boor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.

[9] R. Godement, *Algebra*. Paris, France: Hermann, 1968.

[10] A. Habibi, "Comparison of Nth order DPCM encoder with linear transformation and block quantization techniques," *IEEE Trans. Commun. Technol.*, vol. COM-16, pp. 948–957, 1971.

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

[12] A. Habibi, "Hybrid coding of pictorial data," *IEEE Trans. Commun.*, vol. COM-22, pp. 614–624, 1974.

[13] ——, "Survey of adaptive image coding techniques," *IEEE Trans. Commun.*, vol. COM-25, pp. 1275–1284, 1977.

[14] R. M. Haralick and K. Shanmugan, "Comparative study of a discrete linear basis for image data compression," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-4, pp. 16–27, 1974.

[15] R. M. Haralick and O. Zuniga, "Adaptive image data compression," presented at the IEEE Conf. Pattern Recognition, Image Processing, Chicago, IL, Aug. 1979.

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

[17] K. R. Rao, M. A. Narasimhan, and K. Revuluri, "Image data processing by Hadamard–Haar transforms," *IEEE Trans. Comput.*, vol. C-23, pp. 888–896, 1974.

[18] G. Strang, *Linear Algebra and its Applications*. New York: Academic, 1976.

[19] M. Tasto and P. Wintz, "Image coding of adaptive block quantizations," *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 957–972, 1971.

[20] J. H. Wilkinson, and C. Reinsch, *Handbook for Automatic Computation, Vol. II: Linear Algebra*. Berlin, Germany: Springer-Verlag, 1971.

[21] P. A. Wintz, "Transform picture coding," *Proc. IEEE*, vol. 60, pp. 809–820, 1972.

[22] W. Zschunke, "DPCM picture coding with adaptive prediction," *IEEE Trans. Commun.*, vol. COM-25, pp. 1295–1302, 1977.