COMPUTER PROGRAM FOR FAST KARHUNEN LOEVE TRANSFORM ALGORITHM

FINAL REPORT

Contract No. NAS8-31434

by

ANIL K. JAIN

DEPARTMENT OF ELECTRICAL ENGINEERING
STATE UNIVERSITY OF NEW YORK AT BUFFALO
AMHERST, NEW YORK 14260

FEBRUARY 1976

Prepared for
NASA Marshall Space Flight Center
Huntsville, Alabama

(NASA-CR-144281) COMPUTER PROGRAM FOR FAST KARHUNEN LOEVE TRANSFORM ALGORITHM Final Report (State Univ. of New York, Buffalo.) 98 p HC $5.00 CSCL 09B
N76-22948
Unclas
G3/61 26857

ERRATA

Please note the following corrections as you read this Report:

1. Page 11 - Line after eqn. (15) should read

   Equation (15) is a transcendental equation giving non-harmonic sinewaves...

2. Page 15 - Fourth line below the equation $x^o \overset{\Delta}{=} x - x_b$, replace estended by extended.

3. Page 21 - First line below Eqn. (42) should read

   and the elements $\hat{U}^o_{ij}$ are also uncorrelated ...

4. Page 21 - Eqn. (42) should be

   $$E[\hat{\nu}_{ij}\ \hat{\nu}_{k\ell}] = \beta_1^2\beta_2^2\ \lambda_{ij}\delta_{ik}\delta_{j\ell} \qquad (42)$$

5. Page 23 - Fifth line, second paragraph should read

   may not . . . . . subtract the image mean before . . .

6. Page 24 - Second last setence should read

   Thus, once again we see that even though the covariance function of U_{ij} is nonseparable, the KL transform of "U_{ij} given boundary conditions B" is a fast transform, i.e., the sine transform.

7. Page 26 - Fourth line from bottom should read

   Let n_i = ... to the i^{th} component and . . .

8. Page 28 - Ninth line from top should read

   The boundary . . . . $\{u_{i,o}\cdots\cdots, \underline{u_{N+1,j}}, \cdots\cdots\}$

9. Page 28 - The equation for $\hat{B}^*_{ij}$ near the bottom of the page; the first line of this equation α_2 is missing as marked

   $$B^*_{ij} = \sqrt{\frac{2}{N+1}}\ \{[\alpha_1 \cdot\cdot\cdot\cdot\cdot)\ \lambda_2(j) + \alpha_2 \sin\frac{j\pi}{N+1} \cdot\cdot\cdot\cdot\cdot$$

10. Page 30 - Fourth line from bottom should read

    scheme is . . . . with $\underline{\rho \le 0.7}$, this was . . . .

11. Page 23 - Line 15 . . . . bock - → block

COMPUTER PROGRAM FOR FAST KARHUNEN LOEVE TRANSFORM ALGORITHM

FINAL REPORT

Contract No. NAS8-31434

ANIL K. JAIN, PROJECT DIRECTOR

S.H. WANG, RESEARCH ASSISTANT

DEPARTMENT OF ELECTRICAL ENGINEERING
STATE UNIVERSITY OF NEW YORK AT BUFFALO
AMHERST, NEW YORK 14260

FEBRUARY 1976

NASA TECHNICAL MONITORS

Kenneth Kadramas, NASA Marshall Space Flight Center
Robert R. Jayroe, NASA Marshall Space Flight Center
Edgar M. Van Vleck, NASA Ames Research Center

Prepared for
NASA Marshall Space Flight Center
Huntsville, Alabama

Glossary of Symbols

$\triangleq$	is defined as
$\otimes$	Kronecker Product
FFT	Fast Fourier Transform
DCT	Discrete Cosine Transform
KL	Karhunen-Loeve
KLT	Karhunen-Loeve Transform
FKL	Fast Karhunen-Loeve
m.s.e.	Mean Square Error
w.r.t.	with respect to

TABLE OF CONTENTS

ABSTRACT

The purpose of this study was to apply the fast KL transform algorithm for data compression of a set of four ERTS multispectral images and compare its performance with other techniques studied by TRW, Inc. on the same image data in contract NAS2-8394. The performance criteria used here are mean square error and signal to noise ratio as used in the TRW effort. The results obtained here show a superior performance of the fast KL transform coding algorithm on the data set used and with respect to the above stated performance criteria. A summary of the results is given in Chapter I and details of comparisons and discussion on conclusions are given in Chapter IV.

CHAPTER I

INTRODUCTION

1. Image Transforms

Two dimensional image transforms have attracted considerable attention during the past decade for their application in image data compression (commonly called 'transform coding'). If U represents anN x N image matrix, then, in the context of image transform coding, W is its two dimensional transform where

$$W = TUT' \tag{1}$$

where prime denotes the transpose and T is a unitary two dimensional N x N matrix. The restriction of T to unitary matrices ensures conservation of image energy in the transform domain.

Examples of image transforms are Fourier, Hadamard, Karhunen Loeve (KL), Haar, Slant, Cosine, etc. [1-8] Typically, the transformed image W is such that most of the image energy is concentrated in relatively few samples in the W-space (usually the lower 'frequency' samples) so that these few samples alone are considered important for any subsequent image processing thereby achieving some data compression.

2. Dimensionality vs Optimality

Two considerations which become important in selecting the image transform for data compression are a) Dimensionality and b) Optimality for Compression. By dimensionality we refer to the computational effort required in implementing Eqn. (1). For an arbitrary unitary transform T, it is easy to see that computation of W takes $2N^3$ multiplications and about as many additions. For images the value of N gets large enough ($N \simeq 200$ to 1000) to make this computational load unbearable. Consequently, the evil forces of dimensionality dictate that the choice of T be restircted to certain class of transforms which can yield fast computational algorithms. One class of such transforms called Good Transforms (after I. J. Good[18]) have the property that the matrix T can be written as a product of several sparse matrices in the form

$$T = T_1 T_2 \ldots T_p \tag{2}$$

where T_i, $i = 1,\ldots,p$ ($p<<N$) are matrices with just a few non-zero entries ($\leq r$ entries per row, say, with $r << N$). Thus the multiplication of T with, say, an N x 1 vector, is accomplished in about rpN computations. Thus, if $N \approx 2^p$, then typically, the computations required in (1) reduce to approximately $rN^2 \log_2 N$. Depending on the actual transform, one computation can be defined as one multiplication and one addition/subtraction, e.g., Fourier, Cosine transforms or as one addition/subtraction, e.g., Walsh-Hadamard transform.

By optimality we mean the efficiency of a transform in achieving data compression (or bandwidth compression). Usually this optimality is measured for a class of images rather than for a single image because, conceivably a transform could be optimal for one single image and be very poor for others. This raises the question, "How do we define a class of images?" Here we consider classification of images by their statistical properties as opposed to their linguistic or descriptive properties, although many times it may be possible to quantitatively model linquistic/descriptive images by a statistical expression. For a given class of images having certain second order statistics, the Karhunen Loeve Transform* is shown to be optimal [9-13] in a mean square sense.

Although the KL transform is optimal, it has dimensionality difficulties. First, the KL transform is unique for a class of images. Therefore, it has to be computed for that class. Second, even if a closed form analytic expression for the KL transform is known, the transformation calculations of Eqn. (1) do not, in general, have a fast algorithm available. Therefore, the size of image

* Sometimes also called Hotelling Transform after Hotelling.[7,10]

that can be used for KL transform application is quite small ($\approx$ 8 x 8 or less). An example of utilizing the KL transform technique is by dividing an image into small blocks and then coding each block. Otherwise, most transform coding efforts have resorted to the class of fast image transforms described above.

3. Discrete Cosine, KL, and Fast KL Transforms

Recent experimental results of Ahmed, Natarajan and Rao [6] have shown the behavior of the Discrete Cosine Transform (DCT) to be close to the KL transform for one dimensioanl first-order markov random porcesses whose correlation parameter ρ has a value around 0.9. Actual implementation of the DCT for 2-dimensional image coding has shown that the DCT does perform better than Fourier, Walsh-Hadamard transforms, etc., at low bit rates (bit rates of less than equal to 1 bit/pixel average). However, at higher bit rates ($\simeq$ 2 bits per pixel average), a hybrid coding algorithm employing the DCT in one dimension and DPCM in the other seems to perform better than the 2-dimensional DCT [20]. This, of course, should not happen if the DCT were truly a good approximation to the KL transform, because the rate distortion curve of the KL transform is expected to give a lower bit rate at a given distortion level. This ambiguity regarding the 2-dimensional DCT may be partly be attributed to the fact that image statistics is not gaussian, and partly that the image covariance function is not separable.

4. Summary of Results

In this study, the fast KL transform algorithm developed in the next two chapters is applied to a set of four channel ERTS multispectral images and its performance is compared with other techniques studied by TRW, Inc. [20] in contract NAS2-8394. The image data used in this study is identical with that used by TRW, Inc. and in fact was supplied to us by them. Figure 1 summarizes the results of this study. Here each curve represents average of mean square error over the four channels vs. bit rate obtained by a given technique. Thus if

$$\epsilon_k^2 = \text{mean square error in the } k^{th} \text{ channel image, } k=1,\ldots 4 \qquad (2)$$

then

$$\text{average mean square error} = \tfrac{1}{4} \sum_{k=1}^{4} \epsilon_k^2 \qquad (3)$$

The KL-Cosine/DPCM, KL-Hadamard/DPCM curves, correspond to three dimensional data compression schemes utilizing a 4x4 KL transform along the temporal axis. All curves in Fig. 1 except the 2-dimensional fast KL curves were obtained in the TRW study [20]. The fast KL curves correspond to two different sets of experiments; viz, the first set (solid line) using 255 x 255 size transformation and the second set (dotted lines) using 15 x 15 block transformations on the image. In the other curves 16 x 16 spatial block size was used in performing any transform coding. Comparisons show a superior performance of the fast KL transform coding algorithm over other methods.

A detailed discussion of these and other comparisons with respect to signal to noise ratio etc., as well as conclusions of the study are given in Chapter IV.

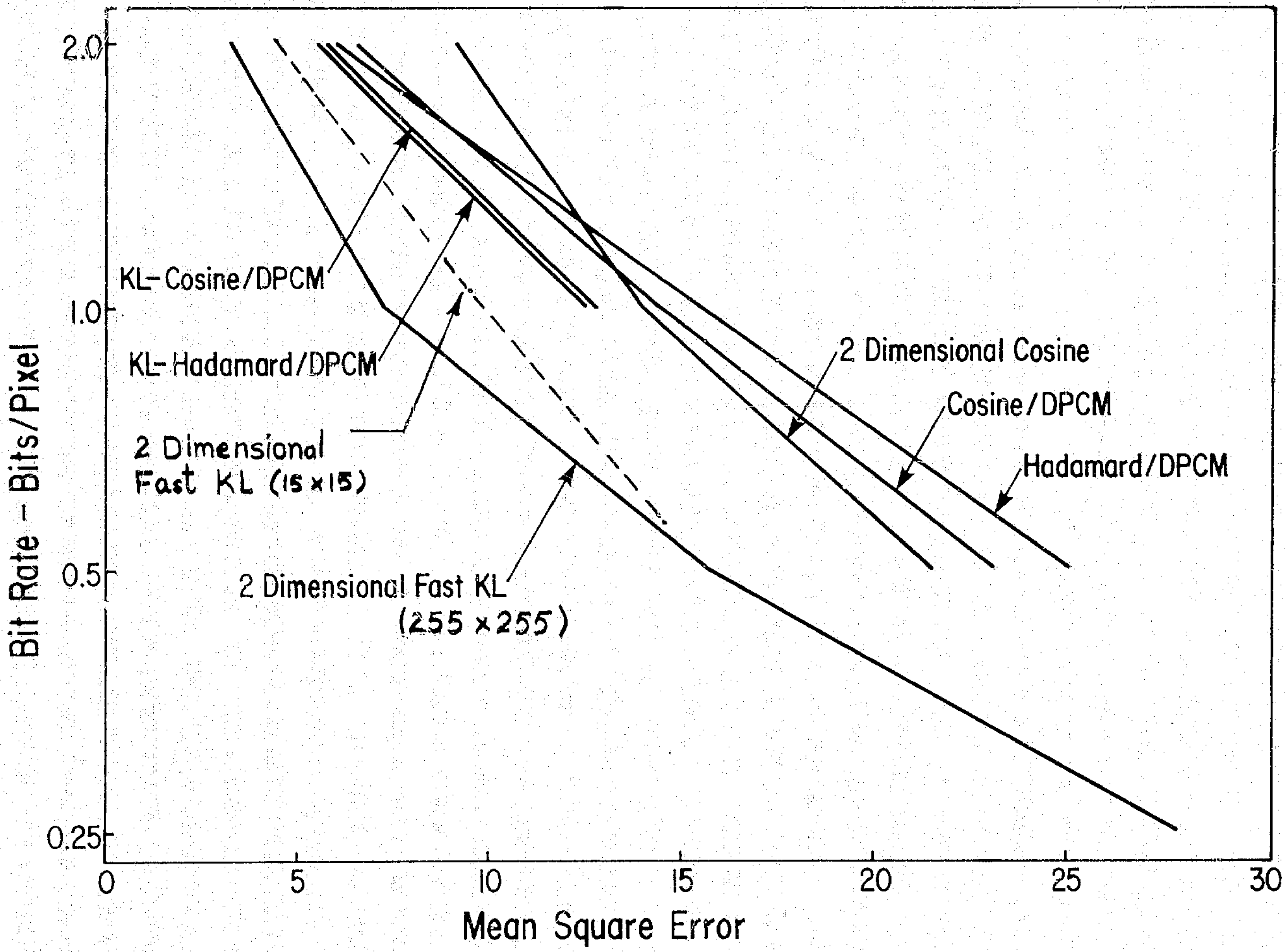


FIGURE 1 Comparison of mean square error for 2 Dimensional Fast KL vs. Other Methods

CHAPTER II

A THEORY OF FAST KARHUNEN LOEVE TRANSFORM

1. Image Covariance Model

An image may be thought of as a sample function of a two dimensional random process. If u_{ij} denotes the brightness at the spatial coordinate (i,j) then u_{ij} is considered as a random process. Consider one such process which has zero mean unit variance and is described by a covariance function given by

$$R(n,m) = E[u_{i+n,j+m}u_{ij}] = \rho_1^{|n|}\rho_2^{|m|} \tag{4}$$

where

$$1 > \rho_i = \text{correlation parameter, } i = 1, 2. \tag{5}$$

The covariance defined above assumes essentially two properties viz., stationarity and separability. Although one might argue for an isotropic non-separable covariance function of the form

$$R(n,m) = \sigma^2\rho^{\sqrt{n^2+m^2}} \tag{6}$$

as being more appropriate, the covariance of Eqn. (4) has been found to be a reasonable assumption and has been applied successfully in many different image processing problems.[5,18,20] The primary reason for preferring Eqn. (4) over Eqn. (6) is due to the amenability of covariance in (4) to simpler analysis and applicability of many one dimensional results.

2. One Dimensional Representation

Let $\{x_i\}$ be a finite one dimensional random process with zero mean and unit variance and a covariance function given by

$$E[x_i x_{i+n}] = \rho^{|n|} \qquad i = 0,1,\ldots,N+1 \quad . \tag{7}$$

The zero mean and unity variance assumptions are non-essential but serve only to present a simplified analysis. It is well known that the sequence x_i can be represented by a first order stationary Markov process as

$$x_{i+1} = \rho x_i + \varepsilon_i \qquad i \geq 0 \tag{8}$$

with

$$E[\varepsilon_i] = 0$$

and

$$E[\varepsilon_i \varepsilon_j] = (1-\rho^2)\delta_{ij} \ , \ \delta_{ij} = \text{Kronecker delta function.}$$

It has been shown earlier [1,14] that the above Markov process for a fixed N, can also be represented by

$$x_i = \alpha(x_{i+1}+x_{i-1}) + \nu_i \quad , \qquad 1 \leq i \leq N \tag{10}$$

where $$\alpha = \frac{\rho}{1+\rho^2} \tag{11}$$

and $\{\nu_i, \ i = 1, \ldots, N\}$ is a well defined random process.

The above representation is defined for a finite process (N = fixed). The interpretation of this representation is as follows. If $\bar{x}_i$ denotes the best linear mean square estimate of x_i obtained from a linear combination of the elements of the partial sequence $\{x_j, \ j \neq i\}$, then the residuals $\{\nu_i\}$ given by

$$x_i - \bar{x}_i = \nu_i$$

are such that the variance

$$E[(x_i-\bar{x}_i)^2] \triangleq E[\nu_i^2]$$

is minimum. This will be called the minimum variance representation. In order to find $\bar{x}_i$ we start by writing

$$\bar{x}_i = \sum_{k=1}^{i} a_{ik}x_{i-k} + \sum_{k=1}^{N+1-i} b_{ik}x_{i+k} \qquad i = 1, \ldots, N$$

and find a_{ik} and b_{ik} for each i and k such that $E[(x_i-\bar{x}_i)^2]$ is minimized. When this minimization is performed [14], the coefficients a_{ik} and b_{ik} are obtained as

$$a_{i1} = \alpha = b_{i1} \qquad 1 \leq i \leq N$$

$$a_{ik} = 0 = b_{ik} \qquad k \geq 2,\ 1 \leq i \leq N$$

3. The 'ν' Process

The minimum variance property of the sequence $\{\nu_i\}$ does not guarantee its being uncorrelated. In fact, in our case, the sequence elements ν_i have nearest neighbor correlation. The correlation properties of the sequence $\{\nu_i\}$ can be expressed as

$$E[\nu_i\nu_j] = \begin{cases} \beta^2 , & i = j \\ -\alpha\beta^2 , & |i-j| = 1 , \quad \beta^2 = \dfrac{1-\rho^2}{1+\rho^2} \\ 0 , & \text{otherwise} \end{cases} \tag{12a}$$

for i,j = 1,...,N, and

$$E[\nu_i x_j] = \beta^2\delta_{ij} \tag{12b}$$

These results are derived in Appendix I.

4. The Karhunen Loeve Transform of Markovian Images

If x is a one dimensional n x 1 vector with covariance matrix R, then the KL transform of x is a matrix Φ, composed of the eigenvectors of R and is defined by the relation

$$\Phi' R \Phi = \Gamma \tag{13}$$

where Γ is a diagonal matrix of eigenvalues γ_i^2. If x if a first-order markov process with covariance given by (see Eqn. (7)), then

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{n-1} \\ \rho & 1 & \rho & \cdots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \rho^2 \\ \vdots & & & \ddots & \rho \\ \rho^{n-1} & \cdots & \rho^2 & \rho & 1 \end{bmatrix}$$

and the elements of the KL transform are given by [11] (for n=even)

$$\Phi_{ij} = a_i \sin \left[\omega_i \left(j - \frac{n+1}{2}\right) + \frac{i\pi}{2}\right] \tag{14}$$

where $\gamma_i^2 = \dfrac{1 - \rho^2}{1 - 2\rho \cos \omega_i + \rho^2}$, a_i is the normalization constant and $\{\omega_i\}$ are the positive roots of

$$\tan n\omega = - \frac{(1-\rho^2) \sin \omega}{\cos \omega - 2\rho + \rho^2 \cos \omega} \tag{15}$$

Equation (15) is a transcendental equation given non-harmonic sinewaves of Eqn. (14). The KL transform of the vector x may be written as $\hat{x} = \Phi' x$ or

$$\hat{x}_i = \sum_{j=1}^{n} \phi_{ji} x_j = \sum_{j=1}^{n} a_j x_j \sin \left[\omega_j \left(i + \frac{n+1}{2}\right) + \frac{j\pi}{2}\right] \tag{16}$$

Due to non-harmonic behavior of the sine terms, a fast algorithm (like the FFT) is unavailable in computing the series of Eqn. (16). Therefore, typically n^2 computations are required in computing $\{\hat{x}_i, i = 1,\ldots,n\}$.

From (13) and (16) and using $\Phi'\Phi = I$, it is easy to see that

$$E[\hat{x}_i\hat{x}_j] = \gamma_i^2\delta_{ij} \tag{17}$$

Since the samples $\hat{x}_i$ are uncorrelated, they are quantized independently. Other advantages of the KL transform are its minimum entropy and minimum mean square error properties which make it optimal for data compression [9, 13]. These properties assure that for a chosen ratio of compression, minimum mean square distortion will result; this in comparison with all other linear unitary transformations for the assumed class of signals.

5. <u>The Fast KL Transform for Markovian Images</u>

Consider the sequence $\{\nu_k, k = 1,\ldots,N\}$, and represent this by a vector ν, then the N x N covariance matrix C given by (12) is

$$C = E[\nu\nu'] = \beta^2 \begin{bmatrix} 1 & -\alpha & & & \\ -\alpha & 1 & -\alpha & 0 & \\ & \ddots & \ddots & \ddots & \\ & 0 & \ddots & \ddots & -\alpha \\ & & & -\alpha & 1 \end{bmatrix} \tag{18a}$$

$$\triangleq \beta^2 Q \tag{18b}$$

where Q is the N x N matrix in (18a). The matrix Q is a symmetric, tridiagonal, Toeplitz matrix.

<u>Theorem 1</u>: The KL transform of the 'ν' sequence $\{\nu_k, k = 1,\ldots N\}$ is given by

$$\psi_{ij} = \sqrt{\frac{2}{N+1}}\ \sin\frac{ij\pi}{N+1} \tag{19}$$

Proof:

The eigenvectors ψ_{ij} and the eigenvalues λ_i, of the N x N symmetric, tridiagonal, Toeplitz Q are given by [14]

$$\psi_{ij} = \sqrt{\frac{2}{N+1}}\ \sin \frac{ij\pi}{N+1} \tag{20}$$

and

$$\lambda_i = 1 - 2\alpha \cos \frac{i\pi}{N+1} \qquad \text{for} \qquad i,j = 1,\ldots,N \quad . \tag{21}$$

Clearly, the matrix C in Eqn.(18a) has its eigenvectors also given by $\{\psi_{ij}\}$ (since β^2 is a scalar constant), so that the matrix $\{\psi_{ij}\}$ is the KL transform of ν.

Theorem 2: [1] For the first order, stationary, finite, Markov sequence $\{x_i,\ i = 0, 1,\ldots,N,\ N+1\}$, if the boundary conditions x_0 and x_{N+1} are given, then the KL transform of the partial sequence $\{x_k,\ k = 1,\ldots,N\}$ given x_o, x_{N+1} is the matrix ψ with elements ψ_{ij} given by Eqn. (19).

Proof: Let the given boundary conditions be

$$x_0 = c,\ x_{N+1} = d \tag{22}$$

If x and ν are defined as N x 1 vectors of components $\{x_1,\ldots,x_N\}$ and $\{\nu_1,\ldots,\nu_N\}$ respectively, then Eqn.(10) can be written as

$$Qx = \nu + b \tag{23}$$

where Q is the N x N tridiagonal matrix in (18a) and b is an N x 1 vector containing only the information at the end points; viz.,

$$b_1 = \alpha c,\ b_N = \alpha d,\ b_k = 0,\ 2 \le k \le N-1 \tag{24}$$

Since 'c' and 'd' are now given, and ν and b are uncorrelated (see (12b))

$$x_b \triangleq \mu \triangleq E[x|c,d] = Q^{-1}b \tag{25}$$

and

$$R_b = E[(x-\mu)(x-\mu)^T|c,d] = Q^{-1}E[\nu\nu^T]Q^{-1} = \beta^2 Q^{-1} \tag{26}$$

Hence, the covariance of x given end conditions is simply-- $\beta^2 Q^{-1}$. Since β^2 is a scalar, the eigenvectors of $\beta^2 Q^{-1}$ are given by $\{\psi_{ij}\}$, defined above.

Equation (23) above can be rewritten as

$$x = Q^{-1}\nu + Q^{-1}b = x^o + \mu = x^o + x_b \tag{27}$$

where we have defined $x^o = Q^{-1}\nu$ and x_b is defined in (25). Thus, Eqn. (27) decomposes the random process x into two terms, viz., x^o and x_b where x^o has zero mean and covariance R_b given by Eqn. (26) and x_b is the conditional mean of x given boundary conditions. Denoting $\hat{x} = \psi x$ and similarly $\hat{\nu}$, $\hat{b}$, $\hat{x}^o$, etc. and realizing from theorem 1 that $\psi Q \psi = \Lambda$, Eqn. (27) can be transformed to yield

$$\hat{x}_i = \frac{\hat{\nu}_i}{\lambda_i} + \frac{\hat{b}_i}{\lambda_i} = \hat{x}^o_i + \hat{\mu}_i \tag{28}$$

The definition of b in (24) gives

$$\hat{b}_i = \sqrt{\frac{2}{N+1}}\ \alpha(c-(-1)^i d) \sin\frac{i\pi}{N+1} = \lambda_i \hat{\mu}_i \tag{29}$$

and application of the result in (26) shows $\{\hat{x}^o_i\}$ are uncorrelated, i.e.,

$$E[\hat{x}^o_i \hat{x}^o_j] = \frac{\beta^2}{\lambda_i}\,\delta_{ij} \tag{30}$$

From Eqn. (26), the eigenvectors are independent of the correlation parameter ρ and only the eigenvalue λ_i depend on the statistics of the random process x. This is in contrast with the eigenfuctions ϕ_{ij}, Eqn. (14), which depend on ρ through ω_i and γ_i^2. Moreover, the eigenvectors of Eqn. (20) are harmonic sine waves, so that a fast computational algorithm is possible and is developed in the next chapter.

The decomposition in (27) has some interesting properties. First x_b, the conditional mean of "x given boundary conditions," depends only on the boundary conditions, as given by (25). The process x_b is called the boundary response of the process x. The process x^o is obtained simply by subtracting the boundary response from the original random process x. Second, the decomposition is orthogonal, i.e.,

$$E[x^o x_b^T] = 0 \tag{31}$$

This follows from Eqn. (5) of Appendix I where

$$E[\nu_i x_{i\pm k}] = 0 \qquad k \neq 0, \qquad 1 \leq i \leq N$$

Since $x^o = Q^{-1}\nu$, and x_b contains x_0 and x_{N+1} only, $E[\nu_i x_0] = E[\nu_i x_{N+1}] = 0$.

When the boundary conditions are known (i.e., c, d are given) then x_b can be easily calculated and the vector x^o is written as a modification of the original vector x according to

$$x^o \triangleq x - x_b$$

The vector x^o can now be coded by its KL transform, which is a fast transform. The original vector x is recovered easily from x^o if the two boundary values c, d are separately coded for transmission. This idea above easily estended to two dimensions when the two dimensional image autocorrelation is separable.

The special properties of the fast KL transform are:

(i) The fast KL transform is independent of the image correlation parameter, ρ. Only the boundary response x_b depends on ρ. Hence, the fast KLT coding algorithm could be useful in adaptive coding schemes, compared to the actual KL transform which varies with ρ.

(ii) The transform domain variances are known as a closed form expression for any fixed ρ. Hence, the quantizer design calculations are simplified (since the transform domain bit assignments depend on the distibution of these variances). These calculations are specially facilitated if the quantizer is to be adaptively changed with the changes in correlation parameter.

(iii) The number of computations required in fast KI transform calculations for an N x N image is of the order $N^2\log_2 N$, the same order as in FFT, or fast DCT. Strictly speaking, the fast KL transform computations are less than the fast DCT computations.

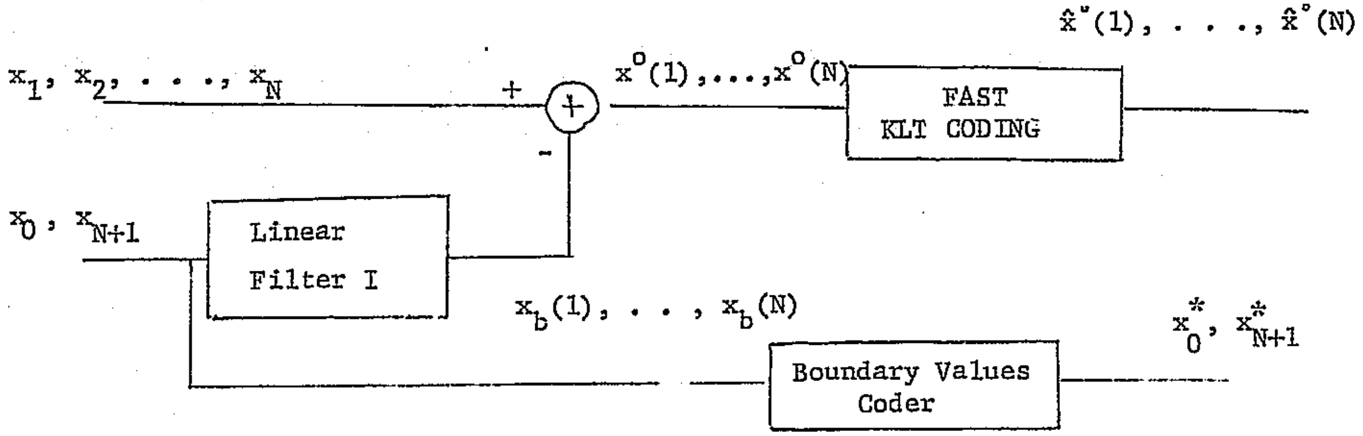
It should be noted that the fast KL transform is not a numerical algorithmic fast solution of the conventional KLT of the data, rather, it is the conventional and fast KL transform of a modification of the data. Figure 2a shows how this modification is achieved. The boundary values x_0, and x_{N+1} are passed through a linear filter to compute the boundary response $x_b(1), \ldots, x_b(N)$. Note that this entire response is generated by only two input values. The modified data is then $x^o(k)$, $1 \le k \le N$, which can be fast KL transform coded. The boundary conditions x_0, x_{N+1} are shown to be uncorrelated with $x^o(k)$, and are therefore coded independently. Figure 1b shows the reconstruction of the data from the fast KL coded data and the boundary conditions. The linear filters employed in figures 2(a) and 2(b) are known apriori and each filter requires only 2N computations for a N x 1 vector for computing its output. The filter calculations and coding of boundary values can be avoided in some practical applications by approximating the boundary values by the mean value of the data vector x and subtracting this mean value from the data vector before compressing it via the fast KL transform. Hence, for zero mean data the boundary conditions may be approximated by zero and only the mean value of the data vector is coded in addition to the modified vector x^o.

6. Two Dimensional Representation

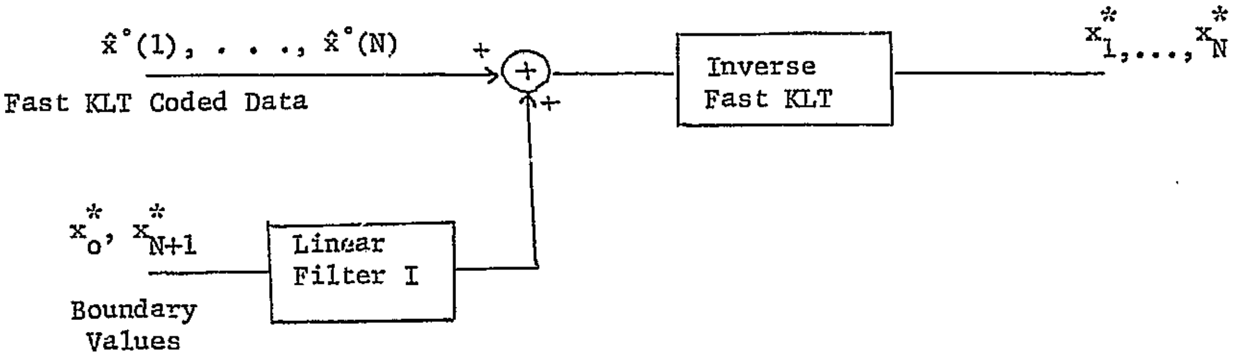
Let the image be described by a zero mean random process u_{ij} with i,j = 0,1,...,N,N+1 and autocorrelation of Eqn. (4). Following Appendix II it can be shown that the minimum variance representation of u_{ij} can be written as

$$u_{ij} - \alpha_1(u_{i+1,j}+u_{i-1,j}) = v_{ij} \qquad (32)$$

$$v_{ij} - \alpha_2(v_{i,j+1}+v_{i,j-1}) = \nu_{ij} \qquad (33)$$



a) Fast KL Transform Decomposition and Coding



b) Reconstruction of Fast KL Coded Data

FIGURE 2: The Concept of Fast KL Transform Coding

where

$$i,j = 1,2,. . ., N \text{ and } \alpha_k = \frac{\rho_k}{1+\rho_k^2}, \quad k = 1, 2.$$

If v_{ij} is eliminated from Eqns. (32) and (33), we get

$$u_{ij} = \alpha_1(u_{i+1,j} + u_{i-1,j}) + \alpha_2(u_{i,j+1} + u_{i,j-1}) - \alpha_1\alpha_2(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}) + \nu_{ij} \tag{34a}$$

$$= \bar{u}_{ij} + \nu_{ij} \tag{34b}$$

Figure (3a) shows the $\bar{u}_{ij}$ as a linear combination of the nearest eight neighbors in this case.

7. The Two Dimensional Fast KL Transform

The two dimensional result is summarized by the following theorem.

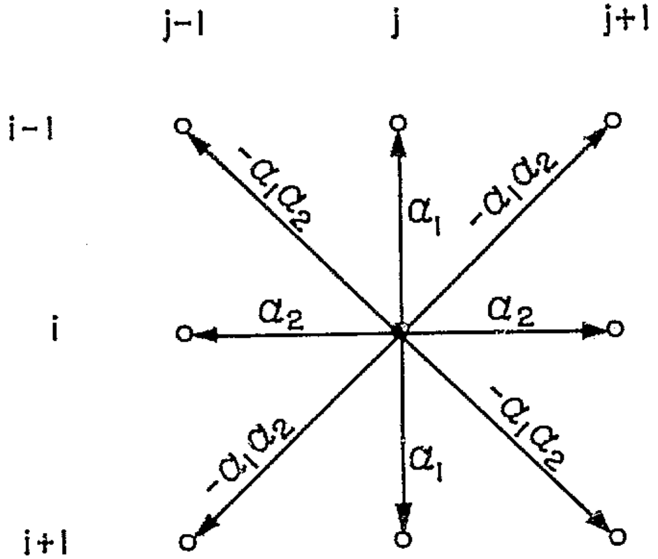
Theorem 3: If u_{ij} is a finite, two dimensional random process defined for $i,j = 0,1,\ldots,N,N+1$, with its autocorrelation given by Eqn. (4), and if the boundary conditions $\{u_{o,j}, u_{N+1,j}, u_{i,o}, u_{i,N+1}$, for all $i,j\varepsilon[0,N+1]\}$ are given, then the KL transform of the partial field $\{u_{k\ell}, k = 1,\ldots,N; \ell = 1,\ldots,N\}$ is given by

$$T = \psi \circledast \psi ,$$

where ψ is SINE transform defined by elements

$$\psi_{ij} = \sqrt{\frac{2}{N+1}} \sin \frac{ij\pi}{N+1} .$$

For proof of this theorem we resort to the two dimensional minimum variance representation of u_{ij} in (32) and (33). If U,V, and ν are N x N matrices of elements u_{ij}, v_{ij}, and ν_{ij} respectively, then



$$u_{ij} = \alpha_1(u_{i+1,j} + u_{i-1,j}) + \alpha_2(u_{i,j+1} + u_{i,j-1})$$
$$-\alpha_1\alpha_2(u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1}+u_{i-1,j-1})$$

$$\alpha_1 = \frac{\rho_1}{1 + \rho_1^2}$$

$$\alpha_2 = \frac{\rho_2}{1 + \rho_2^2}$$

FIGURE 3a: MINIMUM VARIANCE REPRESENTATION MODEL

$$
\begin{aligned}
Q_1U &= V + B_1 \\
VQ_2 &= \nu + B_2
\end{aligned}
\tag{35}
$$

where B_1 and B_2 are N x N matrices containing only the boundary information and are given by

$$
B_1 = \alpha_1 \left[\begin{array}{cccc}
u_{0,1} & u_{0,2} & - - - - \cdots & u_{0,N} \\
 & & & \\
 & & & \\
u_{N+1,1} & & & u_{N+1,N}
\end{array}\right]
\begin{array}{l}
\leftarrow \text{1st row} \\
\uparrow \\
\updownarrow \text{ N-2rows of 'zeros'} \\
\downarrow \\
\leftarrow \text{Nth row}
\end{array}
\tag{36a}
$$

$$
B_2 = \alpha_2 \begin{array}{c}
\begin{array}{cccc}
\text{1st column} & & & \text{Nth column}
\end{array} \\
\left[\begin{array}{cccc}
v_{1,0} & & & v_{1,N+1} \\
v_{2,0} & & & v_{2,N+1} \\
 & & & \\
 & & & \\
v_{N,0} & & & v_{N,N+1}
\end{array}\right] \\
\underbrace{|\longleftarrow\longrightarrow|}_{\text{N-2 columns of zeros}}
\end{array}
\tag{36b}
$$

In Eqn. (36b), the quantities $v_{i,o}$ and $v_{i,N+1}$ are obtained from the boundary information of $u_{i,o}$ and $u_{i,N+1}$, for i = 1,...,N, via Eqn. (32). Eliminating V in Eqns. (35) we get

$$
\begin{aligned}
Q_1UQ_2 &= \nu + B_1Q_2 + B_2 \\
&= \nu + B
\end{aligned}
\tag{37}
$$

where

$$
B \triangleq B_1Q_2 + B_2
$$

contains only the boundary information. If $\bar{U}$, $\bar{\nu}$, and $\bar{B}$ are now defined as $N^2 \times 1$ vectors corresponding to a lexicographic (dictionary) ordering of the matrices U, ν and B respectively, Eqn. (37) can be rewritten, after noting $Q_2=Q_2^T$, as

$$(Q_1 \circledast Q_2)\bar{U} = \bar{\nu} + \bar{B} \tag{38}$$

where $\circledast$ denotes Kronecker product. From Appendix II, the vector $\bar{\nu}$ has zero mean and $E[\bar{\nu}\bar{\nu}^T] = \beta_1^2\beta_2^2(Q_1 \circledast Q_2)$. Hence, from Theorem 1 in the last section, $(\psi \circledast \psi)$ is the $N^2 \times N^2$ KL transform matrix of $\bar{\nu}$ and

$$\bar{U}_b \triangleq \bar{\mu} \triangleq E[\bar{U}|\bar{B}] = (Q_1 \circledast Q_2)^{-1}\bar{B} \tag{39}$$

and

$$R_u \triangleq E[(\bar{U}-\bar{\mu})(\bar{U}-\bar{\mu})^T|\bar{B}] = \beta_1^2\beta_2^2(Q_1 \circledast Q_2)^{-1}, \; \beta_k^2 = \frac{1-\rho_k^2}{1+\rho_k^2}, \; k = 1,2 \tag{40}$$

Hence the matrix of eigenvectors of R_u also is $(\psi \circledast \psi)$, so that the KL transform of $\{u_{ij}$ given boundary conditions $u_{k\ell}$, $k, \ell = 0, N+1\}$ is given by the matrix ψ defined in Eqn. (20). Finally, if Eqn. (37) is pre- and post-multiplied by ψ and we define $W = \psi U \psi$, $\hat{\nu} = \psi\nu\psi$, $\hat{B} = \psi B \psi$, the following relations are easily verified, if $\lambda_1(i)$ and $\lambda_2(j)$ are the eigenvalues of Q_1, Q_2 respectively.

$$w_{ij} = \frac{\hat{\nu}_{ij}}{\lambda_{ij}} + \frac{\hat{B}_{ij}}{\lambda ij} = \hat{U}^0_{ij} + \hat{U}_b(i,j), \quad \lambda_{ij} = \lambda_1(i)\;\lambda_2(j) \tag{41}$$

$$E[\hat{\nu}_{ij}\hat{\nu}_{k\ell}] = \beta_2^4\;\beta_1^2\beta_2^2\;\lambda_{ij}\;\delta_{ik}\;\delta_{j\ell} \tag{42}$$

and the elements $\hat{U}^o_{ij}$ are aslo uncorrelated. Again, from (38) we get the orthogonal decompositon of the image matrix as

$$\tilde{U} = \tilde{U}^o + \bar{U}_b \tag{43a}$$

where $\tilde{U} = (Q_1 \circledast Q_2)^{-1}\tilde{\nu}$, $\bar{U}_b = (Q_1 \circledast Q_2)^{-1}\bar{B}$ (43b)

or in matrix form

$$U = U^o + U_b \tag{44a}$$

where $U^o = Q_1^{-1}\,\nu\,Q_2^{-1}$, $U_b = Q_1^{-1}\,B\,Q_2^{-1}$ (44b)

If we define

$$\hat{x}_{ij} = \frac{\nu_{ij}}{\sqrt{\lambda_{ij}}} \tag{45}$$

then $E[\hat{x}_{ij}] = 0$, $E[\hat{x}_{ij}\hat{x}_{k\ell}] = \beta_1^2\beta_2^2\delta_{ik}\delta_{j\ell}$. Thus $\hat{x}_{ij}$ are identically distributed variables of equal variances. If $\hat{x}_{ij}$ are quantized, then the dynamic range of each quantizers (corresponding to $\hat{x}_{ij}$) will be the same although the number of bits assigned to $\hat{x}_{ij}$ will be different.

8. Comments:

The fast KL transfrom has two steps. First, the residuals ν_{ij} are calculated and then a sine transform of the residuals is taken. Alternately, (see 41), we can first take the sine transform to obtain w_{ij} and then subtract $\hat{B}_{ij}/\lambda_{ij}$ = ($\hat{U}_b(i,j)$ = the sine transform of the boundary response) to obtain $\hat{U}^0_{ij}$. In any event, the variables $\hat{\nu}_{ij}$ (or scaled variables $\hat{x}_{ij}$) quantized and coded in the sine transform domain. In conventional sine transform coding, on the other hand, the sine transform samples w_{ij} will be coded. Further, the variances of $\hat{U}^0_{ij}$ are $\frac{\beta_1^2\beta_2^2}{\lambda_{ij}}$ which are different from the variances of U_{ij} in the sine transform domain. Thus, the quantizers in the sine transform domain and fast KL transfrom domain will be different.

Comparison with Discrete Cosine Transform (DCT)

The DCT of a sequence $\{x_m\}$ is defined as

$$g_o = \frac{\sqrt{2}}{N}\sum_{m=o}^{N-1} x_m, \quad g_k = \frac{2}{N}\sum_{m=o}^{N-1} x_m \cos\frac{(2m+1)k\pi}{2N}, \quad 1 \le k \le N-1 \tag{46}$$

Although both the fast KL and the Discrete Cosine transforms have obvious relationship with DFT, the DCT does not satisfy the KL transform Eqn. (13) for either R (the original covariance of Eqn. (4) or for R_b (the conditional autocorrelation of Eqn. (26)). The relationship between the KL transform of Eqn. (14) and the fast KL transform of Eqn. (20) is more easily seen; the eigenvectors in the latter being periodic with ω_i of (15) being replaced by $\frac{i\pi}{N+1}$.

Necessity of zero mean image

It is easy to check that the sine transform does not diagonalize a matrix of all constant elements. Thus, if the sine transform is the KL transform of a sequence $\{x_i^o, 1 \leq i \leq N\}$, it is not the KL transform of the sequence $\{x_i^o + a\}$ where 'a' is a constant. Since the given image data may not have zero mean, it is essential to subtract the image mena before applying the fast KL image coding algorithm. In blo- block coding it is advisable to make each image block zero mean before coding. The overhead of transmitting the bock mean for 15 x 15 blocks, 7 bit image is only 0.031 bits/pixel.

9. Extension to Markov Images with Nonseparable Covariances Functions

A random field described by the stochastic difference equation.

$$u_{ij} = \frac{\alpha}{2}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) + e_{ij} \tag{47}$$

where $E[e_{ij}]=0$, $E[u_{i\pm k,j\pm \ell}e_{ij}] = C_o^2\,\delta_{k,0}\delta_{\ell,0}$ (48)

$$E[e_{ij}e_{i+k,j+\ell}] = \begin{cases} -\frac{\alpha}{2}C_o^2 & k = \pm 1,\ \ell = 0 \\ -\frac{\alpha}{2}C_o^2 & \ell = \pm 1,\ k = 0,\ \alpha < \frac{1}{2} \\ 0 & \text{Otherwise} \end{cases} \tag{49}$$

generates a Markov-1 field [21] whose covariance function is nonseparable. Such fields could give a better approximation to the actual covariance function of images. If an image is represented by an N x N segment of this field, then following the notation of the previous sections, we can write

$$QU + UQ = e + B \tag{50}$$

where Q, U, B, e are N x N matrices; the matrix Q being tridiagonal as before, and B containing only the boundary information (this B matrix is obviously different from that in (37)). In Kronecker product notation, it can be seen that

$$E[\bar{e}\bar{e}^T] = c_o^2 \, (I \otimes Q + Q \otimes I) \overset{\Delta}{=} c_o^2 \, \bar{R} \tag{51a}$$

$$\bar{\mu} \overset{\Delta}{=} E[\bar{U} \mid B] = \bar{R}^{-1}\bar{B} \tag{51b}$$

$$R_u \overset{\Delta}{=} E[(\bar{U} - \bar{\mu})(\bar{U} - \bar{\mu})^T | \bar{B}] = c_o^2 \, \bar{R}^{-1} \tag{51c}$$

where $Q_{ij} = \begin{cases} \frac{1}{2} & i = j \\ -\frac{\alpha}{2} & |i - j| = 1 \\ 0 & \text{Otherwise} \end{cases}$

The KL transform of $\bar{R}$ is $\psi \otimes \psi$ and hence $\psi \otimes \psi$ is also the KL transform of R_u in (51c). Thus, once again we see that even though the covariance fucntion of U_{ij} given boundary conditions B, is a fast transform, i.e. the sine transform. Hence, images represented by random fields of (47) can be fast KL coded by calculating $\bar{\mu}$ (boundary response), subtracting it from $\bar{U}$ and coding the residual $\bar{U} - \bar{\mu}$ via the (fast) sine transform.

CHAPTER III

THE FAST KARHUNEN LOEVE TRANSFORM CODING ALGORITHM

1. Implementation on One Dimensional Image Data*

It should be pointed out that while Eqn. (20) is directly related to the FFT, the elements ψ_{ij} are not the 'imaginary' term in the conventional Discrete Fourier transform. It is easy to show that ψ_{ij} form a complete orthonormal set of basis vectors, whereas the imaginary terms of a DFT do not form a complete set of basis vectors.

Let

$$y_k \triangleq \sum_{\ell=1}^{N} \psi_{k\ell} x_\ell \qquad = \sqrt{\frac{2}{N+1}} \sum_{\ell=1}^{N} x_\ell \sin \frac{k\ell\pi}{N+1} \tag{52}$$

If the DFT is defined as

$$z_k = \text{DFT}\{z_\ell\} = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} z_\ell \, e^{i \frac{2\pi\ell k}{M}} \quad , \qquad 0 \leq k \leq M-1 \tag{53}$$

then, in order to implement (52) define a variable z_ℓ as

$$\begin{aligned} z_o &= 0 \\ z_\ell &= x_\ell \quad , \qquad \ell = 1,\ldots,N \\ z_\ell &= 0 \quad , \qquad \ell = N+1, N+2, \ldots, 2N+1 \end{aligned} \tag{54}$$

Now let $\hat{z}_k$ be given by

$$\hat{z}_k = \frac{1}{\sqrt{2(N+1)}} \sum_{\ell=0}^{2N+1} z_\ell \sin \frac{2k\ell\pi}{2N+2} = \text{Im}[\text{DFT}\{z_\ell\}] = \frac{y_k}{2} \tag{55}$$

Since $\hat{z}_k$ can be computed via FFT, y_k is obtained via this fast algorithm.

* The x,y, z variables in this section are not the same as used elsewhere. This section only shows the algorithmic implementation of the sine transform.

2. Data Compression Via the Fast Karhunen Loeve Transform

First consider the one dimensional case. From (28) we have

$$\hat{x}_i = \frac{\hat{\nu}_i}{\lambda_i} + \frac{\hat{b}_i}{\lambda_i} \overset{\Delta}{=} \frac{\hat{z}_i}{\sqrt{\lambda_i}} + \frac{\hat{b}_i}{\lambda_i} \qquad (56)$$

where, $\hat{b}_i$ is known whenever c and d are known. The next question is 'What is the bit assignment strategy for quantization?' From Eqn. (56), if $\hat{z}_i$ and $\hat{b}_i$ are observables, the original signal x can be constructed by calculating $\hat{x}_i$, where

$$\hat{z}_i = \frac{\hat{\nu}_i}{\sqrt{\lambda_i}} , \qquad E[\hat{z}_i \hat{z}_j] = \beta^2 \delta_{ij} \qquad (57)$$

Let $c*$, $d*$, $\hat{z}_i^*$ = quantized value of c,d, $\hat{z}_i$ respectively

$\hat{x}_i^*$ = reconstructed value of $\hat{x}_i$ from $\hat{z}_i^*$, $c*$, and $d*$.

The expected mean square quantization error can be written as

$$e_i \overset{\Delta}{=} E[(\hat{x}_i - \hat{x}_i^*)^2] = \frac{1}{\lambda_i^2} E[(\hat{\nu}_i - \hat{\nu}_i^*)^2 + (\frac{2}{N+1})(\sin \frac{i\pi}{N+1})^2 \alpha^2 \cdot [(c-c*) - (-1)^i (d-d*)]^2]$$

Letting $c* \cong c$ and $d* \cong d$, we get

$$e_i = \frac{1}{\lambda_i^2} E[(\hat{\nu}_i - \hat{\nu}_i^*)^2] \overset{\Delta}{=} \frac{1}{\lambda_i} E[(\hat{z}_i - \hat{z}_i^*)^2] = \frac{q_i^2}{\lambda_i}$$

If q_i^2 is small, then to minimize $\sum e_i$ the number of quantization levels for the ith component $\hat{z}_i$ should be proportional to $\frac{1}{\lambda_i}$.

Let n_i = number of bits to be assigned to the ith compenent and let

$$n_i = b_1 - b_2 \log_2 \lambda_i$$

if p' = desired bit rate in bits/pixel, then

$$p'N = \sum_{i=1}^{N} n_i = Nb_1 - b_2 \sum_{i=1}^{N} \log_2 \lambda_i$$

Now recall that $\lambda_i = 1 - 2\alpha \cos \frac{i\pi}{N+1}$,

Clearly, $0 < \lambda_1 < \lambda_2 < \ldots < \lambda_N < 2$ for all $0 < \rho < 1$,

Hence, $n_1 > n_2 > n_3 \ldots > n_N$

Let n_1 = maximum number of bits to be assigned be fixed; then

$$n_1 = b_1 - b_2 \log_2 \lambda_i$$

After solving for b_1 and b_2 we get

$$n_i = n_1 - \frac{(n_1 - p')N \log \frac{\lambda_1}{\lambda_i}}{\sum\limits_{k=1}^{N} \log \frac{\lambda_1}{\lambda_k}} \tag{58a}$$

The result of the above equation will not give integer values to n_i, in general. So a truncation to the nearest integer has to be made.

Let $n_i^* = [n_i]$ = nearest integer to n_i, then the actual bit rate is given by

$$p = \frac{1}{N} \sum_{i=1}^{N} n_i^*$$

Proceeding as in the above one dimensional case and assuming negligible quantization error in the boundary information and defining

$n_{i,j}^*$ = number of bits to be assigned to $\hat{v}_{ij}$

p' = desired bit rate in terms of bits/pixel

we get

$$n_{ij}^* = [n_{ij}] = \left[n_{11} - N^2 \frac{(n_{11} - p') \log \frac{\lambda_{ij}}{\lambda_{11}}}{\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \log \frac{\lambda_{ij}}{\lambda_{11}}} \right] \tag{58b}$$

where [ ] indicates the nearest integer. The two dimensional coding algorithm can be implemented in the following steps.

3. Fast Karhunen Loeve Transform Coding Algorithm

1. Calculate the statistics (i.e. mean, variance, horizontal and vertical correlation parameters) of the source image.

2. Create zero mean image, by subtracting the image mean (or mean of the image block in case of block by block coding) from each point in the image.

3. Use Eqns. (34a) and (34b) to compute $\bar{u}_{ij}$ and ν_{ij} for i, j = 1,...N.

4. Take the two dimensional sine transform of ν_{ij} to obtain $\hat{\nu}_{ij}$. Calculate $\hat{x}_{ij} = \frac{\nu_{ij}}{\lambda_{ij}}$.

5. Quantize $\hat{x}_{ij}$ using n^*_{ij} bits to obtain $\hat{x}^*_{ij}$. Details of quantization method are given in the next chapter. The values $\hat{x}_{ij}^*$ are used for transmission. The boundary information, i.e., $\{u_{i,o}, u_{i,N+1}, u_{o,j}, u_{N+i,j}, i, j = 0,\ldots,N+1\}$ is quantized separately. In practical situations this information may be assumed to be the mean value of the image block.

6. At receiver compute $\hat{B}^*_{ij}$ from $\{u^*_{i,o}, u^*_{i,N+1}, u^*_{o,j}, u^*_{N+1,j}\}$. If $\hat{r}_{o,j}$ is defined as the sine transform of the row $\{u^*_{o,j}, j = 1,\ldots,N\}$, i.e.

$$\hat{r}_{o,j} = \sqrt{\frac{2}{N+1}} \sum_{k=1}^{N} u^*_{o,k} \sin \frac{jk\pi}{N+1}, \qquad j = 1,\ldots,N$$

and $\hat{r}_{N+1,j}$, $\hat{c}_{i,o}$, and $\hat{c}_{i,N+1}$ are similarly defined as the sine transforms of $\{u^*_{N+1,k}\}$, $\{u^*_{k,o}\}$, and $\{u^*_{k,N+1}\}$ respectively, for i, j, k = 1,...,N then it can be seen that the boundary terms $\hat{B}^*_{ij}$ are obtained as

$$\hat{B}^*_{ij} = \sqrt{\frac{2}{N+1}} \Big\{ \alpha_1 \sin \frac{i\pi}{N+1} (\hat{r}_{oj} + (-1)^{i+1} \hat{r}_{N+1,j}) \lambda_2(j) + \sin \frac{j\pi}{N+1}$$

$$(\hat{c}_{io} + (-1)^{j+1} \hat{c}_{i,N+1}) \lambda_1(i) - \alpha_1 \alpha_2 \sqrt{\frac{2}{N+1}} \sin \frac{i\pi}{N+1} \sin \frac{j\pi}{N+1} \cdot$$

$$\cdot [u^*_{oo} + (-1)^{j+1} u^*_{o,N+1} + (-1)^{i+1} u^*_{N+1,o} + (-1)^{i+j+2} u^*_{N+1,N+1}] \Big\}$$

7. Compute $w^*_{ij} = \frac{\hat{x}^*_{ij}}{\sqrt{\lambda_{ij}}} + \frac{\hat{B}^*_{ij}}{\lambda_{ij}}$ and obtain u^*_{ij}, the reconstructed image at the receiver as

$$\{u_{ij}\} = U^* = \psi W^* \psi$$

Figure 3b shows the block diagram flow of the above algorithm.

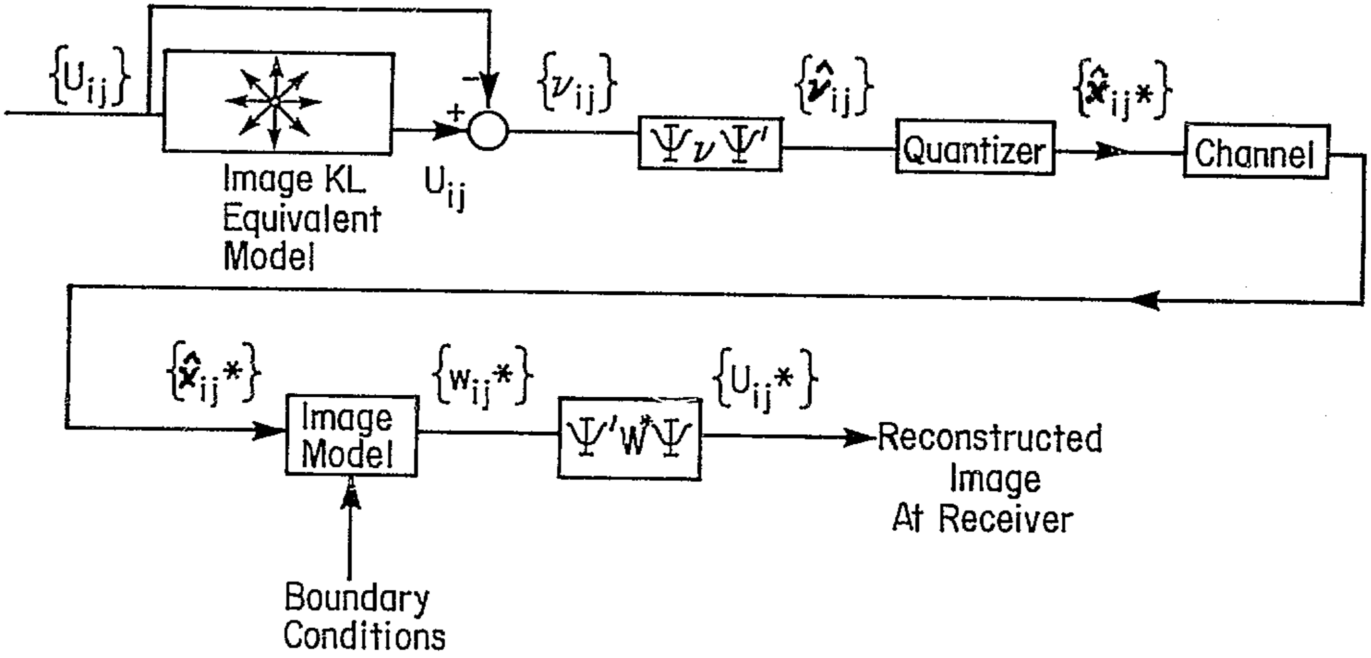


FIGURE 3b: Fast KL Transform Image Coding Algorithm

4. Discussion

In practical implementation of this algorithm, N + 1 should equal 2^n for some integer n, so that an FFT algorithm can be used to implement the sine transform. The value of n_{11} in (58b) is an experimental parameter to be chosen to minimize the mean square error. In our experiments we found the values of n_{11} between 6 to 8 quite satisfactory. These values were determined by testing the mean square error of a few randomly chosen image blocks of Channel 1 image.

For large image block size $N \geq 63$, and/or small correlation parameter value $\rho \leq 0.7$, the effect of boundary values on coder performance becomes less significant. Hence, instead of taking an (N+2) x (N+2) size image block and using the actual boundary values, if an (NxN) image block is used and all the boundary values are assumed to be equal to the block mean then the performance of this coding scheme is unchanged. For 15 x 15 image blocks with $\rho \leq .07$, this was experimentally verified. Note that this approximation does not change the actual boundary value around each image block, but only uses the mean value as the pseudo boundary of the block.

CHAPTER IV

FAST KARHUNEN LOEVE TRANSFORM DATA COMPRESSION STUDIES

The algorithm outlined in the previous chapter was simulated on four 255 x 255 picture elements multispectral earth data images. The digitized images were provided by NASA and are the same as used by TRW, Inc. in a data compression study under Ames Research Center contract NAS2-8394. In all coding experiments, the boundary conditions were assumed to be constant and equal to the mean value of the image block being coded.

1. Source Data

   a. Figures (4) through (7) show the original four multispectral earth data images.

   b. Table 1 lists the calculated statistics of the original four multispectral earth data images.

   c. Figures (8) and (9) show the histograms of the original Channel 1 and Channel 3 multispectral images. The histograms of Channel's 2 and 4 are similar to those of Channels 1 and 3 respectively.

In image coding simulations, two different block sizes were used. First, the entire source image was used as a single 255 x 255 image block. Next, the 255 x 255 source image was divided into 289 *(15 x 15) small image blocks. Unless otherwise stated, (see Table 5) uniform quantizers were used in all the coding experiments. The improvement of a compandor over a uniform quantizer was found to be less than 0.5db in terms of S/N ratio. The performance of different coding schemes is judged by (i) Mean square error and (ii) Signal to Noise Ratio. This is to enable us to make comparison with other techniques. The mean square error in encoding the kth channel image is defined as

$$e_k^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (u_{i,j,k} - u^*_{i,j,k})^2 \qquad k = 1,2,\ldots,4$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

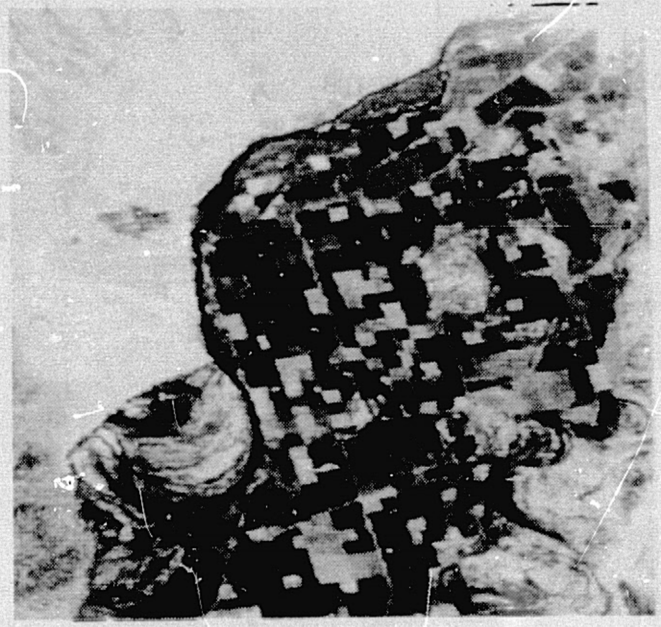


FIGURE 4 CHANNEL 1 IMAGE

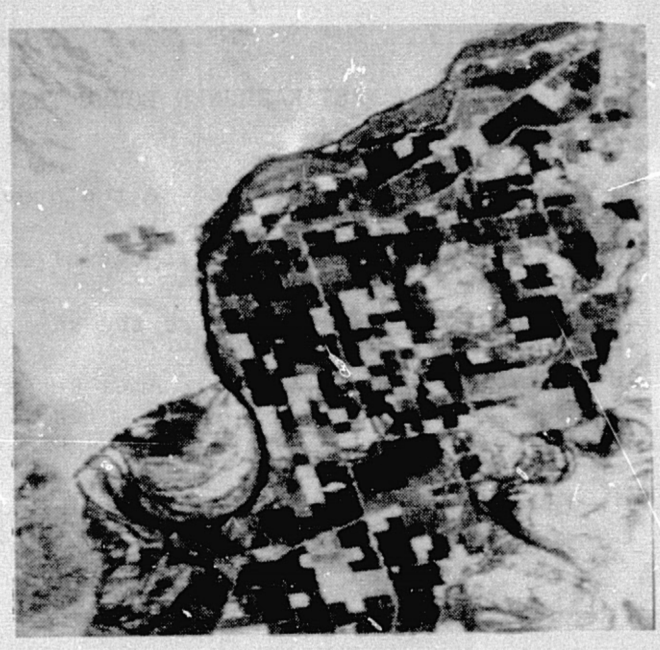


FIGURE 5 CHANNEL 2 IMAGE

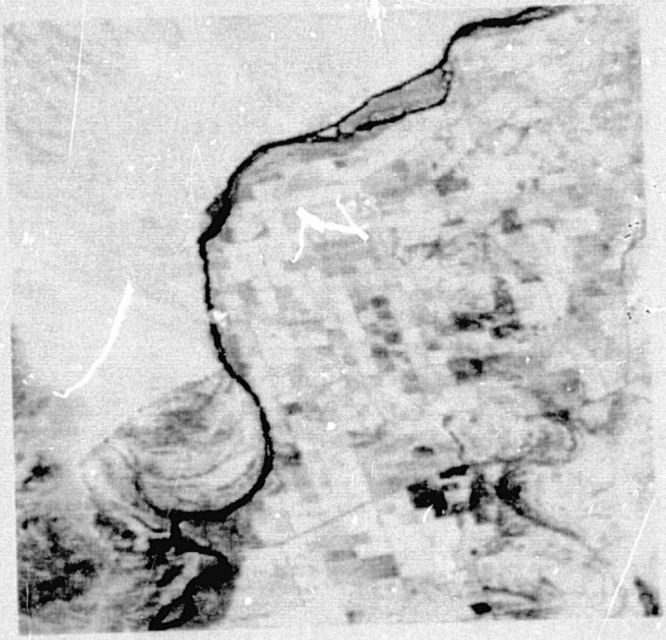


FIGURE 6 CHANNEL 3 IMAGE

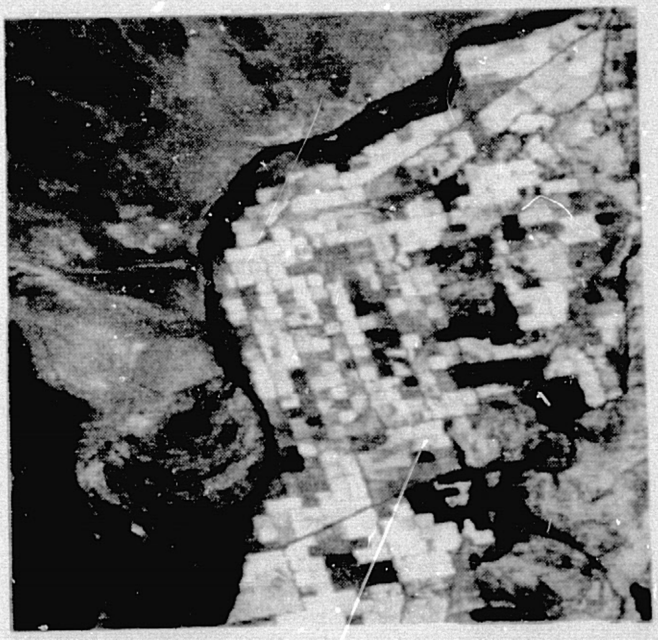
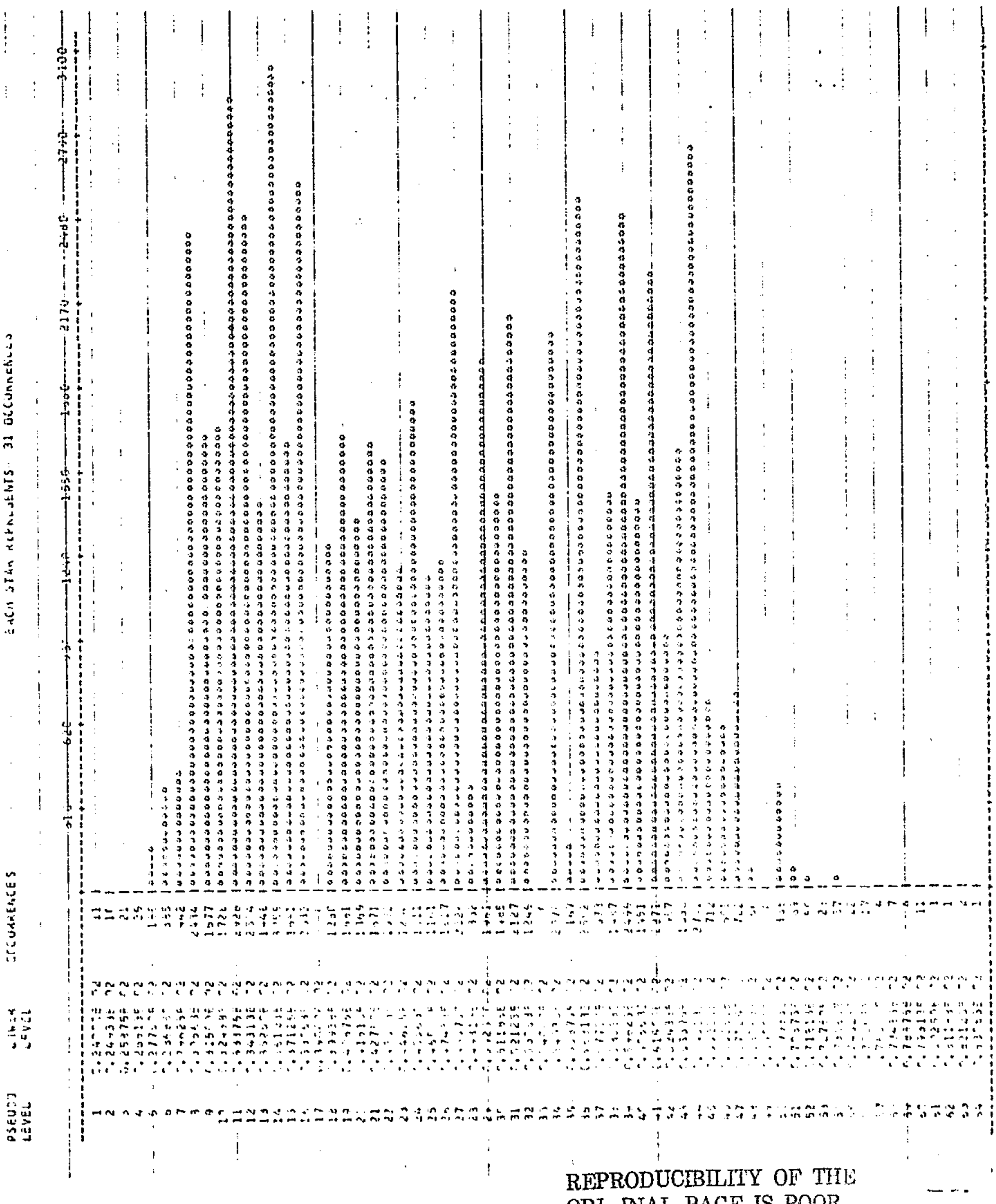


FIGURE 7 CHANNEL 4 IMAGE

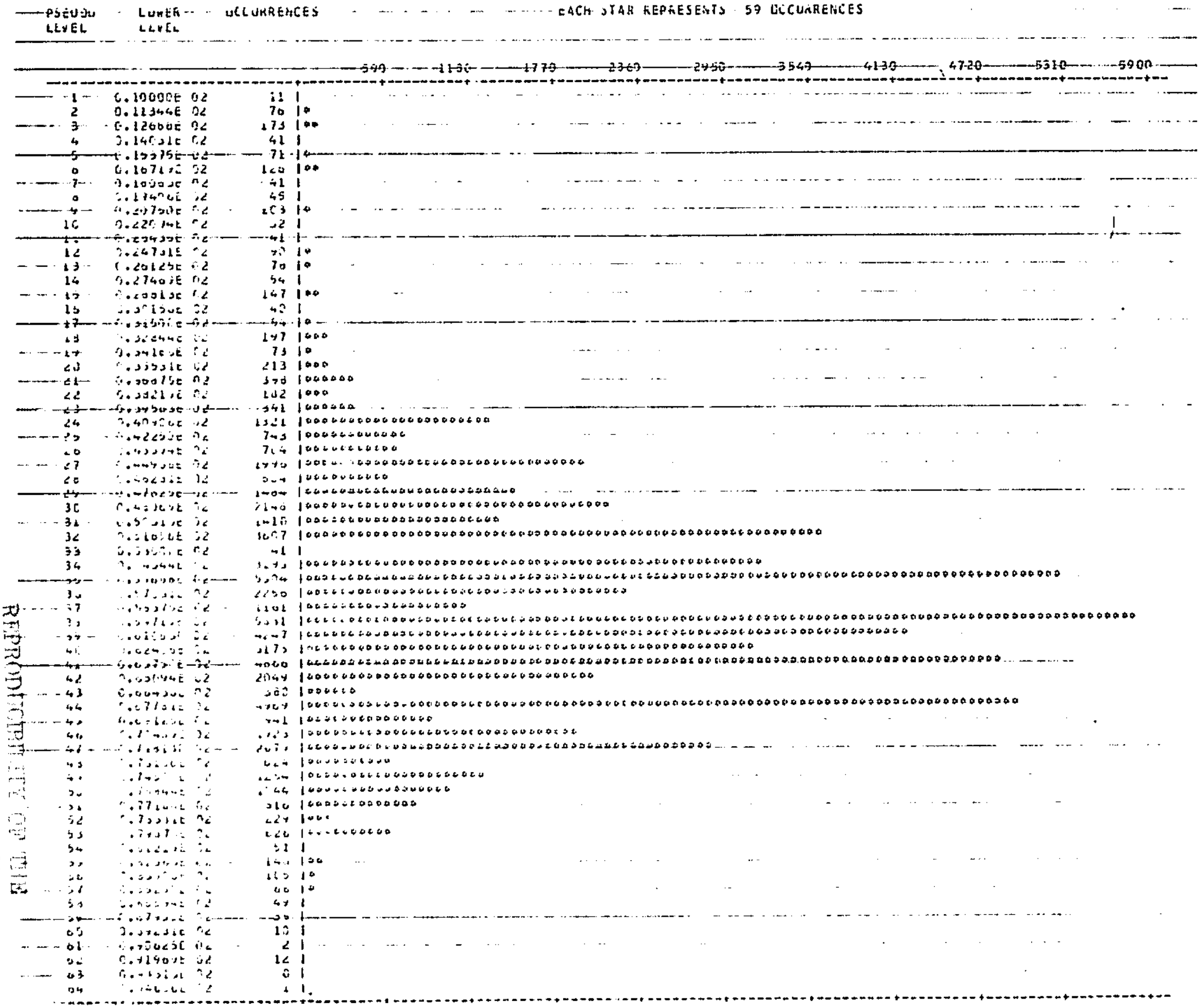
Statistical Parameters Channel No.	Mean	Variance	Average Correlation Parameter ρ (255 x 255 Coding)	Average Correlation Parameter ρ (Block by Block Coding)	
				ρ hori	ρ ver
Channel 1	47.80	130.76	0.93	0.710	0.597
Channel 2	52.13	333.16	0.92	0.730	0.620
Channel 3	59.56	121.16	0.88	0.712	0.591
Channel 4	26.82	41.10	0.89	0.723	0.614

TABLE 1 The Statistics of the 4 Multispectral Earth Data Images



REPRODUCIBILITY OF THE ORI INAL PAGE IS POOR

FIGURE 8 HISTOGRAM OF CHANNEL 1 IMAGE



```
  PSEUDO    LOWER     OCCURRENCES                  EACH STAR REPRESENTS  59 OCCURRENCES
  LEVEL     LEVEL
                                    590    1180    1770    2360    2950    3540    4130    4720    5310    5900
     1   0.10000E 02      11
     2   0.11344E 02      76
     3   0.12688E 02     173
     4   0.14031E 02      41
     5   0.15375E 02      71
     6   0.16719E 02     120
     7   [illegible]      41
     8   [illegible]      45
     9   0.20750E 02     103
    10   0.22094E 02      32
    11   [illegible]      41
    12   0.24781E 02      90
    13   0.26125E 02      70
    14   0.27469E 02      54
    15   [illegible]     147
    16   [illegible]      42
    17   [illegible]      64
    18   [illegible]     197
    19   0.34188E 02      73
    20   0.35531E 02     213
    21   [illegible]     378
    22   [illegible]     182
    23   [illegible]     341
    24   [illegible]    1321
    25   [illegible]     743
    26   [illegible]     704
    27   [illegible]    1990
    28   [illegible]     584
    29   [illegible]    1404
    30   [illegible]    2146
    31   [illegible]    1410
    32   [illegible]    3607
    33   [illegible]      41
    34   [illegible]  [illegible]
    35   [illegible]  [illegible]
    36   [illegible]    2250
    37   [illegible]    1101
    38   [illegible]  [illegible]
    39   [illegible]  [illegible]
    40   [illegible]    3175
    41   [illegible]    4060
    42   [illegible]    2049
    43   [illegible]     380
    44   [illegible]    4969
    45   [illegible]     941
    46   [illegible]  [illegible]
    47   [illegible]  [illegible]
    48   [illegible]     624
    49   [illegible]  [illegible]
    50   [illegible]  [illegible]
    51   [illegible]     816
    52   [illegible]     229
    53   [illegible]     626
    54   [illegible]      51
    55   [illegible]     144
    56   [illegible]     115
    57   [illegible]      66
    58   [illegible]      49
    59   [illegible]      36
    60   [illegible]      10
    61   0.90625E 02       2
    62   0.91969E 02      12
    63   [illegible]       0
    64   [illegible]       1
```

FIGURE 9 HISTOGRAM OF CHANNEL 3 IMAGE

where N = 255, $u_{i,j,k}$ and $u^*_{i,j,k}$ represent the original and the reconstructed images for the kth channel. Channels 1,2,3 are given to be 7 bit images and Channel 4 is a 6 bit image. The signal to noise ratio for channel k image is defined in Decibels as

$$s_k = 10\log_{10} \frac{(p\text{-}p)_k^2}{e_k^2}$$

where $(p\text{-}p)_k$ = peak to peak signal value

= 127 for k = 1,2,3 and 63 for k = 4.

In comparing the results of this study with other methods, bit rate vs. average mean square error curves and bit rate vs. average S/N ratio curves are generated (Figures 1 and 16). This is done by calculating the average mean square error for a fixed bit rate as

$$e^2 = \tfrac{1}{4} \sum_{k=1}^{4} \varepsilon_k^2$$

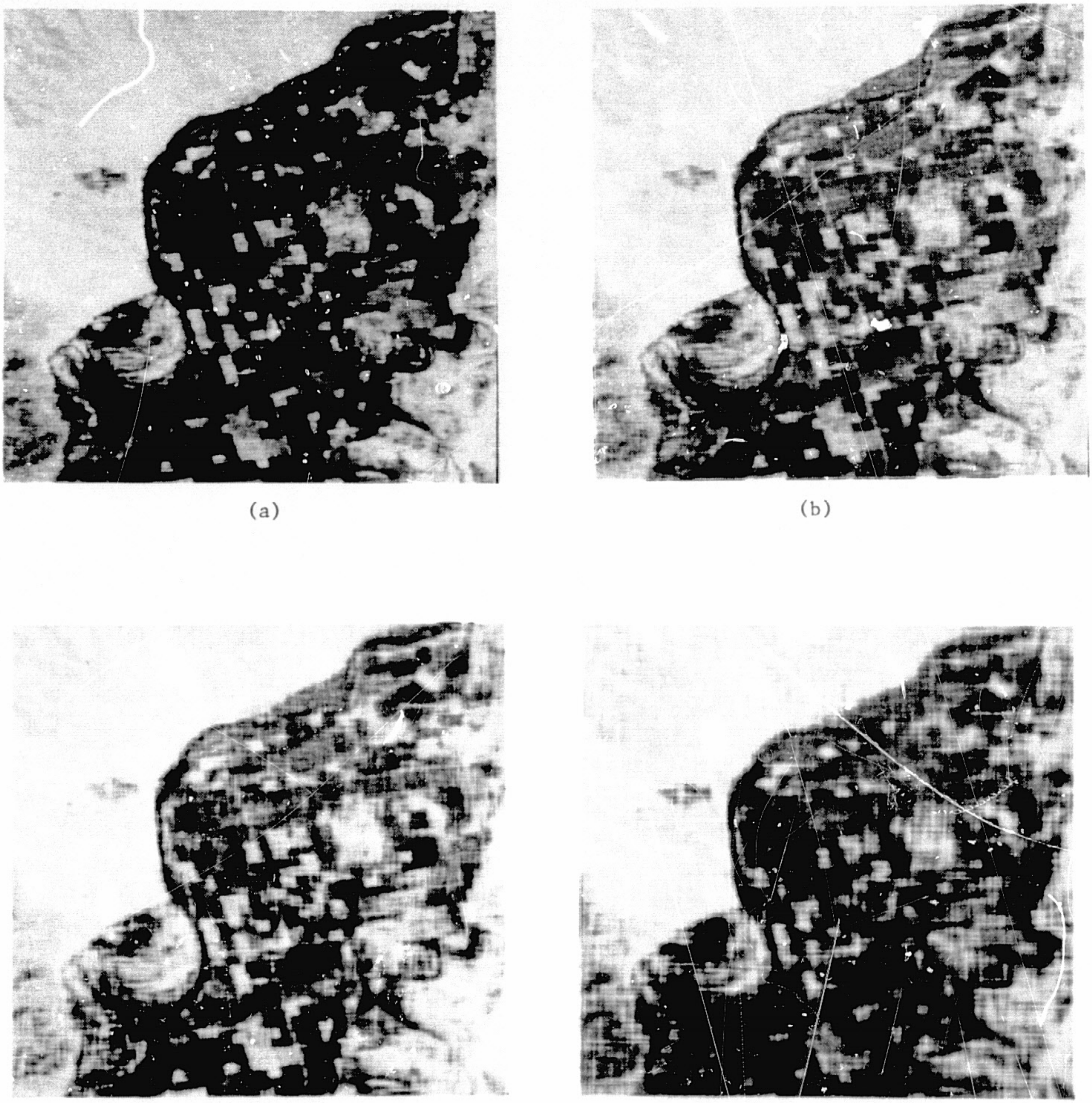
and calculating average S/N ratio for a fixed bit rate as

$$s = \tfrac{1}{4} \sum_{k=1}^{4} s_k$$

These definitions are consistent with the definitions used by TRW, Inc. in their study. The implications shortcomings of these average performance indices will be discussed in section 5.

2. Fast KLT Coding of 255 x 255 Images

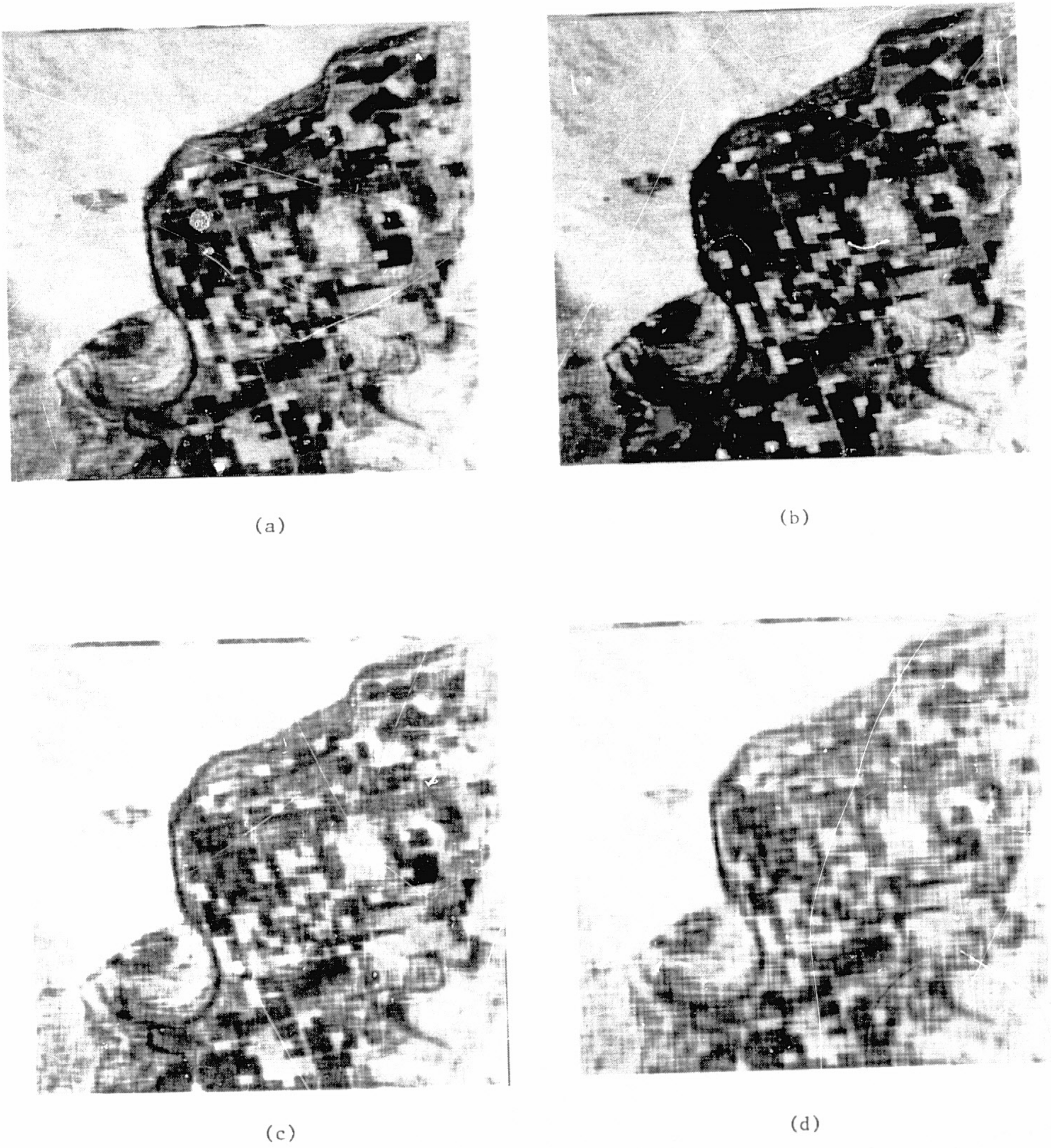
a. Figure (10) shows the 255 x 255 encoded image of the Channel 1 earth data image with 2 bits/pixel average, 1 bits/pixel average, 0.5 bits/pixel average, and 0.25 bits/pixel average. Figures (11) through (13) are the 255 x 255 encoded images for Channel 2, Channel 3, and Channel 4, respectively. All these images were photographed on a Dicomed scanner after contrast stretching each image according to its histogram.



(a) (b)

(c) (d)

FIGURE 10 Fast KLT 255 x 255 encoded image of the Channel 1 Image
(a) 2 bits/pixel (b) 1 bits/pixel
(c) 0.5 bits/pixel (d) 0.25 bits/pixel



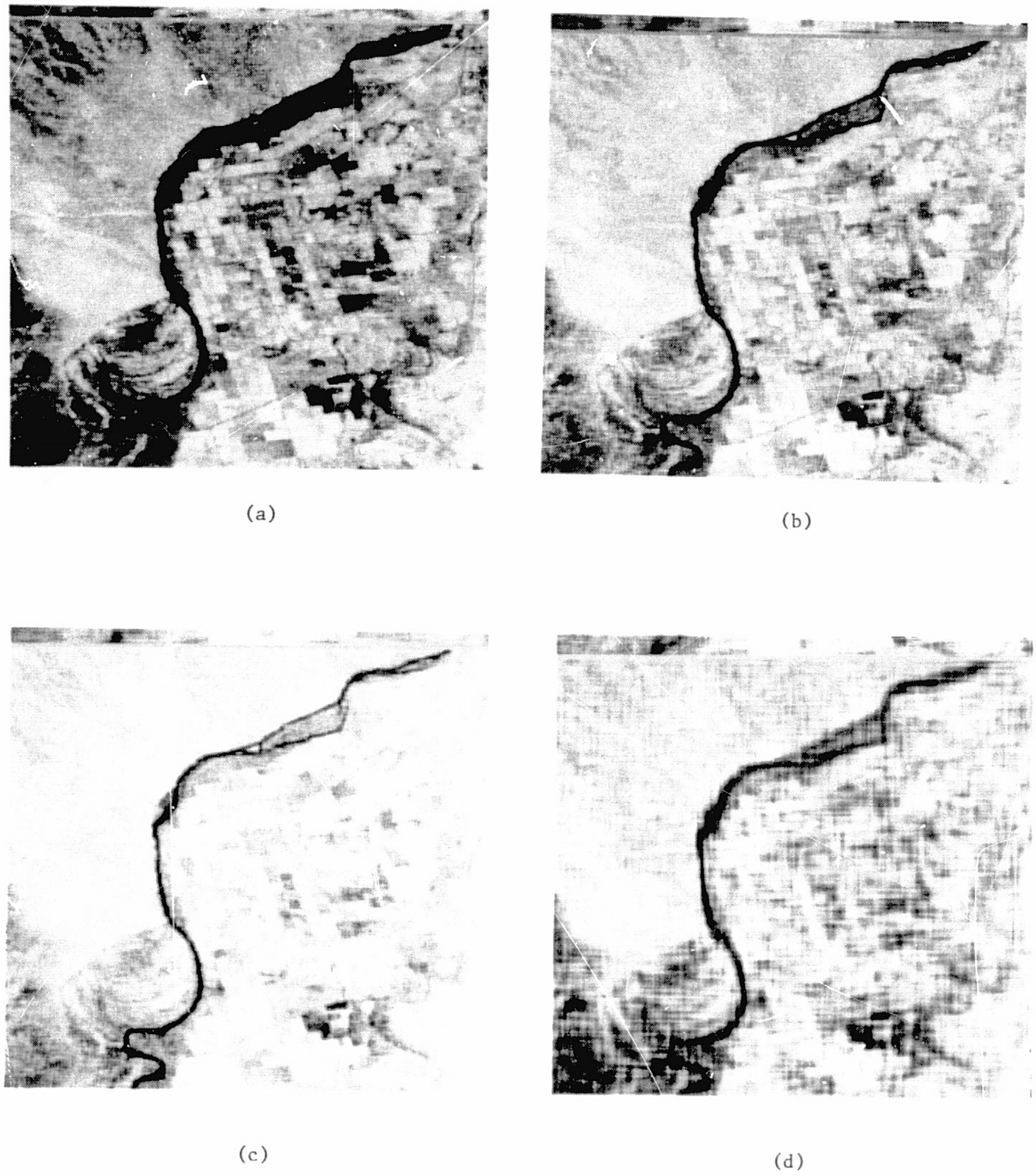
(a)

(b)

(c)

(d)

FIGURE 11 Fast KLT 255 x 255 encoded image of the Channel 2 Image
(a) 2 bits/pixel (b) 1 bits/pixel
(c) 0.5 bits/pixel (d) 0.25 bits/pixel



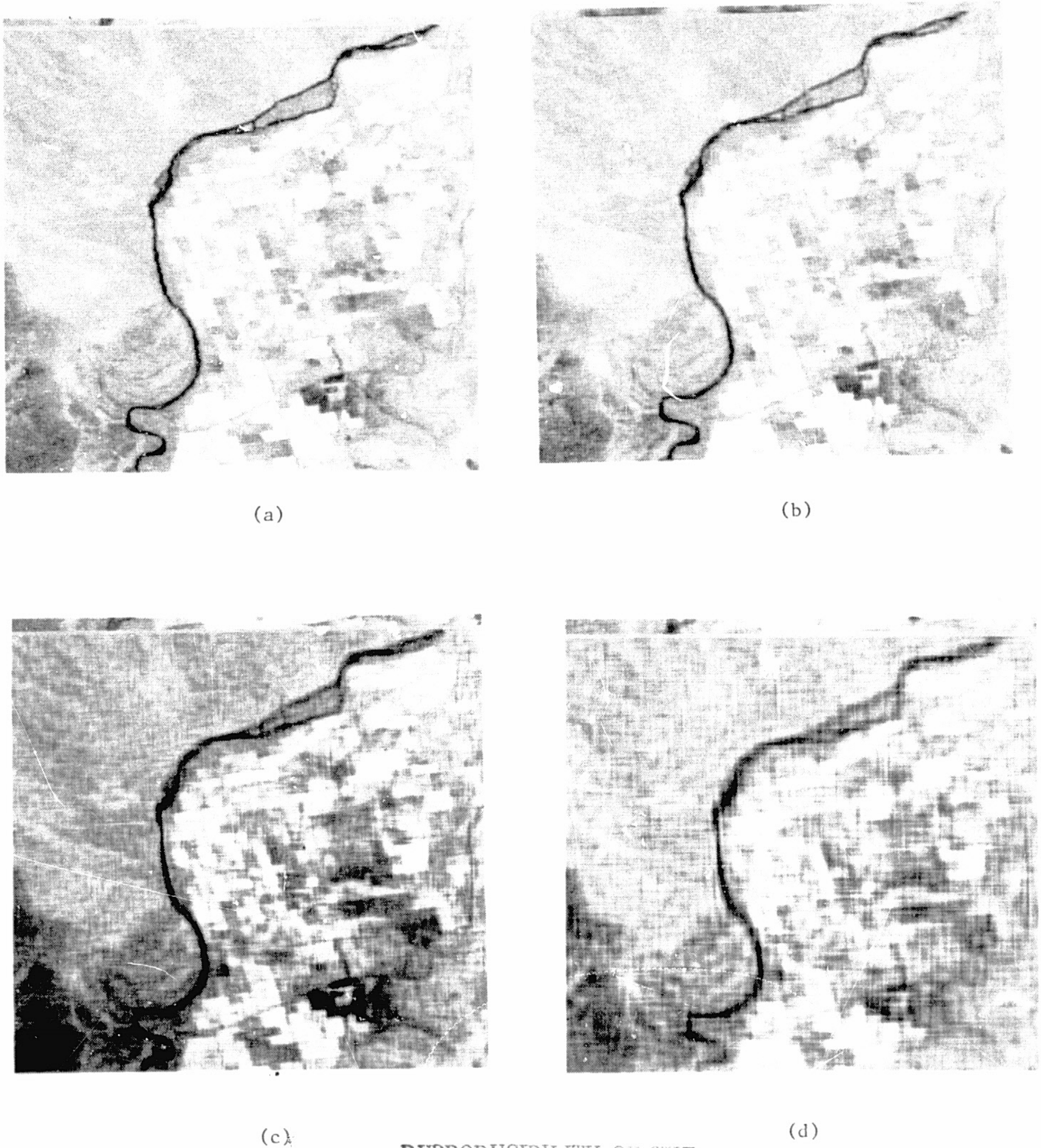
(a)

(b)

(c)

(d)

FIGURE 12 Fast KLT 255 x 255 endoded image of the Channel 3 Image
(a) 2 bits/pixel (b) 1 bits/pixel
(c) 0.5 bits/pixel (d) 0.25 bits/pixel



(a)

(b)

(c)

(d)

FIGURE 13 Fast KLT 255 x 255 encoded image of the Channel 4 Image
(a) 2 bits/pixel (b) 1 bits/pixel
(c) 0.5 bits/pixel (d) 0.25 bits/pixel

b. Table 2 lists the calculated mean square error and signal (peak to peak) to noise (root mean square) ratio of the 255 x 255 encoded image for each channel earth data images.

c. Figure 14 (solid lines) shows the bit rate vs. means square error of table (2).

d. Figure 15 (solid lines) shows the bit rate vs. signal (peak to peak) to noise (root mean square) ratio of Table 2.

e. Table 2 also lists the average mean square error of each channel earth data images.

f. Figures (1) and (16) show the comparison of these results with the results of TRW, Inc. effort cited above [20].

g. Figure (17) shows the bit allocation pattern of a 255 x 255 image at 1 bit/pixel average for Channel 1. The value of n_{11} (number of bits for the first pixel) is a design parameter, in our experiment, several different values of n_{11} were tried and the values between 6 to 8 (for ρ=0.93) were found to be most suitable. In order to simplify the quantizer design, all the elements in the FKL domain that are assigned equal numbers of bits may be assumed to have equal variances.

h. Figures (18) and (19) show the output histograms of Channels 1 and 3 images with 1 bit/pixel average.

In calculating the average bit rate, we define a parameter (called pseudo bit rate constant) PP. For different values of PP the corresponding values of the average actual bit rate denoted by ACURAT are printed out. Then the user can pick a desired average bit rate, i.e. ACURAT, and from this list [see Figure (20)] find the corresponding value of PP. The value of PP is used in the fast KLT coding algorithm program.

	255 x 255 Encoded Images									
Bit Rate	Channel 1		Channel 2		Channel 3		Channel 4		Average Over All Channels	
	M.S.E.	SNR in db	M.S.E.	SNR in db	M.S.E.	SNR in db	M.S.E.	SNR in db	Average M.S.E.	Average SNR in db
2	2.48	38.13	7.39	33.39	3.10	37.19	0.88	36.54	3.46	36.31
1	4.86	35.20	14.46	30.47	8.11	32.96	2.27	32.43	7.43	32.77
0.5	10.12	32.02	30.05	27.28	17.80	29.57	5.19	28.84	15.79	29.43
0.25	18.02	29.51	53.87	24.76	29.93	27.32	9.13	26.38	27.14	26.99

TABLE 2 M.S.E.: Mean square error

SNR in db: Signal*(p-p) to noise (rms) ratio in db

*: p-p values are 127 for Channel 1, 2, 3
63 for Channel 4

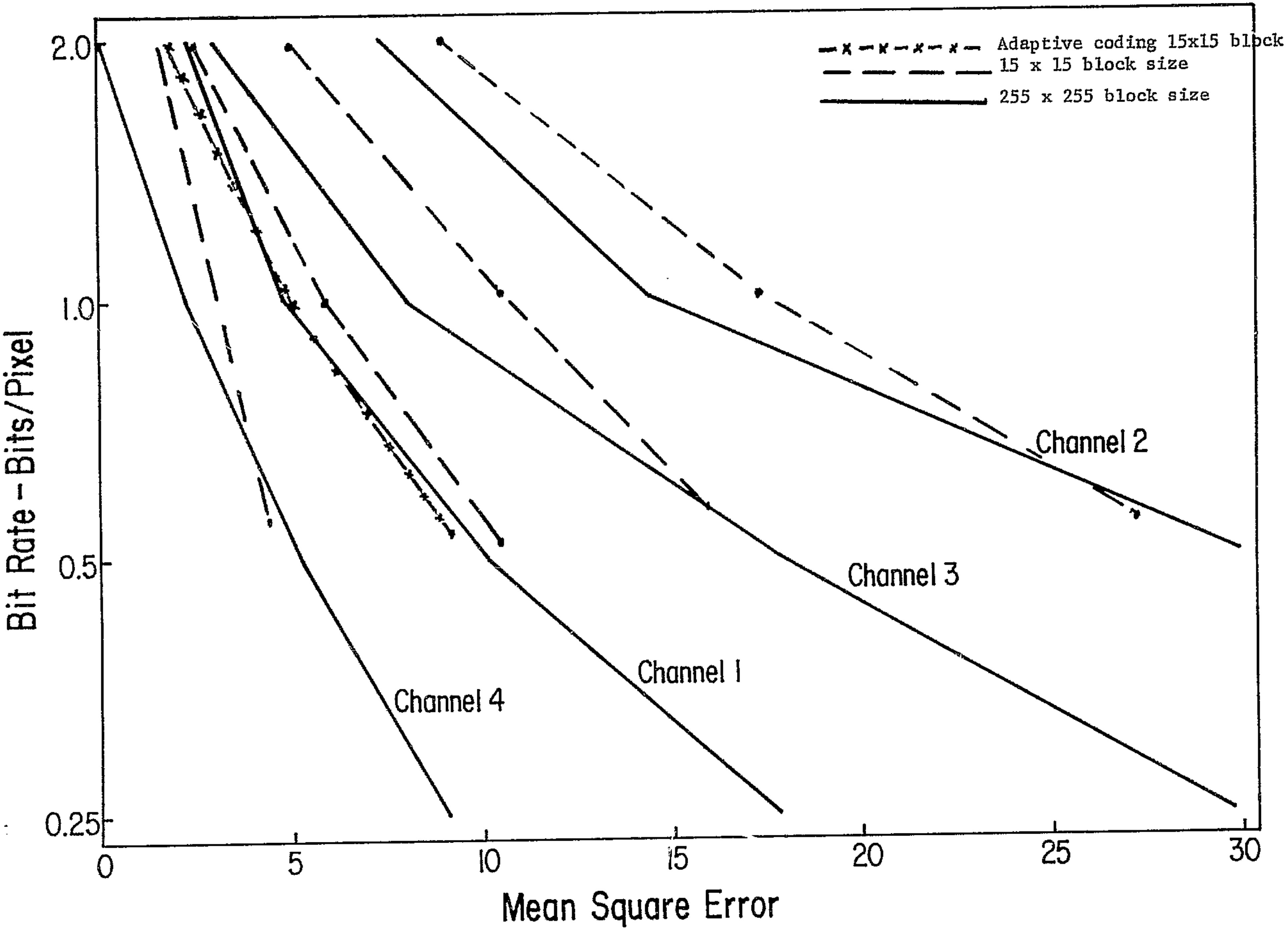


FIGURE 14: Bit rate vs. Mean square error for the multispectral Images.

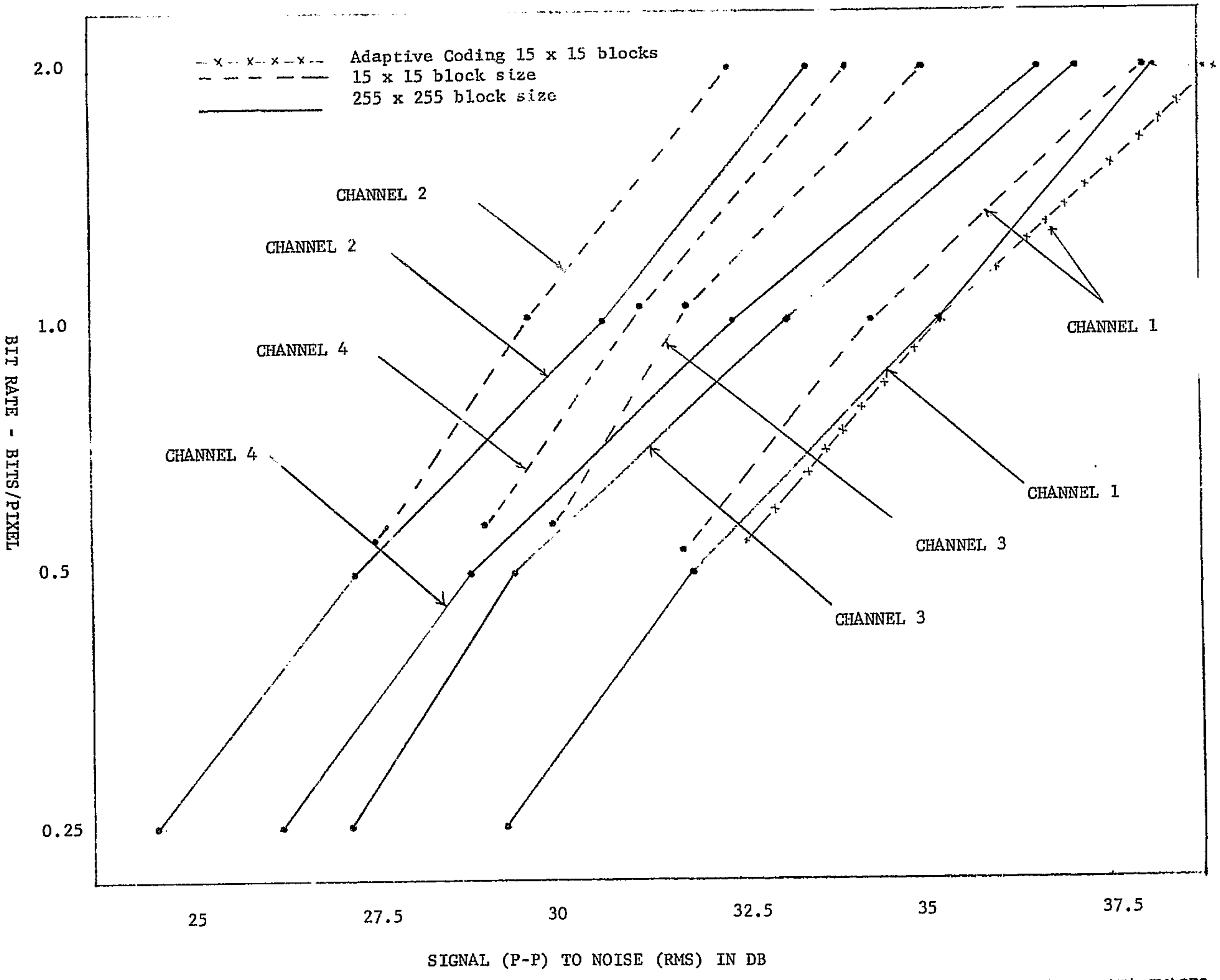


FIGURE 15 BIT RATE VERSUS SIGNAL TO NOISE RAITO (db) of the 4 ENCODED MULTISPECTRAL EARTH DATA IMAGES

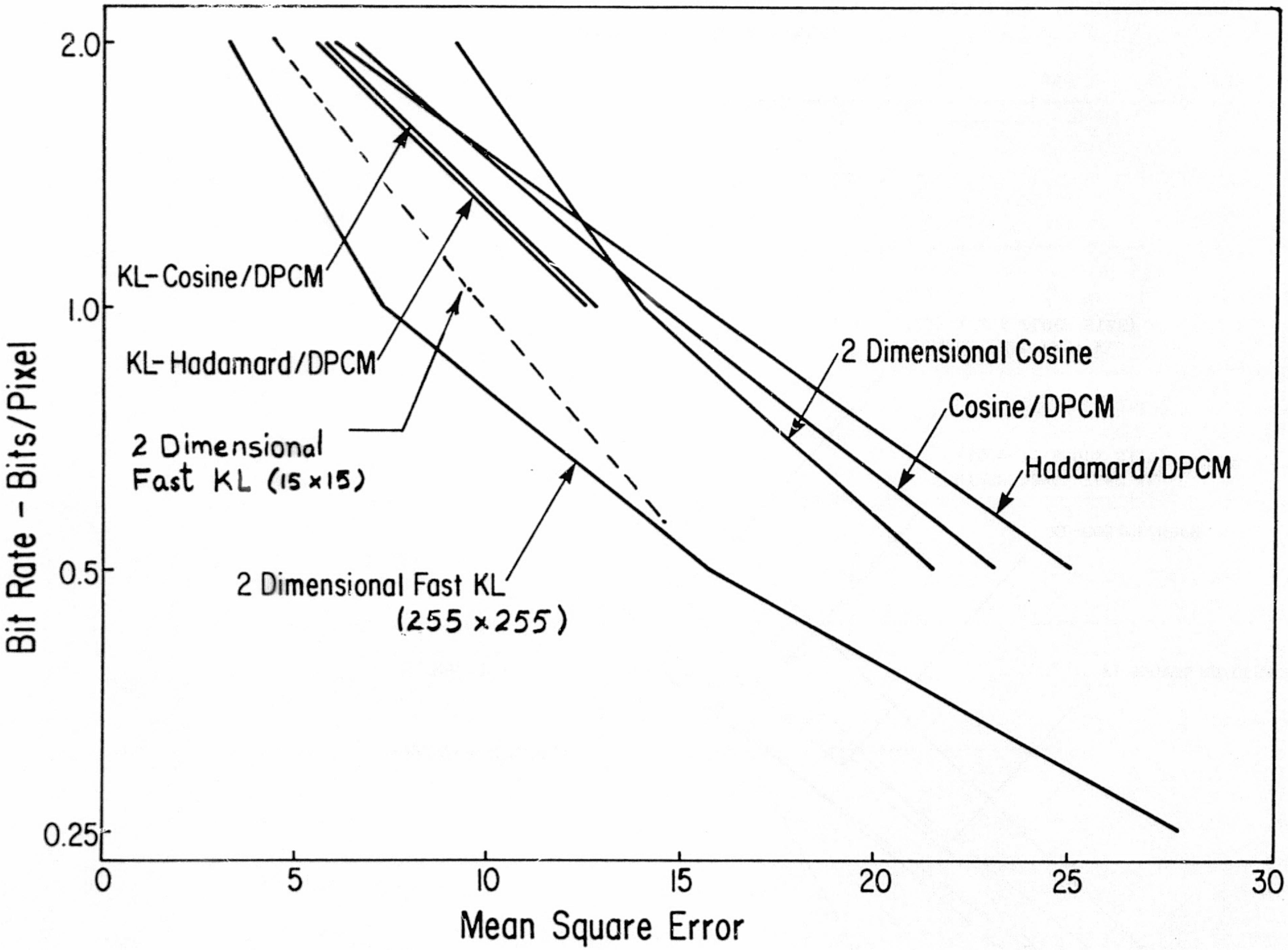


FIGURE 1   Comparison of mean square error for 2 Dimensional Fast KL vs. Other Methods

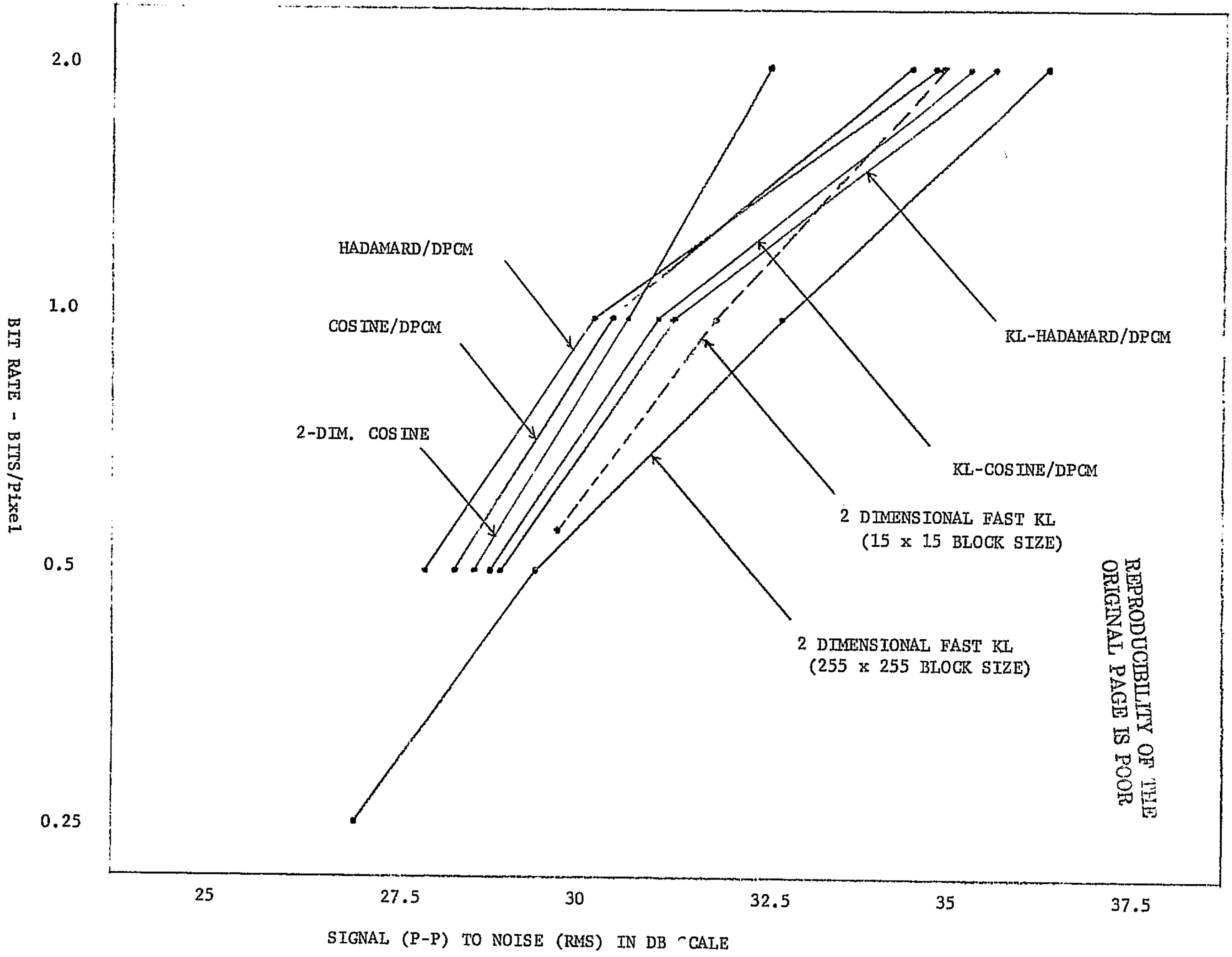


FIGURE 16 COMPARISON OF SIGNAL TO NOISE RATIO (db) FOR 2 DIMENSIONAL FAST KL VS. OTHER METHODS (Uniform Quantizer was used in the fast KL coding Experiments here)

```
7 7 6 6 6 6 5 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
7 6 6 6 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3
6 6 6 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3
6 6 5 5 5 5 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
6 5 5 5 5 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2
6 5 5 5 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2
5 5 5 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2
5 5 5 4 4 4 4 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
5 5 4 4 4 4 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
5 5 4 4 4 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
5 5 4 4 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
5 4 4 4 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1
5 4 4 4 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1
5 4 4 4 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1
4 4 4 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
4 4 4 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
4 4 4 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 4 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 4 3 3 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4 4 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
4 3 3 3 3 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
4 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
4 3 3 3 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
4 3 3 3 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
```

FIGURE 17 a: Bit allocation pattern for a 255 x 255 image with 1 bits/pixel average. The Figure shows bits at every fourth location in the upper left quadrant of the transformed Channel 1 Image.

```
7   7   7   7   7   7   7   6   6   6   6   6   6   6   6   6
7   7   7   7   7   7   6   6   6   6   6   6   6   6   6   6
7   7   7   7   7   7   6   6   6   6   6   6   6   6   6   6
7   7   7   7   7   6   6   6   6   6   6   6   6   6   6   6
7   7   7   7   6   6   6   6   6   6   6   6   6   6   6   6
7   7   7   6   6   6   6   6   6   6   6   6   6   6   5   5
7   6   6   6   6   6   6   6   6   6   6   6   6   5   5   5
6   6   6   6   6   6   6   6   6   6   6   6   5   5   5   5
6   6   6   6   6   6   6   6   6   6   6   5   5   5   5   5
6   6   6   6   6   6   6   6   6   6   5   5   5   5   5   5
6   6   6   6   6   6   6   6   6   5   5   5   5   5   5   5
6   6   6   6   6   6   6   6   5   5   5   5   5   5   5   5
6   6   6   6   6   6   6   5   5   5   5   5   5   5   5   5
6   6   6   6   6   6   5   5   5   5   5   5   5   5   5   5
6   6   6   6   6   5   5   5   5   5   5   5   5   5   5   5
6   6   6   6   6   5   5   5   5   5   5   5   5   5   5   5
```

FIGURE 17b: Bit allocation pattern in the upper left 16 x 16 region of the transformed Channel 1 Image.

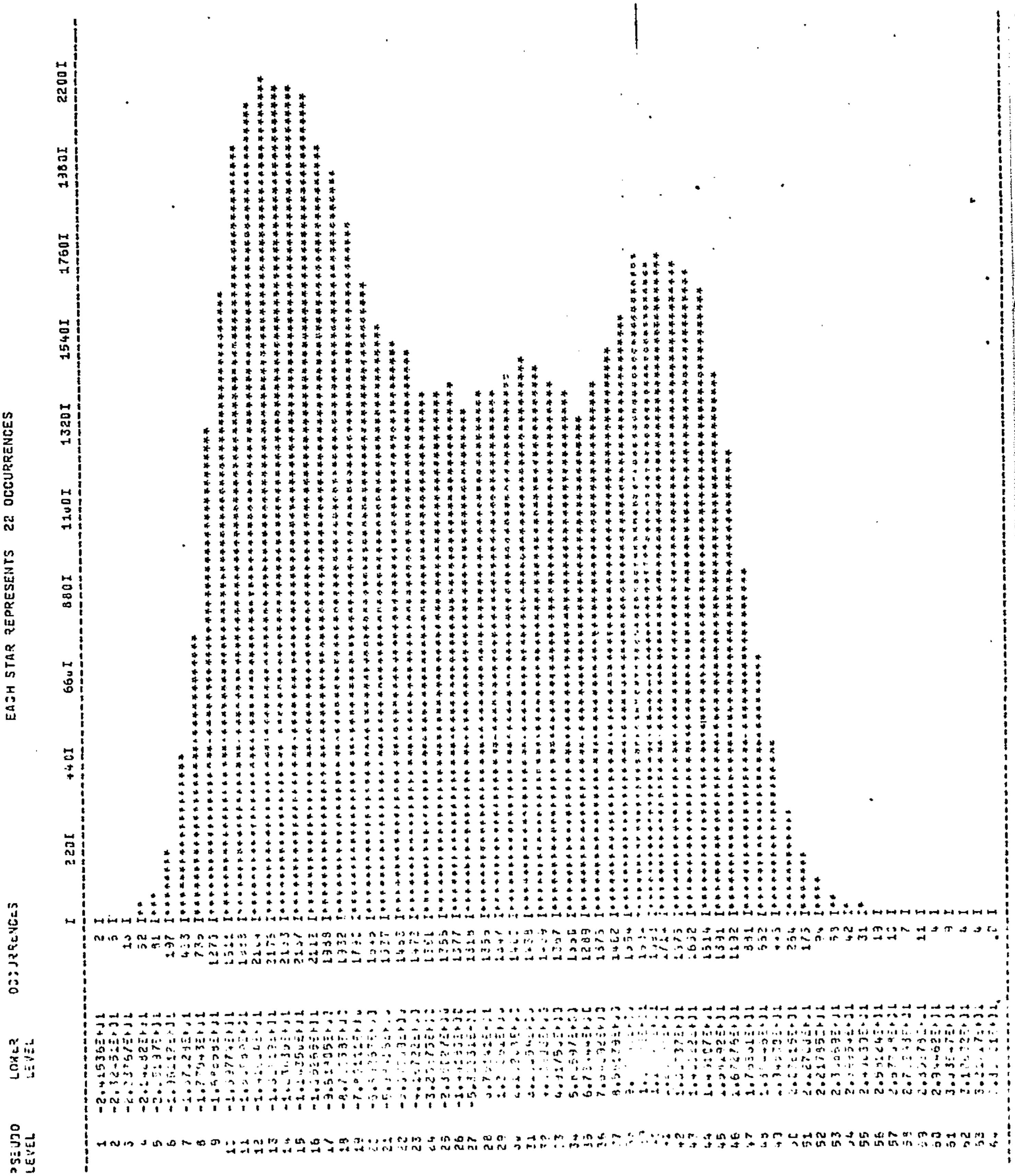
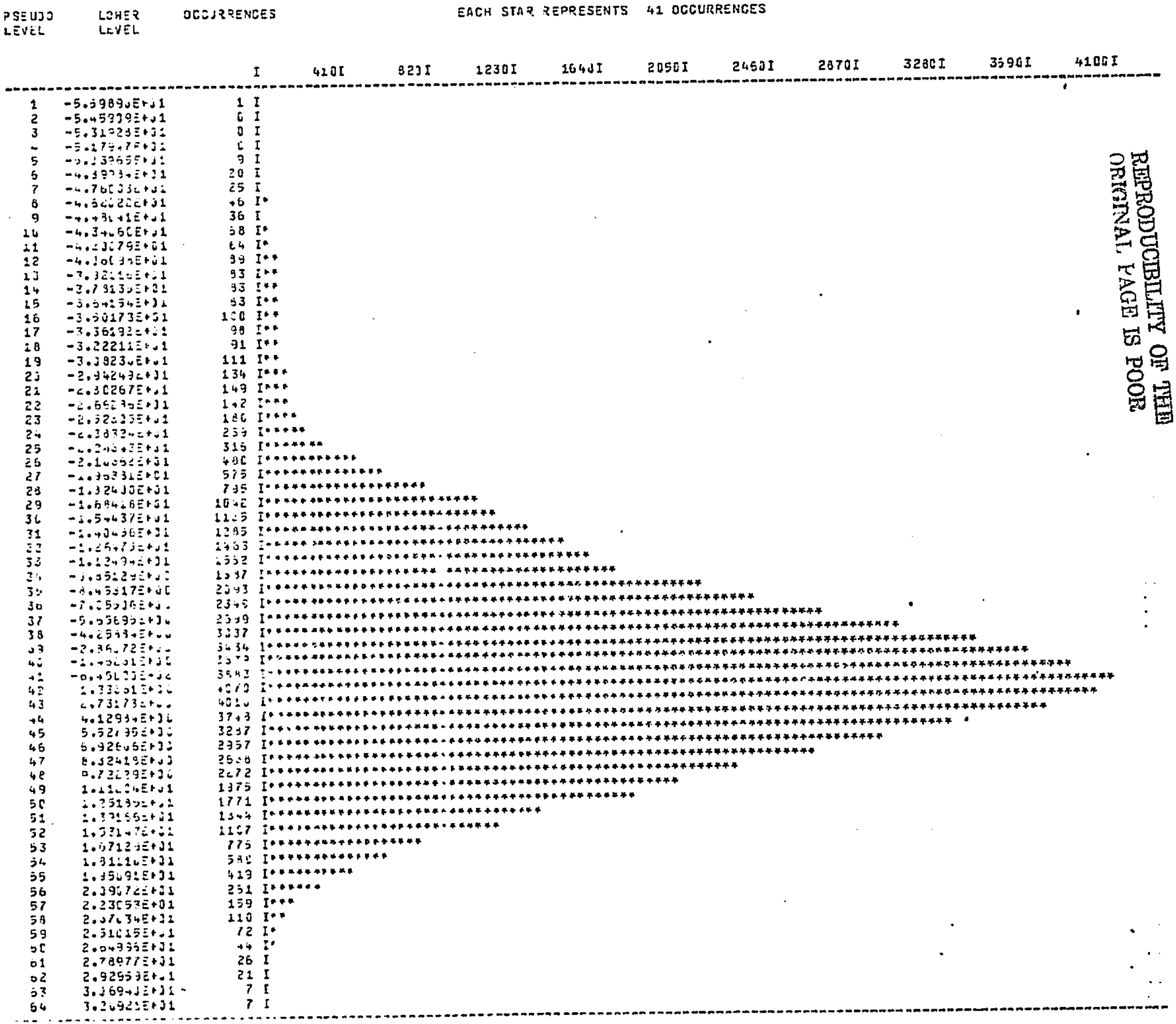


FIGURE 18 HISTOGRAM of the 255 x 255 encoded image of the Channel 1 Image with 1 bits/pixel Average



PSEUDO LEVEL	LOWER LEVEL	OCCURRENCES
1	-5.59890E+01	1
2	-5.45839E+01	0
3	-5.31928E+01	0
4	-5.17947E+01	0
5	-5.03965E+01	9
6	-4.89984E+01	20
7	-4.76003E+01	25
8	-4.62022E+01	46
9	-4.48041E+01	36
10	-4.34060E+01	58
11	-4.20079E+01	64
12	-4.06098E+01	99
13	-3.92116E+01	93
14	-3.78135E+01	83
15	-3.64154E+01	83
16	-3.50173E+01	100
17	-3.36192E+01	98
18	-3.22211E+01	91
19	-3.08230E+01	111
20	-2.94249E+01	134
21	-2.80267E+01	149
22	-2.66286E+01	142
23	-2.52305E+01	186
24	-2.38324E+01	259
25	-2.24343E+01	315
26	-2.10362E+01	400
27	-1.96381E+01	575
28	-1.82400E+01	795
29	-1.68418E+01	1042
30	-1.54437E+01	1125
31	-1.40456E+01	1285
32	-1.26475E+01	1463
33	-1.12494E+01	1552
34	-9.85129E+00	1637
35	-8.45317E+00	2093
36	-7.05506E+00	2345
37	-5.65695E+00	2549
38	-4.25884E+00	3337
39	-2.86072E+00	3434
40	-1.46261E+00	3573
41	-6.45033E-02	3983
42	1.33361E+00	4070
43	2.73173E+00	4010
44	4.12984E+00	3743
45	5.52795E+00	3237
46	6.92606E+00	2957
47	8.32419E+00	2568
48	9.72229E+00	2272
49	1.11204E+01	1975
50	1.25185E+01	1771
51	1.39166E+01	1344
52	1.53147E+01	1167
53	1.67128E+01	775
54	1.81110E+01	582
55	1.95091E+01	419
56	2.09072E+01	251
57	2.23053E+01	159
58	2.37034E+01	110
59	2.51015E+01	72
60	2.64996E+01	44
61	2.78977E+01	26
62	2.92958E+01	21
63	3.06940E+01	7
64	3.20921E+01	7

FIGURE 19 HISTOGRAM of the 255 x 255 encoded image
of the Channel 3 Image with 1 bit/pixel Average

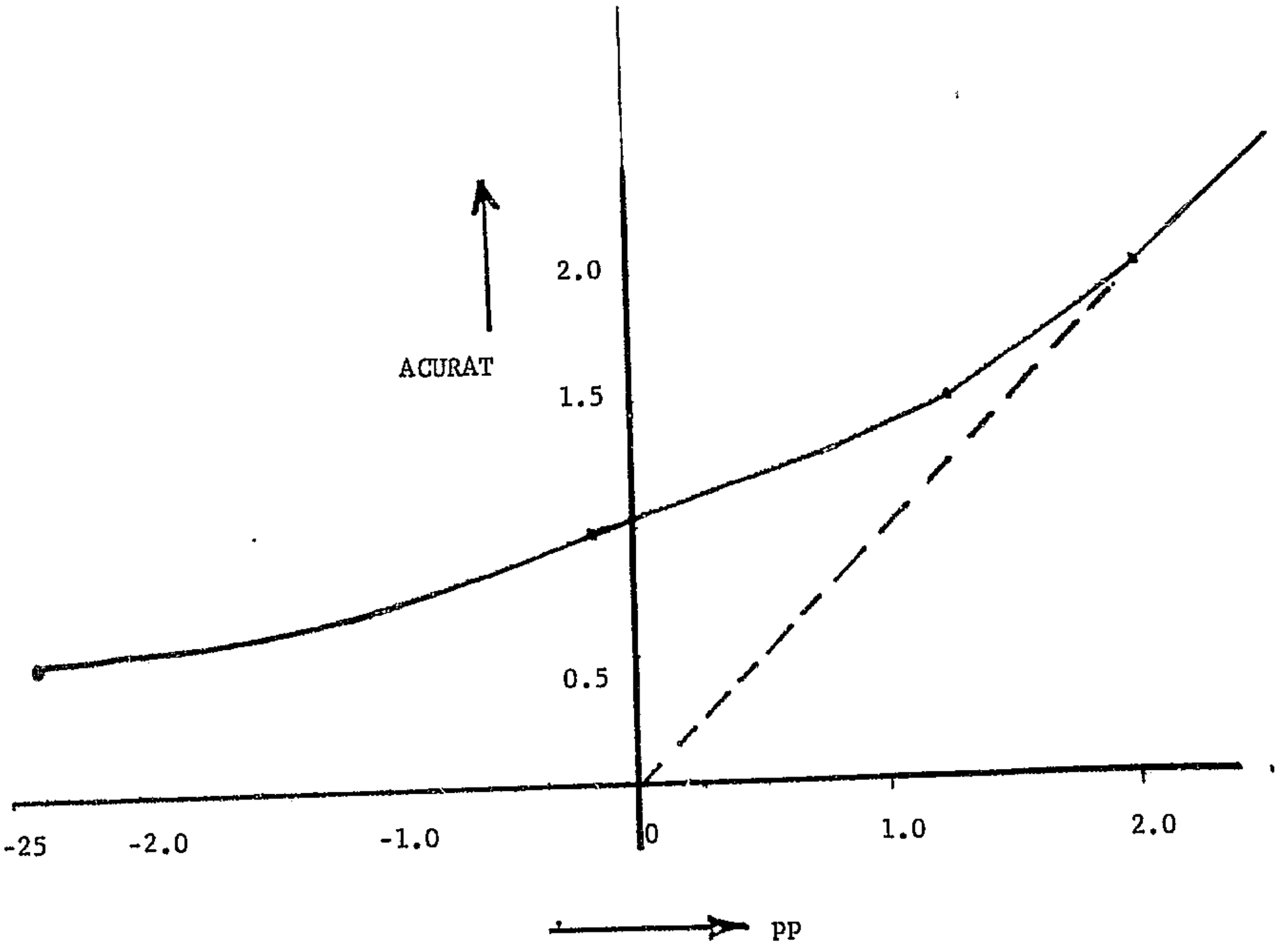


FIGURE 20 Actual Bit Rate vs. Pseudo Bit Rate Constant PP for Channel 1 for 255 x 255 Coding Block

3. Fast KLT Coding of 15 x 15 Image Blocks

In block by block coding, the correlation parameter values as indicated in the last two columns of table 1, were used. These are the average horizontal and vertical correlation parameter values for a 15 x 15 block.

a. Figures (21) and (23) show the block by block encoded image at various bit rates of Channels 1, 2, 3 images respectively.

b. Table 3 lists the calculated mean square error and signal (root mean square) to noise (root mean square) ratio at various bit rates of the four block by block encoded images.

c. Figure (14) (dotted lines) shows the bit rate vs. mean square error of Table 3.

d. Figure (15) (dotted lines) shows the bit rate vs. signal (root mean square) to noise (root mean square) ratio of Table 3.

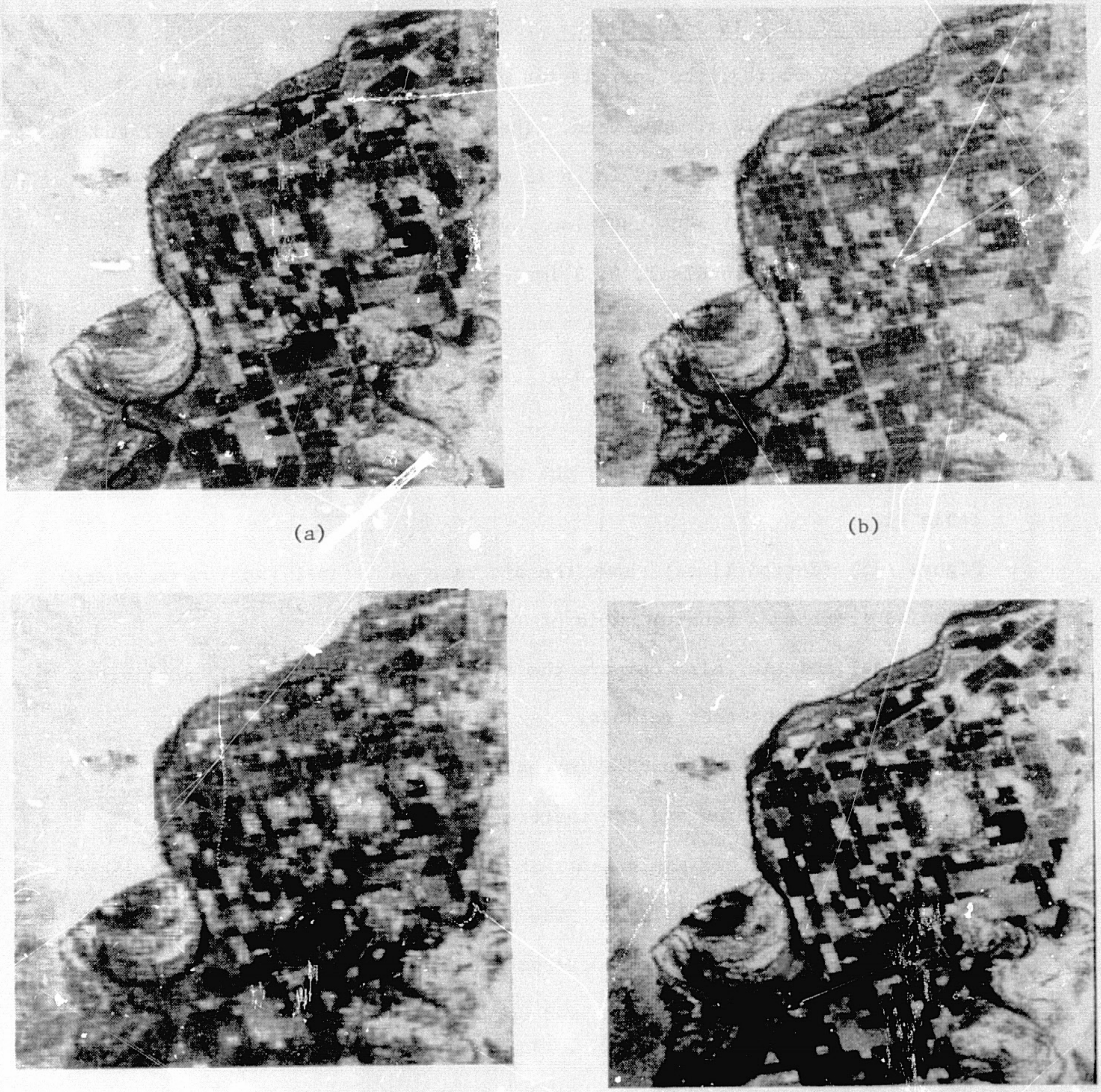
e. Figures (1) and (16) also compare the average performance of 15 x 15 block coding scheme with other methods.

f. The histograms of these encoded images were very similar to those of 255 x 255 block size encoding and are therefore not included here.

g. Figure 24 shows the bit assignments in the transform domain for 15 x 15 block coding of Channel 1 image. The (15 x 15) block encoded images were not contrast enhanced and the corresponding original image is included in Figures 21 through 23 for comparison between the original and encoded images.

4. Other Coding Experiments

In previous experiments, uniform quantization of $\hat{x}_{ij}$ in the transform domain was performed. Further, constant correlation parameter values were used in each dimension, over all the image blocks. However, the value of the correlation parameters can vary over the different image regions. We examined the effect of these variations as well as the effect of different quantization schemes. The various experiments are as follows:



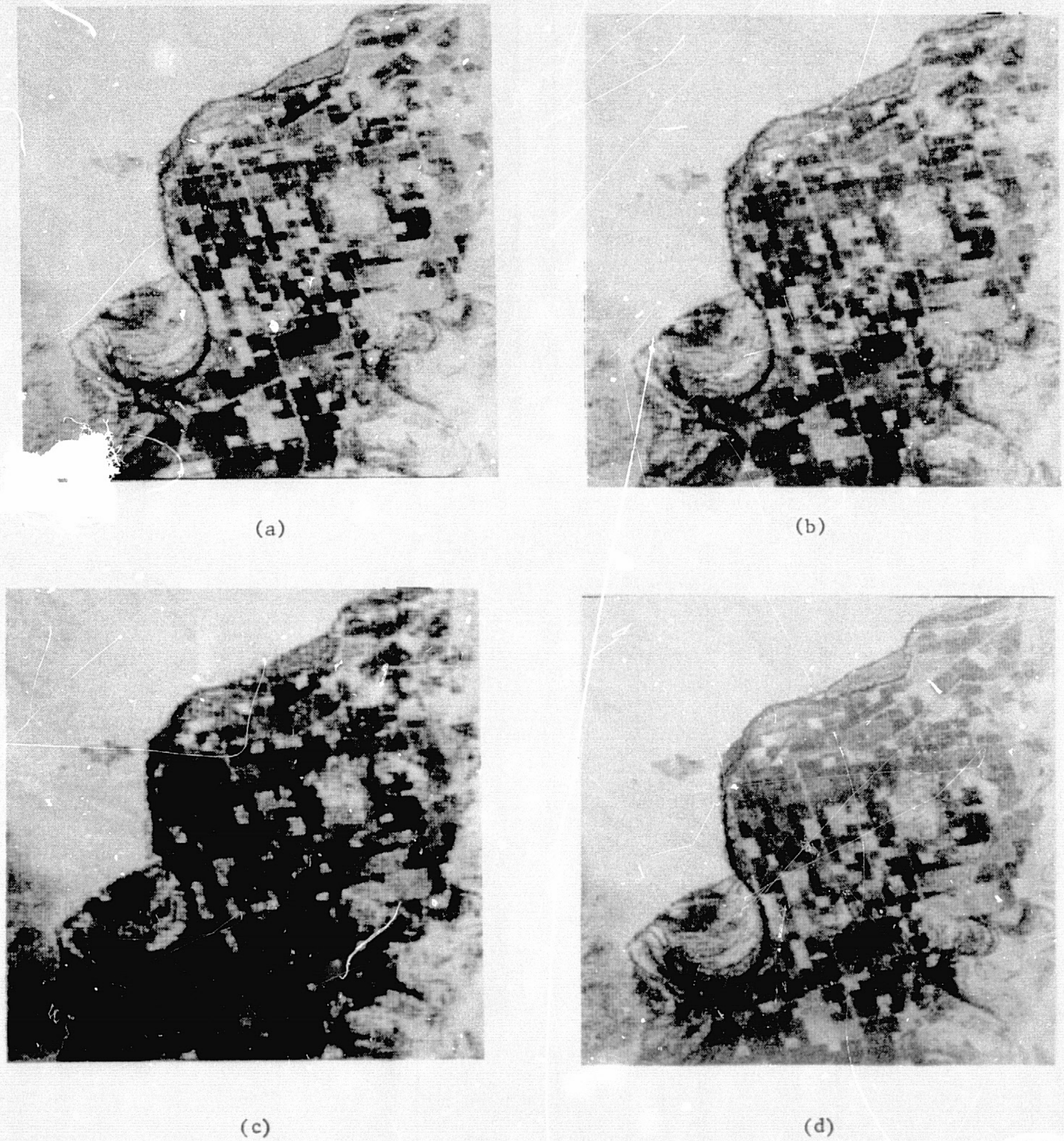
(a)

(b)

(c)

(d)

FIGURE 21 FAST KLT block by block encoded image of the Channel 1 Image
(a) 1.987 bits/pixel (b) 1.004 bits/pixel
(c) 0.471 bits/pixel (d) Original Image



(a)

(b)

(c)

(d)

FIGURE 22 FAST KLT block by block encoded image of the Channel 2 Image
(a) 2.009 bits/pixel (b) 1.058 bits/pixel
(c) 0.591 bits/pixel (d) Original Image

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

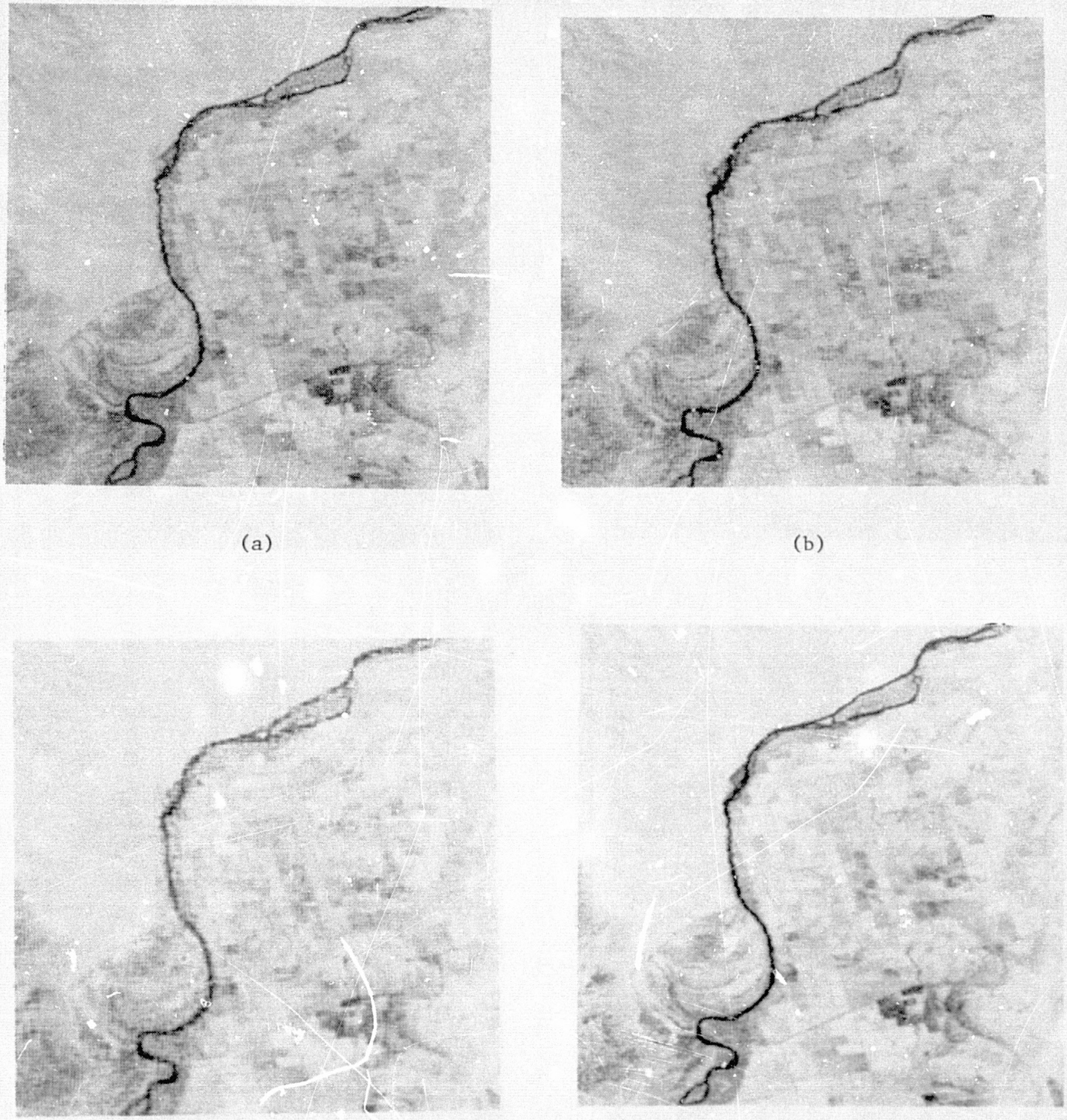


FIGURE 23 FAST KLT block by block encoded image of the Channel 3 Image
(a) 2.000 bits/pixel (b) 1.067 bits/pixel
(c) 0.591 bits/pixel (d) Original Image

Block by Block (15 x 15) Encoded Images												Average Over All Channels		
Channel 1			Channel 2			Channel 3			Channel 4					
Bit Rate	M.S.E.	SNR in db	Bit Rate	M.S.E.	SNR in db	Bit Rate	M.S.E.	SNR in db	Bit Rate	M.S.E.	SNR in db	Average Bit Rate	Average M.S.E.	Average SNR in db
1.996	2.5186	38.064	2.009	9.1307	32.471	2.000	5.1579	34.951	2.000	1.5523	34.077	2.00125	4.590	34.891
1.031	5.9991	34.295	1.058	17.3456	29.684	1.067	10.5176	31.857	1.066	3.0057	31.207	1.0555	9.217	31.761
0.542	10.5002	31.964	0.569	27.4262	27.694	0.591	16.0689	30.016	0.591	4.6548	29.308	0.57325	14.663	29.721
0.102	21.9417	28.664	---	---	---	---	---	---	---	---	---	---	---	---

TABLE 3 M.S.E.: Mean square error

SNR in db: Signal* (p-p) to noise (rms) ratio in db scale

*p-p values are: 127 Channel 1, 2, 3

63 Channel 4

```
 7  7  6  6  5  5  5  4  4  4  4  4  4  4  4
 6  6  5  5  5  4  4  4  4  3  3  3  3  3  3
 6  5  5  4  4  4  3  3  3  3  3  3  2  2  2
 5  5  4  4  3  3  3  3  2  2  2  2  2  2  2
 5  4  4  3  3  3  2  2  2  2  2  1  1  1  1
 4  4  3  3  3  2  2  2  2  1  1  1  1  1  1
 4  4  3  3  2  2  2  1  1  1  1  1  1  1  1
 4  3  3  2  2  2  1  1  1  1  1  1  0  0  0
 3  3  3  2  2  1  1  1  1  1  0  0  0  0  0
 3  3  2  2  2  1  1  1  1  0  0  0  0  0  0
 3  3  2  2  1  1  1  1  0  0  0  0  0  0  0
 3  3  2  2  1  1  1  1  0  0  0  0  0  0  0
 3  3  2  2  1  1  1  0  0  0  0  0  0  0  0
 3  3  2  2  1  1  1  0  0  0  0  0  0  0  0
 3  2  2  2  1  1  1  0  0  0  0  0  0  0  0

AVERAGE BIT RATE

  .19867E+01
```

```
 7  6  6  5  5  4  4  4  3  3  3  3  3  3  2
 6  6  5  4  4  3  3  3  2  2  2  2  2  2  2
 5  5  4  3  3  2  2  2  1  1  1  1  1  1  1
 4  4  3  3  2  2  1  1  1  0  0  0  0  0  0
 4  3  3  2  1  1  1  0  0  0  0  0  0  0  0
 3  3  2  1  1  1  0  0  0  0  0  0  0  0  0
 3  2  2  1  1  0  0  0  0  0  0  0  0  0  0
 2  2  1  1  0  0  0  0  0  0  0  0  0  0  0
 2  2  1  0  0  0  0  0  0  0  0  0  0  0  0
 2  1  1  0  0  0  0  0  0  0  0  0  0  0  0
 2  1  1  0  0  0  0  0  0  0  0  0  0  0  0
 2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0

AVERAGE BIT RATE

  .10044E+01
```

```
 7  6  5  5  4  3  3  2  2  2  1  1  1  1  1
 6  5  4  3  3  2  1  1  1  0  0  0  0  0  0
 4  4  3  2  1  1  0  0  0  0  0  0  0  0  0
 3  3  2  1  0  0  0  0  0  0  0  0  0  0  0
 3  2  1  0  0  0  0  0  0  0  0  0  0  0  0
 2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

AVERAGE BIT RATE

  .47111E+00
```

```
 7  6  5  4  3  2  1  0  0  0  0  0  0  0  0
 5  4  3  2  1  0  0  0  0  0  0  0  0  0  0
 3  2  1  0  0  0  0  0  0  0  0  0  0  0  0
 2  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

AVERAGE BIT RATE

  .23556E+00
```

FIGURE 24 Bit Allocation Pattern for block by block (15x15) encoding of Channel 1 image with 4 different Bit rates

(4-1) Take the average value of horizontal and vertical correlation parameters of the entire 255 x 255 image as a constant correlation parameter value to be used in both dimensions.

(4-2) Use the actual horizontal and vertical calculated correlation parameter values of the entire image. (The image has different values of the correlation parameter in each direction).

(4-3) Divide the image into 15 x 15 small image blocks. Then calculate the vertical and horizontal correlation parameters of each small image block, and take the average over all image blocks. The average of these horizontal and vertical correlation parameters is taken as a single constant correlation parameter value for block by block coding.

(4-4) Here the average horizontal and vertical correlation parameters of each 15 x 15 small image block are taken as the horizontal and vertical correlation parameter values.

(4-5) For each 15 x 15 small image block its own horizontal and vertical correlation parameters are used. We call this 'variable ρ coding method,' or 'adaptive coding method' since the quantizer for each block will be different.

The probability density model of the transform domain variables $\hat{x}_{ij}$ is assumed to be the two sided exponential function

$$p(x) = ce^{-\alpha|x|} \quad ,$$

where

$$c = 1/(\int_{xMIN}^{xMAX} e^{-\alpha|x|}dx)$$

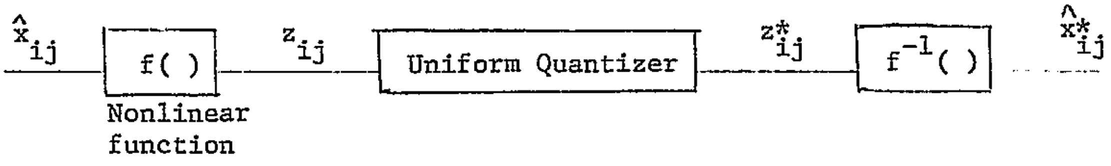
$$\alpha = \sqrt{\frac{2}{\sigma_x 2}} \quad ,$$

$\sigma_x 2$ = variance of x.

xMAX = maximum value of $\hat{x}_{ij}$

xMIN = minimum value of $\hat{x}_{ij}$ = - xMAX

In some of the experiments, a compandor design was used for quantizing the variables $\hat{x}_{ij}$. The compandor is implemented as shown in the accompanying figure.



The nonlinear fu tion f(x) is given by

$$f(x) = \begin{cases} \text{xMAX } [1 - \exp(-\frac{\alpha x}{3})]/[1 - \exp(\frac{\alpha \text{xMAX}}{3})] & x > 0 \\ f(|x|) \;, & x < 0 \end{cases}$$

The output x* is given by f^{-1} as

$$f^{-1}(z) = x^* = \begin{cases} -\frac{3}{\alpha} \log_e [1 - \frac{z}{\text{xMAX}} \{1 - \exp(-\frac{\alpha \text{xMAX}}{3})\}] & z > 0 \\ -f^{-1}(|z|) & z < 0 \end{cases}$$

In our experiments the values of xMAX and xMIN were chosen as follows

xMAX = x mean + 2.5 * SQRT (VAR)

xMIN = x mean - 2.5 * SQRT (VAR)

where

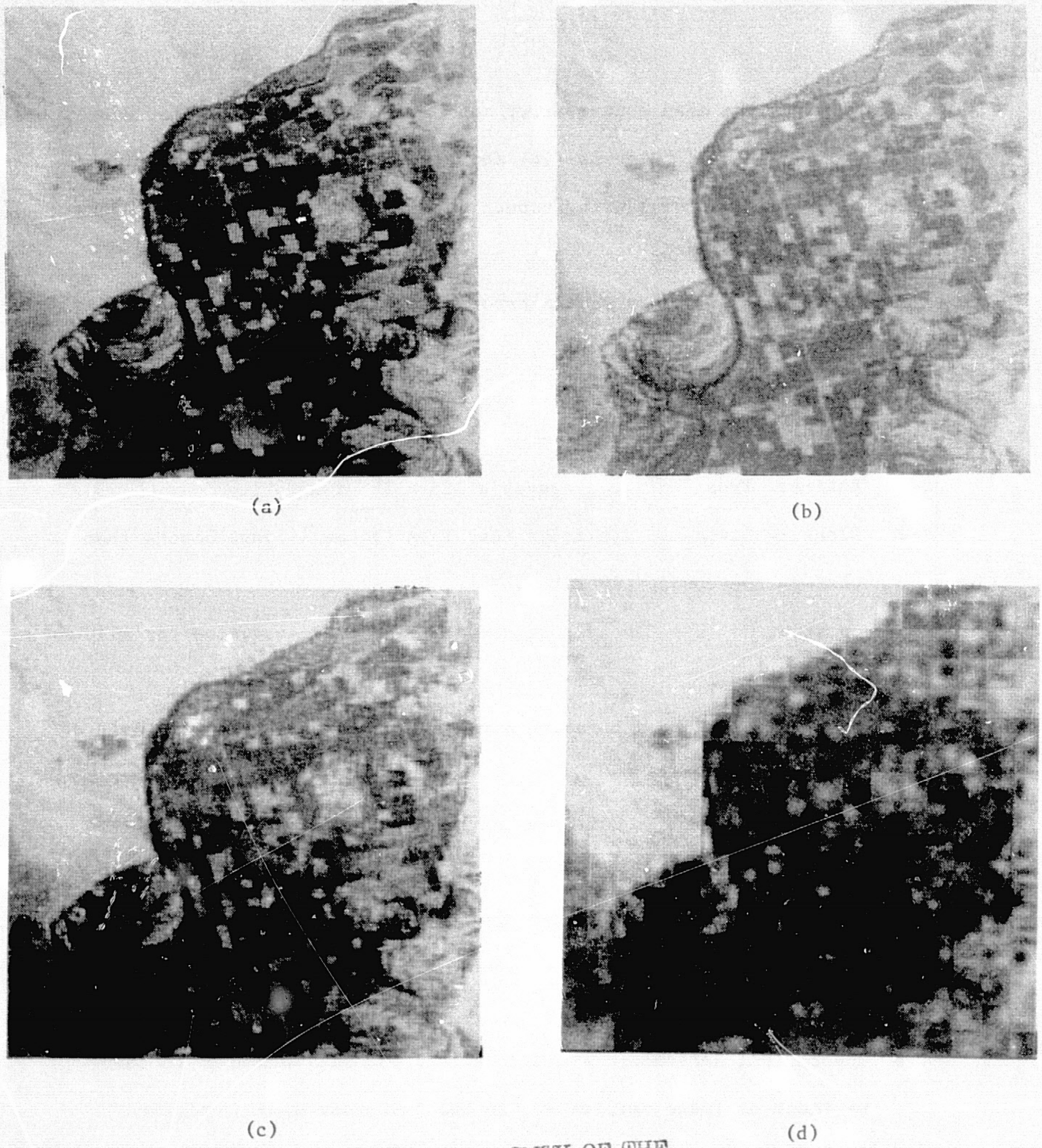
x mean = mean value of $\hat{x}_{ij}$ in the image block

VAR = variance of $\hat{x}_{ij}$

The value VAR can be obtained either by actual calculation or by the theoretical formula for $E[\hat{x}^2_{ij}]$ given in Chapter III.

Results of many experiments that were tried are recorded in Table 5 .

a. Figure (25) shows the block by block encoded channel 1 image with variable ρ in each block. These images also have not been contrast enhanced.



REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

FIGURE 25 FAST KLT adaptive block by block encoded image (with variable ρ in each block)of the Channel 1 Image (a) 2.010 bits/pixel (b) 1.076 bits/pixel (c) 0.616 bits/pixel (d) 0.126 bits/pixel

b. Table 4 lists the mean square error and signal to noise ratio for adaptive coding of Channel 1. Figures (14) and (15) show the performance of adaptive coder for Channel 1 with respect to 15 x 15 block coding and 255 x 255 entire image coding.

c. Table 5 compares the mean square error for 10 different coding experiments.

<u>Explanation of Table 5</u>

c-1 Entire: Treat entire image as a 255 x 255 image block.

c-2 Block : Divide entire image into 15 x 15 small image blocks then perform the coding algorithm.

c-3 Constant ρ: Use the average correlation parameter value for all image blocks.

c-4 Variable ρ: The correlation parameter value varies for different image block.

c-5 Uniform Quantizer: Uniform quantizer in the transform domain is used.

c-6 Compandor: Use compandor in the quantizer design.

c-7 Actual Variance: Use calculated variance of each image block to find xMAX and xMIN.

c-8 Theoretical Variance: Use theoretical formula to calculate the variance of image samples $\hat{x}_{ij}$ in the transform domain and then find xMAX and xMIN as mentioned before.

c-9 Bit Rate: Average bit rate for the entire image.

c-10 M.S. Error: Mean square error

Channel 1	Bit Rate	2.010	1.076	0.616	0.126
	M. S. E.	1.8534	5.0819	8.8548	22.3115
	SNR in db	39.3964	35.0159	32.6042	28.5907

TABLE 4: S/N ratio and M.S.E. vs. Bit Rate for adaptive coding of Channel 1 image.

M.S.E.: Mean square error

SNR in db: Signal (p-p) to noise (rms) ratio in db scale

Experimental Sequence / Items		1	2	3	4	5	6	7	8	9	10
Entire (255 x 255 block)										Yes	Yes
Block (15 x 15 block)		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
Constant ρ	ρ horizontal	0.7106	0.7106	0.7106	0.7106					0.93	0.93
	ρ vertical	0.5976	0.5976	0.5976	0.5976					0.93	0.93
Variable ρ						Yes	Yes	Yes	Yes		
Uniform Quantizer		Yes	Yes			Yes	Yes				
Compandor				Yes	Yes			Yes	Yes	Yes	Yes
Actual Variance		Yes		Yes		Yes		Yes		Yes	
Theoretical Variance			Yes		Yes		Yes		Yes		Yes
Bit Rate (bits/pixel)		1.0044	1.0044	1.0044	1.0044	0.9961	0.9961	0.9961	0.9961	1.00	1.00
M.S. Error		6.4697	6.0278	5.8556	5.1517	5.5427	5.4579	4.8839	5.2107	5.0305	14.2570

TABLE 5 Mean square error comparison of 10 different coding experiments for the earth data channel 1 image

*The box marked Yes means applicable, the empty box means not applicable.

5. Discussion and Comparsions

Complexity of the Fast KL Algorithm

The computational complexity of the fast KL algorithm in terms of number of computations is the same as that of cosine, Fourier and hybrid cosine/DPCM coding algorithms, i.e. $O(N^2 \log_2 N)$ computations are required for coding an N x N block of the image. The computer memory storage requirements are same as those of any two dimensional transform coding algorithm.

Performance Comparison with Respect Image Block Size

Figures 1 and 16 show the comparison between 15 x 15 and 255 x 255 image block size. See Tables 2 and 4 also. It is clear that w.r.t. both the mean square error and S/N ratio 255 x 255 image block size give better compression. On the average, the 255 x 255 block size has maximum advantage over 15 x 15 block size at higher bit rates. For example, at 2 bits/pixel the 255 x 255 block size gives 1.5 Db better performance than 15 x 15 block size. This difference becomes less significant at lower rates e.g., it is 1 Db at 1 bit/pixel and negligible below that rate. This difference is not unexpected because as the block size increases, larger redundancy can be removed.

Performance Comparison Among Different Channels

Figure (14) shows the variation of mean square error of channel image as a function of bit rate. Although the absolute m.s.e. values differ significantly, these differences are understood by studying the variances of the corresponding images (see Table 1). If the m.s.e. value of each encoded image (at a fixed bit rate) are normalized by the variance of the image in Table 1, then it is seen that the normalized m.s.e. are nearly equal for channels 1 and 2 and the same holds for channels 3 and 4 also. For example, at 1 bit/rate; for 15 x 15 block size the normalized m.s.e. values for channels 1,2,3 and 4 are .046, 0.051, 0.087, 0.075 respectively. With respect to m.s.e., then the four channels may be ranked as 1,2,4 and 3. However, on the basis of S/N ratio (see Figure 15) the ranking is 1,3,4, 2. This is because channels 1,2,3 images are 7 bit/pixel images giving a peak to peak value of 127 and channel 4 images are 6 bit/pixel giving a peak to peak value of 63. If the S/N ratio were defined in terms of the ratio of the variance of the original signal to the m.s.e., then the two criteria will give a consistent ranking of channels.

Comparisons With Other Coding Methods

Perhaps the most interesting (and surprising) results of this study are the comparisons of the fast KL coding algorithm with other coding experiments performed by TRW, Inc. (see [20]) on the same data. Figure 1 shows the performance of the fast KL transform coding algorithms vs. other methods. It is clear that the 2 dimensional fast KL algorithm (255x255) as well as (15x15) dominate the other methods including the 3 dimensional methods viz KL (temporal domain)-Cosine/DPCM and KL-Hadamard/DPCM. However, these curves have to be observed with certain caution, First, the mean square error is the average mean square error over the four channel images. A more meaningful comparison could be to compare the average of the normalized m.s.e. of each channel, where the normalized m.s.e. is the ratio of m.s.e. of a channel image and the variance of the original image signal. Also, the various methods should be compared on each individual channel as well. Unfortunately, the m.s.e. (due to other methods) per channel is not available in [20] and hence these comparisons cannot be reported here.

Figure 16 shows the comparisons with respect to average S/N ratio. The 255x255 fast KL coding scheme performs better than the other schemes. The 15 x 15 block fast KL coding performs better than 2-dimensional cosine as well as Cosine/DPCM at 1 bit pixel by more than 1 Db. At higher bit rate, e.g. 2 bits/pixel, the difference between 15x15 fast KL and 2-dimensional cosine is as much as 2.5 Db. This difference is quite surprising. The cosine/DPCM and 15x15 fast KL have nearly equal performance at 2 bits/pixel.

The difference in the performance of fast KL vs. Cosine may partly be attributed to the different correlation parameter values used. For 15x15 block coding we have used lower values (see Table 1, $\rho \simeq .6$ to .7) whereas in the TRW study ρ was taken $\simeq$ .9. This difference will cause the bit assignment in the transform domain to be different.

Another interesting comparison is noted when Figures 1 and 16 are compared. While 15x15 fast KL gives lower average mean square error (at 2 bits/pixel) than the 3-dimensional schemes, it does not give a higher average S/N ratio. This is because averaging the m.s.e. over all channels does not average the S/N ratio over these channels.

Adaptive Coding

In the adaptive coding method we changed the bit assignment from block to block according to the variations in the correlation parameters, keeping the average bit rate of each block fixed. The improvement of this adaptive coding scheme is only marginal over the non adaptive fast KL. However, a better performance can be expected if the average bit rate per block is varied according to its variance and the bit assignment within each block is based according to the correlation parameters of that block.

6. Conclusion

1. To the extent of the usefulness of the average mean square error and average S/N ratio criteria, the fast KL transform coding algorithm gives superior performance as compared with other 2-dimensional coding methods for multispectral images.

2. More detailed comparisons are required to establish the usefulness of this algorithm from a user's point of view. These comparisons should include other criteria such as (i) the spatial distribution of errors in the reconstructed signal (ii) Classification accuracy based on ground truth. Also a larger set of images should be used to validate the conclusions.

3. The fact that correlation parameter values change significantly as the image block size is changed suggests non-stationarities in the images and fast adaptive algorithms could be useful. Since, the fast KL

transform basis vectors are invariant to changes in image statistics, and the transform domain variances are known in closed form, application of this method for adaptive coding is suggested.

4. Since the Cosine/DPCM algorithm has been shown to perform significantly better than 2-dimensional cosine [20] at high bit rates, an interesting question is, "How does fast KL/DPCM perform"?

REFERENCES

1. A. K. Jain, "A Fast Karhunen-Loeve Transform for Finite Discrete Images," *Proc. National Electronics Conference*, Chicago, Illinois, October 1974.

2. H. C. Andrews, *Computer Techniques in Image Processing*, Academic Press, New York, 1970.

3. W. H. Chen, "Slant Transform Image Coding," Ph.D. Thesis, University of Southern California, May 1973.

4. W. K. Pratt, "A Comparison of Digital Image Transforms," The Mervin J. Kelly Communication Conference, Univ. of Missouri, Rolla, October 1970.

5. A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," *IEEE Transactions on Communication Technology*, Vol. Com-19, No. 1, February 1971.

6. N. Ahmed, T. Natarajan, K. R. Rao, 'Discrete Cosine Transform', *IEEE Trans. Comp*., Vol. C-23, No. 1, January 1974, pp. 90-93.

7. P. A. Wintz, "Transform Picture Coding," *Proc. IEEE*, Vol. 60, No. 7, pp. 809-829, July 1972.

8. W. K. Pratt, "Karhunen-Loeve Transform Coding of Images," *Proc. 1970 IEEE Int. Symp. Inform. Theory*, 1970.

9. H. P. Kramer and M. V. Mathews, "A Linear Coding for Transmitting a Set of Correlated Signals," *IRE Trans. Information Theory*, Vol. IT-2, pp. 41-46, September 1956.

10. H. Hotelling, "Analysis of a Complex of Statistical Variables Into Principal Components," J. Educ. Psychology, Vol. 24, pp. 417-441, 498-520, 1933.

11. W. D. Ray and R. M. Driver, "Further Decomposition of the Karhunen-Loeve Series Representation of a Stationary Random Process," *IEEE Trans. Information Theory*, Vol. IT-16, No. 6, pp. 663-668, November 1970.

12. V. R. Algazi and D. J. Sakrison, "On the Optimality of the Karhunen-Loeve Expansion," *IEEE Trans. Information Theory* (corresp.) pp. 319-321.

13. S. Watanabe, "Karhunen-Loeve Expansion and Factor Analysis, Theoretical Remarks and Applications," *Trans. Fourth Prague Conf. Inform. Theory, Statist. Decision Functions and Random Processes*, Prague, 1965, pp. 635-660.

14. A. K. Jain, "Image Coding via a Nearest Neighbors Image Model," *IEEE Trans. Communications*, Vol. COM-23, No. 3, March 1975, pp. 318-331.

15. A. K. Jain and E. Angel, "Image Restoration, Modelling and Reducation of Dimensionality," IEEE Trans. on Computers, Vol. C-23, No. 5, May 1974.

16. I. J. Good, "The Interaction Algorithm and Practical Fourier Analysis," J. Royal Statistical Soc., (London) B20, 361 (1958).

17. E. O. Brigham: The Fast Fourier Transform, Prentice Hall, New Jersey, 1974.

18. A. K. Jain, "A Fast Karhunen-Loeve Transform for Recursive Filtering of Images Corrupted by White and Colored Noise," (to appear).

19. A.K. Jain, "Image Restoration by Operator Factorization of Point Spread Function and Image Covariance Function," Dept. of Electrical Engineering SUNYAB Report 75-001, Oct. 1975.

20. A.Habibi, Study of On-Board Compression of Earth Resources Data, Final Report, Contract No. NAS2-8394, TRW No. 26566, TRW Systems Group, Redondo Beach, Calif. 90278, Sept. 1975.

21. J.W. Woods, "Two Dimensional Discrete Markovian Fields," IEEE Trans. Information Theory, Vol. IT-18, No. 2, March 1972, pp. 232-240.

APPENDIX I

CORRELATION PROPERTIES OF THE MINIMUM VARIANCE REPRESENTATION

The equations of representation of the first order stationary sequence $\{x_i\}$ with zero mean and autocorrelation

$$E[x_i x_j] = \rho^{|i-j|} \tag{1}$$

are given by

$$x_i = \alpha(x_{i+1} + x_{i-1}) + \nu_i \qquad i = 1,\ldots,N \tag{2}$$

$$x_0 = \rho x_1 + \nu_0, \quad x_{N+1} = \rho x_N + \nu_{N+1} \tag{3}$$

where $\alpha = \rho/(1+\rho^2)$. Multiplying (2) by x_{i+k}, taking expectations and using (1) we get

$$\rho^{|k|} = \alpha(\rho^{|k-1|} + \rho^{|k+1|}) + E[\nu_i x_{i+k}] \tag{4}$$

which for $|k| \geq 1$ and with $\alpha = \rho/(1+\rho^2)$, gives

$$E[\nu_i x_{i+k}] = 0, \quad k \neq 0 \quad 1 \leq i \leq N \tag{5}$$

Similarly for $k = 0$, $E[x_i \nu_i] = (1-\rho^2)/(1+\rho^2) \triangleq \beta^2; \quad 1 \leq i \leq N$ (6)

Multiplication of (2) by ν_{i+k}, taking expectations and use of (5) and (6) yields

$$E(\nu_i \nu_{i+k}] = -\alpha[\beta^2 \delta_{k,1} + \beta^2 \delta_{k,-1}] + \beta^2 \delta_{k,o}$$

$$= \alpha\beta^2(\delta_{k,1} + \delta_{k,-1}) + \beta^2 \delta_{k,o} \tag{7}$$

where δ_{ij} is the Kronecker delta function. Similar procedure when applied to equations (3) gives

$$E[\nu_0 x_k] = (1-\rho^2)\delta_{k,0} \quad ; \quad E[\nu_0 \nu_k] = (1-\rho^2)\delta_{k,0} - \alpha(1-\rho^2)\delta_{k,1} \quad , \tag{8}$$

and

$$E[\nu_{N+1} x_k] = (1-\rho^2)\delta_{k,N+1} \quad ; \quad E[\nu_{N+1} \nu_k] = (1-\rho^2)\delta_{k,N+1} - \alpha(1-\rho^2)\delta_{k,N} \quad . \tag{9}$$

APPENDIX II

A TWO DIMENSIONAL IMAGE REPRESENTATION

Let $\{u_{ij},\ i, j = 0,1,\ldots,N,N+1\}$ be a two dimensional stationary sequence with zero mean and a separable autocorrelation function given by

$$E[u_{ij}u_{i+n,j+m}] = \rho_1^{|n|}\rho_2^{|m|}\ , \text{ Let } \rho_1 = \rho_2 \tag{1}$$

Let $\bar{u}_{ij}$ be the best linear, mean square, estimate of u_{ij} obtained from all u_{mn} not including the point u_{ij}. This is obtained by first writing

$$\bar{u}_{ij} = \sum_{\substack{k \\ k+\ell\neq 0}} \sum_{\ell} a_{k\ell}(u_{i+k,j+\ell}+u_{i-k,j-\ell}+u_{i-k,j-\ell}+u_{i-k,j+\ell}) \tag{2}$$

and minimizing $E[u_{ij}-\bar{u}_{ij})^2]$ over the coefficients $a_{k\ell}$. Differentiation of this expression with respect to $a_{k\ell}$, setting it equal to zero and after some algebraic manipulations we get for $1 \le (i,j) \le N$

$$\rho^{|k|}\rho^{|\ell|} = \sum_{k'} \sum_{\ell'} a_{k'\ell'}\ (\rho^{|k+k'|} + \rho^{|k-k'|})(\rho^{|\ell+\ell'|}+\rho^{|\ell-\ell'|})\quad , \tag{3}$$

for all $k + \ell \neq 0$, $k' + \ell' \neq 0$. From (3) it can be proven $a_{01} = \rho/(1+\rho^2) = a_{10}$; $a_{11} = a_{01}^2$ and $a_{k\ell} = 0$ for $k^2 + \ell^2 \ge 4$, so that the two dimensional representation equation defined as $u_{ij} = \bar{u}_{ij} + \nu_{ij}$, becomes

$$\begin{aligned} u_{ij} &= \alpha(u_{i+1,j}+u_{i,j+1}+u_{i,j-1}+u_{i-1,j}) \\ &\quad - \alpha^2(u_{i+1,j-1}+u_{i+1,j+1}+u_{i-1,j+1}+u_{i-1,j-1}) + \nu_{ij} \end{aligned} \tag{4}$$

where $\alpha = \rho/(1+\rho^2)$. Eqn. (4) can be rearranged to give

$$u_{ij} - \alpha(u_{i+1,j}+u_{i-1,j}) = v_{ij} \tag{5}$$

$$v_{ij} - \alpha(v_{i,j+1}+v_{i,j-1}) = \nu_{ij} \quad \text{for} \quad 1 \le (i,j) \le N \tag{6}$$

Following Appendix I, the statistical properties of v_{ij} and ν_{ij} are obtained, for $1 \le (i,j) \le N$, as

$$E[v_{ij}] = 0 = E[\nu_{ij}] \tag{7}$$

$$E[v_{ij}v_{k\ell}] = \beta^2 \rho^{|j-\ell|}(\delta_{ik}-\alpha\delta_{i+1,k}- \alpha\delta_{i-1,k}) \tag{8}$$

$$E[\nu_{ij}\nu_{k\ell}] = \beta^4(\delta_{ik}-\alpha\delta_{i+1,k}-\alpha\delta_{i-1,k})(\delta_{j\ell}-\alpha\delta_{j+1,\ell}-\alpha\delta_{j-1,\ell}) \tag{9}$$

where

$$\beta^2 = (\frac{1-\rho^2}{1+\rho^2}) \quad .$$

At boundary points (i=0,N+1;j=0,N+1) the minimum variance representation equations are somewhat different. These equations are not given since they are not required for the derivation of the fast KL transform result. Similar results are obtained when $\rho_1 \neq \rho_2$ in (1) above.

APPENDIX III

USAGE OF COMPUTER PROGRAMS

Two computer programs written in FORTRAN IV have been developed. These are:

a. IMAGE ANALYSIS PROGRAM

b. FAST KL TRANSORM CODING PROGRAM

These programs have been implemented on the NASA/MSFC IBM 360/65 and NASA/Ames CDC 7600 computers. This chapter explains the usage of these programs.

1. IMAGE ANALYSIS PROGRAM

The MAIN routine of this part of computer program does the following jobs:

a. Print out the histogram and statistical parameters of the image

b. Calculate the horizontal and vertical correlation parameters over all image blocks

c. List a pseudo bit rate constant versus actual bit rate average table

The subroutines used in this program are:

a. KAR - This subroutine finds the maximum, minimum, and mean for each image block, creates zero mean image block and calculates ρ_{10}, ρ_{01} and bit allocation for each pixel.

b. HISTO-Prints out the maximum, minimum, mean, standard deviation and the histogram of the 255 x 255 image.

c. CORRE-Calculates the horizontal and vertical correlation parameters of 255 x 255 image.

The definition of all the parameters used in these subroutines have been clarified in the computer listings.

2. PROGRAM USER'S GUIDE OF IMAGE ANALYSIS PROGRAM

A 256 x 256 image is loaded on a standard magnetic tape. In order to process via the Fast KL Transform technique, the actual dimension of the image used is 255 x 255, because if N is the image size N+1 must be a power of 2 to implement the sine transform.

2.1 PROGRAM INPUT

The JCL cards of the image are shown here as a reference for the user.

```
//GO.FT01F001  DD  DSN=INPT1, UNIT=TAPE 9, VOLUME=(PRIVATE,,SER=A0001),
//    DCB=(DEN=2, LRECL=512, BLKSIZE=516, RECFM=VS), LABEL=(1, BLP,, IN),
//    DISP=(OLD, DELETE)
```

This means the image is loaded in the following format

a. 9-track magnetic tape

b. 800 BPI (bits per ir. h)

c. Integer* 2 (2 logical   ds to represent one integer pixel data)

There is one input data card and it is:

```
$INPUT  IFRAME=1, N=15, N11=7, ILINE=17, $END
```

2.2 PROGRAM OUTPUT

Output listing of this program includes

a. Histogram of 255 x 255 entire image

b. Horizontial and vertical correlation parameters of 255 x 255 entire image

c. Print out a table of pp(I) and ACURAT(I), which allows the user to pick any desired bit rate of the image to be used later in the Fast KL Transform coding program.

3. FAST KL TRANSFORM CODING PROGRAM

This computer program does the following jobs.

a. Creates differential image of a 15 x 15 image block, $\{\nu_{ij}\}$.

b. Applies Fast KL Transform to $\{\nu_{ij}\}$.

c. Calculates bit assignments to different elements in the transform domain.

d. Performs quantization.

e. Applies inverse Fast KL Transform.

f. Stores final result as a 255 x 255 image on magnetic tape.

g. Prints out the Histogram and the results of analysis of the encoded image.

The subroutines are:

a. KAR -Creates a zero mean image and computes bit allocation pattern of a 15 x 15 image block. This subroutine is linked to subroutines CODE, QUANT, RECON.

b. CODE-Creates a differential image of a 15 x 15 image block, calculate the mean, variance, standard deviation of the transform domain 15 x 15 image block. This subroutine also calls subroutine XFORM.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

The definitions of IFRAME, N and N11 are given in program listing, parameters PAIO, PAOI, and PCON are obtained form the Image Analysis Program. PCON has a same meaning as PP, the pseudo bit rate. In our example PCON = 0.125, was picked from PP(96), and the corresponding value of ACURAT (96) = 0.99607837 which almost equals the one bit/pixel average rate.

In this program, two magnetic tapes are used, one for the original 255 x 255 image, the other for storing the final 255 x 255 encoded image. The JCL format cards are shown here for the user's reference.

Tape 3 (original 255 x 255 image)

```
//GO.FT03F001  DD  DSN=INPT1, UNIT=TAPE 9, VOLUME=(PRIVATE,,SER=A0001),
// DCB=(DEN=2, LRECL=512, BLKSIZE=516, RECFM=VS), LABEL=(1, BLP,,IN),
```

Tape 4 (final 255 x 255 encoded image)

```
//GO.FT04F001  DD  DSN=OUPT1, UNIT=TAPE 9, VOLUME=(PRIVATE,,SER=SAVE),
// DCB=(DEN=2, LRECL=1024, BLKSIZE=1028, RECFM=VS), LABEL=(1, BLP,,OUT),
// DISP=(NEW, DELETE)
```

c. QUANT-Performs quantization on the transform domain 15 x 15 block image samples $\hat{X}_{ij}$.

d. XFORM-Initializes and rearranges the data to call subroutine HARM and implements the Fast KL Transform on a 15 x 15 image block.

e. RECON-Reconstructs the 15 x 15 encoded image block from transform domain quantized 15 x 15 image block.

f. HARM--An FFT subroutine which is available in the IBM scientific subroutine package.

g. HISTO-Calculates the statistics and mean square error of 255 x 255 final encoded image and prints out the output histogram of the same image.

4. PROGRAM USER'S GUIDE OF FAST KL TRANSFORM CODING PROGRAM

4.1 PROGRAM INPUT

First select the actual or desired bit rate for encoding the image (say 1 bit/pixel). Then the input data card for 1st channel image, 15 x 15 block coding, and horizontal and vertical correlation values of 0.71, 0.598 respectively is;

```
$INPUT  IFRAME=1, N=15, N11=7, PA10=0.71059489, PA01=0.59758818, PCON=0.125, $END
```

4.2 PROGRAM OUTPUT

Output listing of this program is simple and includes

a. The actual bit rate of 255 x 255 encoded image.

b. Output histogram of 255 x 255 encoded image, various statistical parameter, the mean square error in encoding, and the signal to noise ratio of the encoded image.

The listing of subroutine HARM is not given in this report, since this is a standard program available in the IBM scientific subroutine package.

APPENDIX IV

IMAGE ANALYSIS PROGRAM

```
C
C     THIS PROGRAM DOES FOLLOWING JOBS:
C     1. PRINT OUT ORIGINAL IMAGE HISTOGRAM AND ANALYSIS
C     2. CALCULATE HORIZONTAL AND VERTICAL CORRELATION PARAMETERS OF
C        ENTIRE IMAGE
C     3. FIND SUITABLE PSEUDO BIT RATE CONSTANT FOR DIFFERENT BIT
C        RATE ASSIGNMENT
C
C     IU     : HORIZONTAL INPUT DATA STRING
C     P10    : HORIZONTAL CORRELATION PARAMETER OF AN IMAGE BLOCK
C     P01    : VERTICAL CORRELATION PARAMETER OF AN IMAGE BLOCK
C     PA10   : AVERAGE OF HORIZONTAL CORRELATION PARAMETER OVER ALL
C              IMAGE BLOCKS
C     PA01   : AVERAGE OF VERTICAL CORRELATION PARAMETER OVER ALL
C              IMAGE BLOCKS
C     ACURAT: ACTURAL AVERAGE BIT RATE
C     IB     : 15 BY 15 BLOCK IMAGE
C     AMEAN  : MEAN OF BLOCK IMAGE
C     VAR    : VARIANCE OF BLOCK IMAGE
C     PP     : PSEUDO BIT RATE CONSTANT
C     XMAX   : MAXIMUM VALUE OF ENTIRE IMAGE
C     XMIN   : MINIMUM VALUE OF ENTIRE IMAGE
C     XMEAN  : MEAN OF ENTIRE IMAGE
C     XVAR   : VARIANCE OF ENTIRE IMAGE
C     RANGE  : MAXIMUM VALUE MINUS MINIMUM VALUE
C     TOTAL  : SUM OF TOTAL VALUES
C     SD     : STANDARD DEVIATION OF ENTIRE IMAGE
C     R10    : HORIZONTAL CORRELATION PARAMETER OF ENTIRE IMAGE
C     R01    : VERTICAL CORRELATION PARAMETER OF ENTIRE IMAGE
C
      DIMENSION IB(255,15),ACURAT(400)
      COMMON/ANALYS/XMAX,XMIN,XMEAN,XVAR,TOTAL
      COMMON/SPACE/IU(256)
      COMMON/THING/PA10,PA01,ACR(400)
      COMMON/GROUP/U(16,16),PP(400),N,N11,ALAMDA(15,15)
      INTEGER*2 IU
      NAMELIST/INPUT/IFRAME,N,N11,ILINE
      NAMELIST/OUTPUT/PA10,PA01
  110 FORMAT(1X,*PP(*,I3,*)=*,F12.6,5X,*ACURAT(*,I3,*)=*,F12.8)
  111 FORMAT(/////)
      READ(5,INPUT)
      WRITE(6,INPUT)
      PA10=0.
      PA01=0.
      XMAX=0.
      XMIN=256.
      XMEAN=0.
      XVAR=0.
      ALINE=ILINE
      AN=N
      CONST=17.*ALINE
      DO 20 I=1,16
      DO 20 J=1,16
      U(I,J)=0.
```

```
20 CONTINUE
   DO 21 I=1,400
   ACR(I)=0.
21 CONTINUE
   PP(1)=2.5


   DO 22 I=2,400
   PP(I)=PP(I-1)-0.025
22 CONTINUE
   DO 10 IC=1,ILINE
   DO 11 J=1,15
   READ(1) (IU(K),K=1,256)
   DO 12 I=1,255
   IB(I,J)=IU(I)
12 CONTINUE
11 CONTINUE
   DO 13 I=1,17
   IS=(I-1)*15+1
   ISP=IS+14
   DO 14 J=1,15
   DO 14 IA=IS,ISP
   II=IA-IS+1
   U(II,J)=IB(IA,J)
14 CONTINUE
   CALL KAR
13 CONTINUE
10 CONTINUE
   RA10=RA10/CONST
   RAC1=RAC1/CONST
   DUM=AN*AN*CONST
   TOTAL=XMEAN
   XMEAN=XMEAN/DUM
   XVAR=XVAR/DUM-XMEAN*XMEAN
   DO 50 JA=1,400
   ACURAT(JA)=ACR(JA)/DUM
50 CONTINUE
   REWIND 1
   CALL HISTO
   REWIND 1
   CALL CORRE
   WRITE(6,111)
   WRITE(6,OUTPUT)
   DO 25 I=1,400
   WRITE(6,110) I,PP(I),I,ACURAT(I)
25 CONTINUE
   STOP
   END
```

```
      SUBROUTINE KAR
      COMMON/ANALYS/XMAX,XMIN,XMEAN,XVAR,TOTAL
      COMMON/GROUP/U(16,16),PP(400),N,N11,ALAMDA(15,15)
      COMMON/THING/PA10,PA01,ACR(400)
      SUM1=0.
      SUM2=0.
      DO 10  J=1,N
      DO 10  I=1,N
      SUM1=SUM1+U(I,J)
      SUM2=SUM2+U(I,J)*U(I,J)
      IF(U(I,J)  .GT.  XMAX)  XMAX=U(I,J)
      IF(U(I,J)  .LT.  XMIN)  XMIN=U(I,J)
   10 CONTINUE
      AMEAN=SUM1/(N*N)
      XMEAN=XMEAN+SUM1
      XVAR=XVAR+SUM2
      SUM=0.
      DO 11  J=1,N
      DO 11  I=1,N
      U(I,J)=U(I,J)-AMEAN
      SUM=SUM+U(I,J)*U(I,J)
   11 CONTINUE
      VAR=SUM/(N*N)


      P10=0.
      P01=0.
      DO 21  J=1,N
      DO 21  I=1,N
      P10=P10+U(I,J)*U(I+1,J)
      P01=P01+U(I,J)*U(I,J+1)
   21 CONTINUE
      P10=P10/(VAR*N*N)
      P01=P01/(VAR*N*N)
      PA10=PA10+P10
      PA01=PA01+P01
      ALP1=P10/(1.+P10*P10)
      ALP2=P01/(1.+P01*P01)
      ANSQ=N*N
      AA=3.1415926/(N+1)
      SUM=0.
      DO 50  I=1,N
      DO 50  J=1,N
      DUM=(1.-2.*ALP1*COS(I*AA))*(1.-2.*ALP2*COS(J*AA))
      ALAMDA(I,J)=DUM
      SUM=SUM+ALOG10(DUM/ALAMDA(1,1))
   50 CONTINUE
```

```
      DEN=SUM
      A11=N11
      DO 53 JA=1,400
      PCON=PP(JA)
      DUM=ANSC*(A11-PCON)/DEN
      NSUM=0
      DO 54 J=1,N
      DO 54 I=1,N
      AR=ALAMDA(I,J)
      NADUM=A11-DUM*ALOG10(AR/ALAMDA(1,1))+0.5
      IF(NADUM .GT. N11) NADUM=N11
      IF(NADUM .LT. 0) NADUM=0
      NSUM=NADUM+NSUM
   54 CONTINUE
      ACR(JA)=ACR(JA)+NSUM
   53 CONTINUE
      RETURN
      END
      SUBROUTINE HISTO
      DIMENSION X(256),ICT(64),TEMP(65),SYMBOL(100),IGRAPH(10),BAR(132)
      COMMON/ANALYS/XMAX,XMIN,XMEAN,XVAR,TOTAL
      COMMON/SPACE/IU(256)
      INTEGER*2 IU
      DATA ICT/64*0/,TEMP/65*0./,IGRAPH/10*0/,STAR/1H*/,BLANK/1H /,
     *DASH/1H-/,PLUS/1H+/
  100 FORMAT(/////)
  101 FORMAT(1H0)
  103 FORMAT(1X,8HMAXIMUM=,E12.5,1X,8HMINIMUM=,E12.5,1X,6HRANGE=,E12.5,1
     *X,6HTOTAL=,E12.5,1X,5HMEAN=,E12.5,1X,19HSTANDARD DEVIATION=,E12.5)
  104 FORMAT(1X,6HPSEUDO,5X,5HLOWER,5X,11HOCCURRENCES,20X,21HEACH STAR R
     *EPRESENTS ,I3,12H OCCURRENCES,/1X,5HLEVEL,6X,5HLEVEL)
  105 FORMAT(31X,10(5X,I5))
  106 FORMAT(1X,132A1)
  107 FORMAT(3X,I2,2X,E13.5,3X,I6,1X,1HV,100A1)
      RANGE=XMAX-XMIN
      SD=SQRT(XVAR)
      WRITE(6,100)
      WRITE(6,103) XMAX,XMIN,RANGE,TOTAL,XMEAN,SD
      XINC=(XMAX-XMIN)/64.
      TEMP(1)=XMIN
      DO 3 I=1,255
      READ(1) (IU(J),J=1,256)


      DO 18 J=1,256
      X(J)=IU(J)
   18 CONTINUE
      DO 4 J=1,255
      DO 5 K=1,64
      TEMP(K+1)=TEMP(K)+XINC
      IF(X(J)-TEMP(K+1)) 6,6,5
    5 CONTINUE
    6 ICT(K)=ICT(K)+1
    4 CONTINUE
    3 CONTINUE
      ICTMAX=0
      DO 7 I=1,64
      IF(ICT(I).GT.ICTMAX) ICTMAX=ICT(I)
    7 CONTINUE
      NU=ICTMAX/100.+1.
      WRITE(6,101)
      WRITE(6,104) NU
      DO 13 I=1,10
      IGRAPH(I)=I*NU*10
   13 CONTINUE
      WRITE(6,101)
      WRITE(6,105) IGRAPH
      DO 14 I=1,132
      BAR(I)=DASH
   14 CONTINUE
```

```
      DO 16 I=30,130,10
      BAR(I)=PLUS
16    CONTINUE
      WRITE(6,106) BAR
      DO 9 I=1,64
      IF(ICT(I).GE.NO) GO TO 9
      DO 10 K=1,100
      SYMBOL(K)=BLANK
10    CONTINUE
      GO TO 11
9     J=ICT(I)/NO
      J1=J+1
      DO 12 K=1,J
      SYMBOL(K)=STAR
12    CONTINUE
      DO 15 K=J1,100
      SYMBOL(K)=BLANK
15    CONTINUE
11    WRITE(6,107) I,TEMP(I),ICT(I),(SYMBOL(K),K=1,100)
8     CONTINUE
      WRITE(6,106) BAR
      WRITE(6,105) IGRAPH
      RETURN
      END
      SUBROUTINE CORRE
      DIMENSION A(256),B(256)
      COMMON/ANALYS/XMAX,XMIN,XMEAN,XVAR,TOTAL
      COMMON/SPACE/IU(256)
      INTEGER*2 IU
      DATA B/256*0./
  120 FORMAT(/////)
  121 FORMAT(1X,*HORIZONTAL CORRELATION PARAMETER OF ENTIRE IMAGE =*,
     *F12.8,9X,*VERTICAL CORRELATION PARAMETER OF ENTIFE IMAGE =*,F12.8)
      R10=0.
      R01=0.
      DO 19 I=1,255
      READ(1) IU
      DO 50 K=1,256
      A(K)=IU(K)
   50 CONTINUE
      DO 20 J=1,254
      R10=(A(J)-XMEAN)*(A(J+1)-XMEAN)+R10
   20 CONTINUE
      IF(I .GT. 1) GO TO 4
      GO TO 7
    4 DO 6 J=1,256
      R01=(B(J)-XMEAN)*(A(J)-XMEAN)+R01
    6 CONTINUE
    7 DO 33 J=1,256
      B(J)=A(J)
   33 CONTINUE
   19 CONTINUE
      R10=R10/(XVAR*255.*255.)
      R01=R01/(XVAR*255.*255.)
      WRITE(6,120)
      WRITE(6,121) R10,R01
      RETURN
      END
```

APPENDIX V

FAST KARHUNEN LOEVE TRANSFORM COMPUTATION LISTING

```
C
C      THIS PROGRAM DOES FOLLOWING JOBS:
C      1. GET DIFFERENTIAL IMAGE
C      2. PERFORM FAST K. L. XFORM
C      3. CALCULATE BIT RATE ASSIGNMENT
C         ( USE CONSTANT,RHO, IN ORDER TO GET SAME BIT
C         ASSIGNMENT PATTERN FOR EACH IMAGE BLOCK )
C      4. APPLY UNIFORM QUANTIZER TECHNIQUE
C      5. PERFORM INVERSE FAST K. L. XFORM
C      6. STORE FINAL RESULT IMAGE ON TAPE
C      7. PRINT OUT OUTPUT ANALYSIS AND HISTOGRAM
C
C      IU     : HORIZONTAL INPUT DATA STRING
C      PCON   : PSEUDO BIT RATE CONSTANT
C      PA10   : AVERAGE OF HORIZONTAL CORRELATION PARAMETER OVER ALL
C               IMAGE BLOCKS
C      PA01   : AVERAGE OF VERTICAL CORRELATION PARAMETER OVER ALL
C               IMAGE BLOCKS
C      AIB    : 256 BY 15 IMAGE BLOCK
C      N11    : ASSUMED BIT RATE FOR THE MOST SIGNIFICANT PIXEL IN XFORM
C               DOMAIN, THIS VALUE IS CALCULATED FROM THEORETICAL FORMULA
C      ACURAT : ACTUAL AVERAGE BIT RATE
C      AMEAN  : MEAN OF ORIGINAL IMAGE BLOCK
C      VAR    : VARIANCE OF ORIGINAL IMAGE BLOCK
C      NLEV   : 15 BY 15 BIT PATTERN ARRAY
C      XMEAN  : MEAN OF XFORM DOMAIN IMAGE BLOCK
C      XVAR   : VARIANCE OF XFORM DOMAIN IMAGE BLOCK
C      XDEVI  : STANDARD DEVIATION OF XFORM DOMAIN IMAGE BLOCK
C      EMAX   : MAXIMUM CLIPPING LEVEL OF XFORM DOMAIN IMAGE BLOCK
C      EMIN   : MINIMUM CLIPPING LEVEL OF XFORM DOMAIN IMAGE BLOCK
C      XMAX   : MAXIMUM VALUE OF FINAL RESULT IMAGE
C      XMIN   : MINIMUM VALUE OF FINAL RESULT IMAGE
C      RANGE  : MAXIMUM VALUE MINUS MINIMUM VALUE
C      TOTAL  : SUM OF TOTAL VALUES
C      XYMEAN : MEAN OF FINAL RESULT IMAGE
C      SD     : STANDARD DEVIATION OF FINAL RESULT IMAGE
C      ERR1   : MEAN SQUARE ERROR
C      ERR2   : ROOT MEAN SQUARE ERROR
C      ERRDB1 : MEAN SQUARE ERROR IN DB SCALE
C      ERRDB2 : ROOT MEAN SQUARE ERROR IN DB SCALE
C      SN1    : SIGNAL TO NOISE RATIO ( RANGE VS ERR2 )
C      SN2    : SIGNAL TO NOISE RATIO ( SD VS ERR2 )
C      SNDB1  : SN1 IN DB SCALE
C      SNDB2  : SN2 IN DB SCALE
C
       DIMENSION AIB(256,15)
       COMMON/ANALYS/IU(256),X(256)
       COMMON/GROUP/U(17,17),PCON,N11,N,PA10,PA01,ALP1,ALP2,
      *NLEV(15,15),ALAMDA(15,15),VAR,V(15,15),XH(15,15),
      *XHS(15,15),USTAR(15,15),USTARH(15,15)
       COMMON/RATE/NACK,IBBW
       NAMELIST/INPUT/IFRAME,N,N11,PA10,PA01,PCON
       NAMELIST/OUTPUT/ACURAT
       INTEGER*2 IU
       REAL*4 AIB
       REAL*4 X
      ASQ=255*255
      NAGR=0
      IBOW=0
      DO 40 J=1,17
      DO 40 I=1,17


      U(I,J)=0.
   40 CONTINUE
      READ(5,INPUT)
      WRITE(6,INPUT)
      DO 5 IC=1,17
      DO 11 J=1,15
      READ(3) (IU(I),I=1,256)
      DO 2 I=1,255
      AIB(I,J)=IU(I)
    2 CONTINUE
   11 CONTINUE
      DO 82 NII=1,17
      IS=1+15*(NII-1)
      ISP=IS+14
      DO 10 J=1,15
      JJ=J+1
      DO 20 I=IS,ISP
      II=I-IS+2
      U(II,JJ)=AIB(I,J)
   20 CONTINUE
   10 CONTINUE
      IBOW=IBOW+1
      CALL KAR
      DO 65 J=1,15
      DO 65 I=IS,ISP
      II=I-IS+1
      AIB(I,J)=USTARH(II,J)
   65 CONTINUE
   82 CONTINUE
      DO 63 J=1,15
      AIB(256,J)=0.
      WRITE(4) (AIB(I,J),I=1,256)
   63 CONTINUE
    5 CONTINUE
      DO 30 I=1,256
      AIB(I,1)=0.
   30 CONTINUE
      WRITE(4) (AIB(I,1),I=1,256)
      ACR=NAGR
      ACURAT=ACR/(15.*15.)
      WRITE(6,OUTPUT)
      REWIND 3
      REWIND 4
      CALL HISTO
      STOP
      END
      SUBROUTINE KAR
      COMMON/GROUP/U(17,17),PCON,N11,N,PA10,PA01,ALP1,ALP2,
     *NLEV(15,15),ALAMDA(15,15),VAR,V(15,15),XH(15,15),
     *XHS(15,15),USTAR(15,15),USTARH(15,15)
      COMMON/OUT/AMEAN
      COMMON/RATE/NACR,IRUN
      NP1=N+1
      NP2=N+2
      SUM=0.
      DO 10 J=2,NP1
      DO 10 I=2,NP1
      SUM=SUM+U(I,J)
   10 CONTINUE
      AMEAN=SUM/(N*N)
      SUM=0.
      DO 11 J=2,NP1
      DO 11 I=2,NP1
      U(I,J)=U(I,J)-AMEAN


      SUM=SUM+U(I,J)*U(I,J)
   11 CONTINUE
      VAR=SUM/(N*N)
      ALP1=PA10/(1.+PA10*PA10)
      ALP2=PA01/(1.+PA01*PA01)
      ANSQ=N*N
      AA=3.1415926/(N+1)
      IF(IRUN .GT. 1) GO TO 30
      SUM=0.
      DO 50 I=1,N
      DO 50 J=1,N
      DUM=(1.-2.*ALP1*COS(I*AA))*(1.-2.*ALP2*COS(J*AA))
      ALAMDA(I,J)=DUM
      SUM=SUM+ALOG10(DUM/ALAMDA(1,1))
   50 CONTINUE
      DEN=SUM
      A11=N11
      DUM=ANSQ*(A11-PCON)/DEN
      DO 54 J=1,N
      DO 54 I=1,N
      AR=ALAMDA(I,J)
      NADUM=A11-DUM*ALOG10(AR/ALAMDA(1,1))+0.5
      IF(NADUM .GT. N11) NADUM=N11
      IF(NADUM .LT. 0) NADUM=0
      NACR=NADUM+NACR
      NLEV(I,J)=NADUM
   54 CONTINUE
   30 CALL CODE
      CALL QUANT
      CALL RECON
      RETURN
      END
      SUBROUTINE CODE
      COMMON/GROUP/U(17,17),PCON,N11,N,PA10,PA01,ALP1,ALP2,
     *NLEV(15,15),ALAMDA(15,15),VAR,V(15,15),XH(15,15),
     *XHS(15,15),USTAR(15,15),USTARH(15,15)
      COMMON/OTHER/XMEAN,XVAR,XDEVI
      DIMENSION AF(15,15),BF(15,15)
      DO 10 JJ=1,N
      J=JJ+1
      DO 20 II=1,N
      I=II+1
      V(II,JJ)=U(I,J)-ALP1*(U(I+1,J)+U(I-1,J))-ALP2*(U(I,J+1)+U(I,J-1))
     *+ALP1*ALP2*(U(I+1,J-1)+U(I+1,J+1)+U(I-1,J+1)+U(I-1,J-1))
      AF(II,JJ)=V(II,JJ)
   20 CONTINUE
   10 CONTINUE
      NC=N
      CALL XFORM(NC,AF,BF)
      DO 30 J=1,N
      DO 30 I=1,N
      DEMN=SQRT(1./ALAMDA(I,J))
      VH=BF(I,J)
      XH(I,J)=DEMN*VH
   30 CONTINUE
      XMEAN=0.
      DO 50 J=1,N
      DO 50 I=1,N
      XMEAN=XMEAN+XH(I,J)
   50 CONTINUE
      XMEAN=XMEAN/(N*N)
      TEMP1=PA10*PA10
      TEMP2=PA01*PA01
      XVAR=(1.-TEMP1)*(1.-TEMP2)*VAR/(1.+TEMP1)/(1.+TEMP2)


      XDEVI=SQRT(XVAR)
      RETURN
      END
      SUBROUTINE QUANT
      COMMON/GROUP/U(17,17),PCON,N11,N,PA1G,PA01,ALP1,ALP2,
     *NLEV(15,15),ALAMDA(15,15),VAR,V(15,15),XH(15,15),
     *XHS(15,15),USTAR(15,15),USTARH(15,15)
      COMMON/OTHER/XMEAN,XVAR,XDEVI
      EMAX=XMEAN+2.5*XDEVI
      EMIN=XMEAN-2.5*XDEVI
      DO 50 J=1,N
      DO 50 I=1,N
      NIJ=NLEV(I,J)
      IF(NIJ .LE. 0) GO TO 61
      M=2**NIJ
      Q=(EMAX-EMIN)/M
      V1=EMIN+Q/2.
      IX=(XH(I,J)-EMIN)/Q
      VHOLD=V1+IX*Q
      IF(XH(I,J) .GE. EMAX) VHOLD=EMAX-Q/2.
      IF(XH(I,J) .LE. EMIN) VHOLD=V1
      GO TO 59
   61 VHOLD=XMEAN
   59 XHS(I,J)=VHOLD
   50 CONTINUE
      RETURN
      END
      SUBROUTINE XFORM(N,ADUM,BDUM)
      DIMENSION ADUM(15,15),BDUM(15,15),DDUM(15,15),
     *ZDUM(64),M(3),INV(8),S(8)
      CC=SQRT(2./(N+1))
      IFSET=1
      M(1)=5
      M(2)=0
      M(3)=0
      NT=2*(N+1)
      DO 5 IST=1,2
      DO 5 I=1,N
      DO 20 J=1,NT
      JP=1+(J-1)*2
      JI=JP+1
      ZDUM(JP)=0.
      ZDUM(JI)=0.
      IF(J .GE. 2 .AND. J .LE. N+1) GO TO 19
      GO TO 20
   19 IF(IST .EQ. 1) AGG=ADUM(I,J-1)
      IF(IST .EQ. 2) AGG=DDUM(I,J-1)
      ZDUM(JP)=AGG
   20 CONTINUE
      CALL HARM(ZDUM,M,INV,S,IFSET,IFERR)
      DO 21 J=1,NT
      JP=1+(J-1)*2
      JI=JP+1
      IF(J .GE. 2 .AND. J .LE. N+1) GO TO 18
      GO TO 21
   18 ADD=CC*ZDUM(JI)
      IF(IST .EQ. 1) DDUM(J-1,I)=ADD
      IF(IST .EQ. 2) BDUM(J-1,I)=ADD
   21 CONTINUE
    5 CONTINUE
      RETURN
      END
      SUBROUTINE RECON
      COMMON/GROUP/U(17,17),FCON,N11,N,PA10,PA01,ALP1,ALP2,
     *NLEV(15,15),ALAMDA(15,15),VAR,V(15,15),XH(15,15),
     *XHS(15,15),USTAR(15,15),USTARH(15,15)
      COMMON/OTHER/XMEAN,XVAR,XDEVI
      COMMON/OUT/AMEAN
      DIMENSION AF(15,15),BF(15,15)
      DO 10 J=1,N
      DO 10 I=1,N
      DEMN=SQRT(1./ALAMDA(I,J))
      USTAR(I,J)=XHS(I,J)*DEMN
      AF(I,J)=USTAR(I,J)
   10 CONTINUE
      NO=N
      CALL XFORM(NO,AF,BF)
      DO 30 J=1,N
      DO 30 I=1,N
      USTARH(I,J)=BF(I,J)+AMEAN
   30 CONTINUE
      RETURN
      END
      SUBROUTINE HISTO
      DIMENSION ICT(64),TEMP(65),SYMBOL(100),IGRAPH(10),BAR(132),XX(256)
      COMMON/ANALYS/IU(256),X(256)
      DATA ICT/64*0/,TEMP/65*0./,STAR/1H*/,BLANK/1H /,DASH/1H-/,PLUS/1H+
     */
      INTEGER*2 IU
      REAL*4 X
100   FORMAT(//////)
101   FORMAT(1H0)
103   FORMAT(1X,*MAXIMUM=*,E12.5,1X,*MINIMUM=*,E12.5,1X,*RANGE=*,E12.5,1
     *X,*TOTAL=*,E12.5,1X,*MEAN=*,E12.5,1X,*STANDARD DEVIATION=*,E12.5)
104   FORMAT(1X,*PSEUDO*,5X,*LOWER*,5X,*OCCURRENCES*,21X,*EACH STAR REPR
     *ESENTS *,I5,* OCCURRENCES*/1X,*LEVEL*,6X,*LEVEL*)
105   FORMAT(31X,10(4X,I6))
106   FORMAT(1X,132A1)
107   FORMAT(3X,I2,2X,E13.5,3X,I6,1X,*I*,100A1)
108   FORMAT(1X,*MEAN SQUARE ERROR=*,E12.5,23X,*ROOT MEAN SQUARE ERROR=*
     *,E12.5)
109   FORMAT(1X,*SIGNAL TO NOISE RATIO 1=*,E12.5,17X,*SIGNAL TO NOISE RA
     *TIO 2=*,E12.5)
110   FORMAT(1X,*SIGNAL TO NOISE RATIO 1 IN DB SCALE=*,E12.5,5X,*SIGNAL
     *TO NOISE RATIO 2 IN DB SCALE=*,E12.5)
111   FORMAT(1X,*MEAN SQUARE ERROR IN DB SCALE=*,E12.5,11X,*ROOT MEAN SQ
     *UARE ERROR IN DB SCALE=*,E12.5)
      XMAX=0.
      XMIN=256.
      SUM1=0.
      SUM2=0.
      ERRSUM=0.
      DO 1 I=1,255
      READ(4) X
      READ(3) IU
      DO 17 K=1,256
      XX(K)=IU(K)
17    CONTINUE
      DO 2 J=1,255
      SUM1=SUM1+X(J)
      SUM2=SUM2+X(J)*X(J)
      ERRSUM=ERRSUM+(X(J)-XX(J))*(X(J)-XX(J))
      IF(X(J).GT.XMAX) XMAX=X(J)
      IF(X(J).LT.XMIN) XMIN=X(J)
2     CONTINUE
1     CONTINUE
      RANGE= 127.
      TOTAL=SUM1
      XXMEAN=SUM1/(255.*255.)
      SUM2=SUM2/(255.*255.)-XXMEAN*XXMEAN
      SD=SQRT(SUM2)
      ERR1=ERRSUM/(255.*255.)
      ERR2=SQRT(ERR1)
      SN1=RANGE/ERR2
      SN2=SD/ERR2
      ERRDB1=10.*ALOG10(ERR1)
      ERRDB2=20.*ALOG10(ERR2)
      SNDB1=20.*ALOG10(SN1)
      SNDB2=20.*ALOG10(SN2)
      PRINT 100
      PRINT 103, XMAX,XMIN,RANGE,TOTAL,XXMEAN,SD
      PRINT 101
      PRINT 108, ERR1,ERR2
      PRINT 101
      PRINT 111, ERRDB1,ERRDB2
      PRINT 101
      PRINT 109, SN1,SN2
      PRINT 101
      PRINT 110, SNDB1,SNDB2
      REWIND 4
      XINC=(XMAX-XMIN)/64.
      TEMP(1)=XMIN
      DO 3 I=1,255
      READ(4) X
      DO 4 J=1,255
      DO 5 K=1,64
      TEMP(K+1)=TEMP(K)+XINC
      IF(X(J)-TEMP(K+1)) 6,6,5
5     CONTINUE
6     ICT(K)=ICT(K)+1
4     CONTINUE
3     CONTINUE
      ICTMAX=0
      DO 7 I=1,64
      IF(ICT(I).GT.ICTMAX) ICTMAX=ICT(I)
7     CONTINUE
      NO=ICTMAX/100.+1.
      PRINT 101
      PRINT 104, NO
      DO 13 I=1,10
      IGRAPH(I)=I*NO*10
13    CONTINUE
      PRINT 101
      PRINT 105, IGRAPH
      DO 14 I=1,132
```

C✓

```
      BAR(I)=DASH
14    CONTINUE
      DO 16 I=30,130,10
      BAR(I)=PLUS
16    CONTINUE
      PRINT 106, BAR
      FNO=NO
      DO 8 I=1,64
      IF(ICT(I).GE.NO) GO TO 9
      DO 10 K=1,100
      SYMBOL(K)=BLANK
10    CONTINUE
      GO TO 11
9     J=ICT(I)/FNO
      J1=J+1
      DO 12 K=1,J
      SYMBOL(K)=STAR
12    CONTINUE


      DO 15 K=J1,100
      SYMBOL(K)=BLANK
15    CONTINUE
11    PRINT 107, I,TEMP(I),ICT(I),(SYMBOL(K),K=1,100)
8     CONTINUE
      PRINT 106, BAR
      PRINT 105, IGRAPH
      RETURN
      END
```