A FAST TWO-DIMENSIONAL KARHUNEN-LOEVE TRANSFORM

Robert M. Haralick
Norman Griswold
Nimita Kattiyalukwanich
University of Kansas Center for Research, Inc.
Remote Sensing Laboratory
Lawrence, Kansas 66045

Abstract

One frequently used image compression method is based on transform coding. In terms of RMS error, the best transform is the Karhunen-Loeve (Principal Components). This method is not generally used due to computational complexity. In this paper we show that under isotropic conditions the Karhunen-Loeve is almost separable and that an approximate fast principal components transform exists. Our results indicate that the fast K-L is nearly as good as the true K-L and that it yields better results than other discrete transforms such as DLT, SLANT, or Hadamard. The approximations and errors are discussed in terms of the RMS and RMS correlated error.

Introduction

The purpose of transform coding is to store or represent data in a reduced dimensional space and yet preserve the data structure. When mean square error is the optimality criterion, the principal components or Karhunen-Loeve expansion is the best. However, the principal components technique is generally not used in transform coding image compression work because of its computational complexity. In this paper we describe how to implement the principal components or Karhunen-Loeve transform (KLT) as a fast transform for image data when the image data satisfies mild stationarity and isotropic conditions.

Figure 1 illustrates the typical way transform coding is done. The N_r row and N_c columns image is partitioned into subimages or windows each having K_r rows and K_c columns. There are \( \left( \frac{N_r}{K_r} \right) \times \left( \frac{N_c}{K_c} \right) \) such subimages. Each subimage is transformed using a Hadamard, Fourier, Slant, Discrete Cosine, or Discrete Linear Basic transform (all of which have fast implementations) or by the principal components or Karhunen-Loeve transform (which is slow). Those transform domain components which have the highest energy or variance are quantized and stored or transmitted while those transform domain components which have lowest energy or variance are not retained and effectively set to zero. Data compression is achieved because the number of bits required to encode the highest energy transform components are much less than the number of bits required to encode the data in its original spatial form.

In order to achieve the data compression by the principal components technique, the grey levels in each of the \( \left( \frac{N_r}{K_r} \right) \times \left( \frac{N_c}{K_c} \right) \) subimages must be arranged as a vector and the \( (K_rK_c) \times (K_rK_c) \) auto-covariance matrix for a sample fraction \( f \) these vectors must be computed. This requires:

\[
f \left( \frac{N_r}{K_r} \right) \left( \frac{N_c}{K_c} \right) (K_rK_c) = f N_r N_c K_r K_c\]

operations. Next the eigenvectors and eigenvalues of the autocovariance matrix must be found. This requires on the order of \( (K_rK_c)^3 \) operations. To take the dot product of each subimage with \( K_rK_c \) vectors to obtain the transformed image requires \( (N_r/K_r)(N_c/K_c)(K_rK_c)^2 \) operations. The fast transform technique we describe requires:

\[
\left( \frac{N_r}{K_r} \right) \left( \frac{N_c}{K_c} \right) (K_rK_c) (K_r + K_c)
\]

operations. This represents a savings factor of \( K_rK_c/(K_r + K_c) \).

Stationarity and Isotropy

In order to begin our discussion of how a fast KLT comes about, we first must discuss what we mean by stationarity and isotropy for an image. Suppose an image I has \( N_r \) rows and \( N_c \) columns. Partition this image into mutually exclusive subimages each \( K_r \) rows by \( K_c \) columns. (We assume \( N_r \) is an integer multiple of \( K_r \) and \( N_c \) is an integer multiple of \( K_c \).) We say the image I is stationary (in the weak sense) when two conditions are satisfied:

1. The mean grey tone of all resolution cells situated in subimage row column coordinates \( \{i,j\} \) is the same constant \( \mu \) which is independent of the relative subimage coordinates \( \{i,j\} \).

2. The grey tone covariance of all pairs of resolution cells situated in subimage row column coordinates \( \{i_1,j_1\}, \{i_2,j_2\} \) is a function \( \alpha(n_1,n_2) \) only of the row column translations \( (n_1,n_2) \) and independent of the relative subimage coordinates \( \{i,j\} \).

As illustrated in Figure 2, condition (1) implies, for example, that the average grey tone of all resolution cells occupying the upper left hand corner of the subimages equals the average of all resolution cells occupying the upper right hand corner of the subimages. In other words, fix a relative coordinate of the subimage. Then the average grey tone taken over all resolution cells having those relative coordinates in the subimages is equal to the average taken over any other relative coordinates.

As illustrated in Figure 3, condition (2) implies, for example, that the average second moment grey tone taken between resolution cells occupying the first and second columns of the first row in each subimage must equal the average second moment grey tone taken between resolution cells occupying the fifth and sixth columns of the third row in each subimage. In other words, fix some relative coordinate of the subimage. Then choose a row column translation. Then the second moment grey tone taken between all resolution cells situated in the specified relative coordinates in the specified subimage and the second moment grey tone taken between all resolution cells situated in the specified relative coordinates and in the specified subimage having those relative coordinates shifted by the row and column translation is independent of the specified relative coordinates and only a function of the translation factor.

An image I is isotropic if it is stationary and if the covariance depends only on the spatial distance of the translations.
Figure 1. The transform coding technique

Figure 2. Illustrates that in a stationary image the average greytone taken over all resolution cells marked in image (a) equals the average greytone taken over all resolution cells marked in image (b)

Figure 3. Illustrates that in stationary images the second moment statistics taken over all resolution cells marked in image (a) equals the second moment statistics taken over all resolution cells marked in image (b)
Thus, for example, the covariance for a shift of one row down and two columns over is not necessarily equal to the covariance for a shift of two rows down and one column over for a stationary image, but since they both represent translation of distance \( \sqrt{5} \) they would be equal for an isotropic image.

In more mathematical terms, let

\[
Z_r = \{0, 1, \ldots, N_r - 1\}
\]

be the set of row indexes for digital image \( I \)

\[
Z_c = \{0, 1, \ldots, N_c - 1\}
\]

be the set of column indexes for digital image \( I \).

Suppose \( N_r \) is an integer multiple of \( K_r \) and \( N_c \) is an integer multiple of \( K_c \).

Let

\[
R(i, j) = \{(a, b) \in Z_r \times Z_c \mid a \mod K_r = i \text{ and } b \mod K_c = j\}
\]

\[
S(i, j, n_1, n_2) = \{l : (a, b), (c, d) \in (Z_r \times Z_c) \times (Z_r \times Z_c) \mid c = a + n_1, d = b + n_2, \text{ and } (a, b) \in R(i, j)\}
\]

An image \( I : Z_r \times Z_c \rightarrow G \) is stationary if and only if

\[
(1) \mu = \frac{1}{\#R(i, j)} \sum_{(a, b) \in R(i, j)} I(a, b) \text{ for each } (i, j)
\]

\[
(2) \sigma(n_1, n_2) = \frac{1}{\#S(i, j, n_1, n_2)} \sum_{(a, b), (c, d) \in S(i, j, n_1, n_2)} (I(a, b) - \mu)(I(c, d) - \mu) \text{ for each } (i, j)
\]

An image \( I : Z_r \times Z_c \rightarrow G \) is isotropic if and only if \( I \) is stationary and

\[
\sigma(n_1, n_2) = \sigma(m_1, m_2) \text{ whenever } n_1^2 + n_2^2 = m_1^2 + m_2^2
\]

Consider now how we could compute the covariance matrix for an image which satisfies or almost satisfies the stationarity or isotropy conditions. Shown in Figure 4 is a 10x10 image. We will partition it into 4x4 subimages (Figure 5). Figure 6 shows the general form of the autocovariance matrix when no stationarity or isotropicity assumptions are made.

To determine the covariance matrix with stationarity assumed, we generate a tableau which depicts all pairs of resolution cells situated in the same translational relationship [see Table 1]. From this table the stationary autocovariance may be formed, Figure 7. If the tableau is modified by the definition of isotropy we can generate a new table and subsequent covariance array, Table 2, and Figure 8. This assumption basically establishes equivalence classes of the lettered columns in Table 1 in the following manner:

| \(a, g\) | \(x, r\) |
| \(q, s\) | \(t, s, p, j\) |
| \(v, d\) | \(f, n, b, h\) |
| \(w, k\) | \(e, v, c, o\) |
| \(m, i\) |

**Symmetry Properties of a Covariance Matrix of an Isotropic Image**

A covariance matrix of an isotropic image has symmetry properties which permit the rapid computation of its eigenvectors. Such a covariance matrix can always be partitioned into \( K_r \times K_r \) submatrices, each submatrix being of the toeplitz form. Each such toeplitz submatrix has only \( K_c \) distinct entries as shown in Figure 9. The number of times each distinct entry occurs is given by:

<table>
<thead>
<tr>
<th>entry name</th>
<th>number of times entry occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>(v_1)</td>
<td>(n_1 = K_c)</td>
</tr>
<tr>
<td>(v_2)</td>
<td>(n_2 = 2n_1^2)</td>
</tr>
<tr>
<td>(v_3)</td>
<td>(n_3 = n_2^2)</td>
</tr>
<tr>
<td>(v_4)</td>
<td>(n_4 = n_3^2)</td>
</tr>
<tr>
<td>(v_{K-1})</td>
<td>(n_{K-1} = n_{K-2}^2)</td>
</tr>
<tr>
<td>(v_K)</td>
<td>(n_K = n_{K-1}^2 = 2)</td>
</tr>
</tbody>
</table>

146 / SPIE Vol. 66 (1975) Efficient Transmission of Pictorial Information
Figure 4. Illustrates a $16 \times 16$ image partitioned into $4 \times 4$ subimages.

Figure 6. Illustrates that the auto-covariance matrix for a non-stationary image partitioned into $4 \times 4$ subimages, has 136 distinct entries. Each distinct entry is labeled with a two character label.
Table 1 lists in each column all pairs of resolution cells situated in the same translational relationship. For example, all the resolution cell pairs listed under column c are related by three rows down and one row across.

<table>
<thead>
<tr>
<th>Resolution Cell</th>
<th>1,1</th>
<th>1,2</th>
<th>1,3</th>
<th>1,4</th>
<th>2,1</th>
<th>2,2</th>
<th>2,3</th>
<th>2,4</th>
<th>3,1</th>
<th>3,2</th>
<th>3,3</th>
<th>3,4</th>
<th>4,1</th>
<th>4,2</th>
<th>4,3</th>
<th>4,4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>y</td>
<td>x</td>
<td>w</td>
<td>v</td>
<td>q</td>
<td>p</td>
<td>r</td>
<td>k</td>
<td>i</td>
<td>h</td>
<td>d</td>
<td>e</td>
<td>c</td>
<td>b</td>
<td>a</td>
<td></td>
</tr>
<tr>
<td>1,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. Form of auto-covariance with stationary assumption.
Table 2 lists in each column all pairs of resolution cells situated in the same distance relationship. For example, all the resolution cell pairs listed under column c are related by distance $\sqrt{10}$.

Figure 8. Isotropic auto-covariance form
The total number of distinct submatrices in the partition of the covariance matrix is \( K_r \). The submatrices themselves are organized as a Toeplitz form. As shown in Figure 10 the number of times each submatrix is repeated in the covariance matrix is given by:

<table>
<thead>
<tr>
<th>submatrix</th>
<th>number of times submatrix occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 )</td>
<td>( m_1 = K )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>( m_2 = 2m_1 - 2 )</td>
</tr>
<tr>
<td>( S_3 )</td>
<td>( m_3 = m_2 - 2 )</td>
</tr>
<tr>
<td>( S_{K_r-1} )</td>
<td>( m_{K_r-1} = m_{K_r-2} - 2 )</td>
</tr>
<tr>
<td>( S_{K_r} )</td>
<td>( m_{K_r} = m_{K_r-1} - 2 = 2 )</td>
</tr>
</tbody>
</table>

Figure 9. Illustrates a 5x5 submatrix of the covariance matrix of an isotropic image. The submatrix is Toeplitz and has only five distinct entries: \( v_1, v_2, v_3, v_4, v_5 \).

Figure 10. Illustrates how the covariance matrix of an isotropic image can be partitioned into submatrices each of the Toeplitz form shown in Figure 9.

Figure 11 shows the form of a covariance matrix for an isotropic image partitioned into 3 row by 4 column subimages. With all this symmetry, we certainly should expect that the calculation of the eigenvectors of this matrix involves less work than we would need to do for a general covariance matrix. Perhaps if we are lucky, the transformation defined by the eigenvectors may even have a fast implementation.

Figure 11. Isotropic auto-covariance form when an isotropic image is partitioned into 3 x 4 subimages \( K_r = 3 \)

\[ \sum \]

Theory of Composite Matrices

Our situation is a fortunate one because the theory of composite matrices says that, if a matrix \( A \) can be partitioned into submatrices which have the same set of eigenvectors, then the eigenvectors of \( A \) can be formed as the direct product of the eigenvectors of the submatrices with the eigenvectors of the matrix of corresponding eigenvalues. Hence if:

\[
A = \begin{pmatrix}
A_1 & A_2 & A_3 \\
A_4 & A_5 & A_6 \\
A_7 & A_8 & A_9 \\
\end{pmatrix}
\]
and \( v \) is an eigenvector of each submatrix \( A_i \) with corresponding eigenvalue \( \lambda_i \), and \( u \) is an eigenvector of the matrix

\[
\begin{pmatrix}
\lambda_1 & \lambda_2 & \lambda_3 \\
\lambda_4 & \lambda_5 & \lambda_6 \\
\lambda_7 & \lambda_8 & \lambda_9 \\
\end{pmatrix}
\]

with corresponding eigenvalue \( \eta \) then: the direct product* \( uv \) is an eigenvector of \( A \) with corresponding eigenvalue \( \eta \).

The covariance matrix for the isotropic image almost has the required property of the composite matrix. The covariance matrix can be partitioned into submatrices all of the same toeplitz form. However, this is not a guarantee that the submatrices have the same eigenvectors.

Empirical work with the submatrices of the covariance matrix of an isotropic image indicates that the submatrices are almost multiples of one another and therefore have almost the same eigenvector set. This justifies the following heuristic. Generate a submatrix with the property that the normed squares of the matrix of the differences between the best multiple of it and any given submatrix of the covariance matrix is the smallest when averaged over all submatrices. Then replace each submatrix with the best multiple of the generated submatrix. This creates a covariance matrix having the required composite structure. The eigenvectors can be determined quickly and the transformation defined by the eigenvectors has a fast implementation.

To demonstrate this approximation, consider the isotropic auto-covariance matrix in Figure 11. If we generate the second moment matrix considering each submatrix as a vector the following form emerges:

\[
K_c = \begin{pmatrix}
p & q & r & q & p & q & p & q & p \\
q & r & s & t & s & r & s & t & s \\
r & s & t & s & r & s & t & s & r \\
p & q & r & q & p & q & p & q & p \\
q & r & s & t & s & r & s & t & s \\
r & s & t & s & r & s & t & s & r \\
p & q & r & q & p & q & p & q & p \\
q & r & s & t & s & r & s & t & s \\
r & s & t & s & r & s & t & s & r \\
\end{pmatrix} = \Sigma
\]

Figure 12. Form of the second moment matrix derived from the isotropic covariance matrix with each submatrix as a vector. All entries labeled \( q \), for example, are the sums of products of entries labeled \( i \) and \( j \) respectively of the matrix in Figure 11.

Notice that all the distinct variables are in the first submatrix. It is, therefore, possible to find the eigenvector having largest eigenvalue of this matrix without forming the entire matrix. All that is necessary is to store the first submatrix of the second moment matrix.

Let \( v' = (v_1, \ldots, v_9) \) be the eigenvector of \( \sum_1 \) having largest eigenvalue \( \lambda \). Then \( \sum_1 v' = \lambda v' \).

The first row of the matrix equation is

\[
pv_1 + qv_2 + rv_3 + qv_4 + pv_5 + qv_6 + rv_7 + qv_8 + pv_9 = \lambda v_1
\]

and the fifth row is

\[
pv_1 + qv_2 + rv_3 + qv_4 + pv_5 + qv_6 + rv_7 + qv_8 + pv_9 = \lambda v_5
\]

Obviously, the first row and the fifth row are equal so \( v_5 = v_1 \). In like manner, the other rows may be compared, resulting in:

\[
\begin{align*}
v_1 &= v_5 = v_9 = \lambda v_1 \\
v_2 &= v_4 = v_6 = \lambda v_2 \\
v_3 &= v_7 = \lambda v_3
\end{align*}
\]

Separating these simultaneous equations we have

\[
\begin{bmatrix}
3p & 4q & 2r \\
3q & 4r & 2s \\
3r & 4s & 2u
\end{bmatrix}
\begin{bmatrix}
v_1 \\
v_2 \\
v_3
\end{bmatrix} = \begin{bmatrix}
\lambda v_1 \\
\lambda v_2 \\
\lambda v_3
\end{bmatrix}
\]

This always happens because in the general case we have only \( K_c \) independent equations, all of which make up the first submatrix, of the second moment matrix, derived from an isotropic covariance array in this manner. We need not worry about which elements are equal as long as our index follows the toeplitz form. By using the resulting eigenvector and symmetry the appropriate multiples for each submatrix of the auto-covariance matrix can be determined. The coefficients found by this procedure may be illustrated by:

\[
C_1 = (v_1 v_2 v_3 v_2 v_1 v_3 v_2 v_1 v_3)^T (i i g f i g f i)^T
\]

* If \( u' = (u_1, \ldots, u_n) \) and \( v' = (v_1, \ldots, v_m) \), then the direct product

\[
(uv')^T = (u_1 v_1, u_1 v_2, \ldots, u_1 v_m, u_2 v_1, u_2 v_2, \ldots, u_2 v_m, \ldots, u_n v_1, u_n v_2, \ldots, u_n v_m)
\]

SPIE Vol. 66 (1975) Efficient Transmission of Pictorial Information / 151
or as \( \mathbf{c}_1 = (v_1, v_2, v_3) \begin{pmatrix} 31 \\ 4g \\ 2f \end{pmatrix} \) using just the first three elements.

Likewise:

\[ \mathbf{c}_2 = (v_1, v_2, v_3) \begin{pmatrix} 3g \\ 4h \\ 2e \end{pmatrix} \]

\[ \mathbf{c}_3 = (v_1, v_2, v_3) \begin{pmatrix} 3f \\ 4e \\ 2d \end{pmatrix} \]

\[ \mathbf{c}_4 = (v_1, v_2, v_3) \begin{pmatrix} 3c \\ 4b \\ 2a \end{pmatrix} \]

The composite matrix \( \mathbf{c} \) can then be immediately formed as:

\[
\mathbf{c} = \begin{bmatrix}
\begin{array}{ccc}
\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\
\mathbf{v}_2 & \mathbf{v}_1 & \mathbf{v}_2 \\
\mathbf{v}_3 & \mathbf{v}_2 & \mathbf{v}_1
\end{array}
\end{bmatrix}
\]

The composite matrix resulting from this operation has several properties which should be recognized: First the submatrices of the composite matrix all have the same eigenvectors (they commute). Second, each submatrix is of the same Toeplitz form. Third, corresponding to the shared eigenvectors \( \mathbf{V}_i \) is the lambda matrix of corresponding eigenvalues,

\[
\lambda = \begin{pmatrix}
\lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \\
\lambda_5 & \lambda_6 & \lambda_7 & \lambda_8 \\
\lambda_9 & \lambda_10 & \lambda_11 & \lambda_12 \\
\lambda_13 & \lambda_14 & \lambda_15 & \lambda_16
\end{pmatrix}
\]

\( \lambda_1 \) being the eigenvalue of \( \mathbf{c}_1 \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_2 \\ v_1 \end{pmatrix} \) under the eigenvector \( \mathbf{V}_1 \). Obviously, for each \( \mathbf{V}_i \) the lambda matrices differ only by a multiplicative constant and so also share the same eigenvectors. The direct product of these two sets of eigenvectors are the eigenvectors of \( \mathbf{c} \).

Since the eigenvectors of \( \mathbf{c} \) can be represented by the direct product of two sets of vectors, the eigenvector transformation has a fast implementation. (See Figures 13 and 14).

Once the fast implementation has been defined by the direct product relation of the shared eigenvectors of the submatrices and the shared eigenvectors of the lambda matrices we have created a set of basis vectors which may be used to define an orthogonal transformation on an image. (See Figures 15 and 16).
Figure 13. Each subimage has three rows by four columns. The fast implementation transforms each of the four rows of the subimage first and then transforms each of the three resulting columns.
Figure 14. The inverse transform operates on each of the three columns and then operates on each of the four resulting rows.
FORWARD TRANSFORM PROCESS

\[ \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix} \text{ Apply } E \text{ Transform by Row} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \\ f_{10} & f_{11} & f_{12} \end{bmatrix} \text{ Transform by Column} \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix} \]

INVERSE TRANSFORM PROCESS

\[ \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix} \text{ Apply } V \text{ Transform by Column} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \\ f_{10} & f_{11} & f_{12} \end{bmatrix} \text{ Transform by Row} \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix} \]

Figure 15. Row-Column operation with basis vectors.

\[ E = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{bmatrix} \text{ orthogonal row transformation basis vectors} \]

\[ V = \begin{bmatrix} v_{11} & v_{21} & v_{31} & v_{41} \\ v_{12} & v_{22} & v_{32} & v_{42} \\ v_{13} & v_{23} & v_{33} & v_{43} \\ v_{14} & v_{24} & v_{34} & v_{44} \end{bmatrix} \text{ orthogonal column transformation basis vectors} \]

Figure 16. Basis vector sets derived from composite matrix for direct product implementation.
Computational Results

In terms relative to the transform coding of the image, what we have essentially done is to see under what conditions the Karhunen-Loeve transform is separable so that each subimage can be transformed by first operating on its rows and then by columns. We determined that in order for this to happen the image had to be isotropic and we had to approximate the submatrices of the isotropic covariance matrix with submatrices having the same eigenvectors. In order to determine if the image isotropically assumption and the approximations were good ones for image data we ran some experiments comparing the standard K-L transform with the fast approximate K-L transform. We found, as described in the following discussion, that the fast approximate K-L transform gives results almost as good as the Hadamard, Fourier, and Slant. A 512 by 512, 6 bit digital image was compressed at ratios of 2 bits/pel, 1 bit/pel, and .5 bits/pel. A comparison was made between the Standard K-L transform, the Fast K-L transform, the Discrete Linear Basis, and the SLANT transform. The error criteria chosen was the RMS and RMS correlated measures (Haralick & Shanmugam, 1974).

The first error investigated was that of the stationarity and isotropic assumptions. The L2 matrix norm of the differences between the various combinations are:

\[ \| A - I \|_{L2} = \| A - S \|_{L2} + \| S - I \|_{L2} \]

and our results satisfy the triangular inequality. In terms of distance measure Figure 17 depicts the differences of these approximations. The percentage differences, from the average variance of the auto-covariance, is 1.43% for the stationary assumption and 1.13% for the isotropic assumption. Examples of the actual auto-covariance matrix and auto-covariance matrices under the stationarity and isotropicity assumption are including in Figures 18, 19, and 20 respectively.

The eigenvectors of the standard auto-covariance matrix do not, in general, have the discrete sequence property which other fast transforms have. The eigenvectors resulting from the fast K-L do have the sequence property as the other fast transforms have. The eigenvectors for both the standard K-L and Fast K-L are compared in Figures 21 and 22. The subimage size used was 4 x 4 only because of memory and computational restrictions on determining the standard K-L. The fast version can use much larger subimages because of the symmetry and storage savings.

Results of the compression at 2 bits/pel, 1 bit/pel, and .5 bits/pel are plotted for comparison in Figure 23. It is apparent that the Fast K-L out-performs the other transforms and is closest to the optimum, differing from the optimum KL only by approximately 1%. Table 3 gives a comparison of these errors.

Conclusions

In terms relative to the transform coding of an image we have presented the conditions under which the Karhunen-Loeve transform is separable and developed an approximate fast K-L transform. We have discussed the approximations of stationarity and isotropicity in terms of the L2 distance norms. Our results indicate that the fast K-L transform is comparable to other discrete transforms both in terms of sequence and compression performance. Experimental data indicates that the fast K-L transform is closest to the optimum, differing from the optimum K-L by approximately 1%.

References


Acknowledgment

This work was supported in part by the Air Force Avionics Laboratory, Wright-Patterson Air Force Base Ohio, U. S. Air Force contract number F33615-74-C-1093.
R. M. Haralick is with the Remote Sensing Lab and the Department of Electrical Engineering, University of Kansas, Lawrence, Kansas, 66045.
N. Griswold is with the Air Force Avionics Laboratory presently working at the Remote Sensing Lab, University of Kansas, Lawrence, Kansas, 66045.
Figure 17. Illustrates the distance between the actual auto-covariance matrix computed with no assumption, the stationarity assumption and the isotropic assumption.

Figure 18. Auto-covariance matrix derived from original image (upper triangle).

Figure 19. Auto-covariance matrix derived using stationary assumption.
Figure 20. Auto-covariance matrix derived using isotropic assumption.

Figure 21-22. Comparison of eigenvectors for Standard K-L and Fast K-L in order of sequency.
TABLE 3
TABLE OF RMS ERROR

<table>
<thead>
<tr>
<th>TRANSFORM</th>
<th>COMPRESSION RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.0 bits/pel</td>
</tr>
<tr>
<td>STANDARD K-L</td>
<td>2.1971</td>
</tr>
<tr>
<td>FAST K-L</td>
<td>2.1956</td>
</tr>
<tr>
<td>DLB</td>
<td>2.2492</td>
</tr>
<tr>
<td>SLANT</td>
<td>2.2492</td>
</tr>
</tbody>
</table>

% ERROR COMPARED TO STANDARD K-L

<table>
<thead>
<tr>
<th>TRANSFORM</th>
<th>COMPRESSION RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.0 bits/pel</td>
</tr>
<tr>
<td>FAST K-L</td>
<td>1.000</td>
</tr>
<tr>
<td>DLB</td>
<td>1.448</td>
</tr>
<tr>
<td>SLANT</td>
<td>1.448</td>
</tr>
</tbody>
</table>

RMS CORRELATED ERROR BETWEEN STANDARD K-L AND FAST K-L

<table>
<thead>
<tr>
<th>DISTANCE</th>
<th>RMS ERROR</th>
<th>CORREL. ERROR</th>
<th>RMS ERROR</th>
<th>CORREL. ERROR</th>
<th>RMS ERROR</th>
<th>CORREL. ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.0 bits/pel</td>
<td>1.0 bits/pel</td>
<td>.5 bits/pel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STANDARD K-L</td>
<td>2.19071 .10238 .9976</td>
<td>4.1855 .12813 .5363</td>
<td>5.9461 .13094 .5487</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAST K-L</td>
<td>2.362 .10413 .3056</td>
<td>4.2098 .10424 .4308</td>
<td>5.0094 .20142 1.0089</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 23a. Comparison of RMS Error as a function of compression ratio for K-L, Fast K-L, Slant, and DLB.

Figure 23b. Comparison of percentage error as a function of compression ratio for Fast K-L, Slant, and DLB.