

**THE UNIVERSITY OF MICHIGAN  
COMPUTING RESEARCH LABORATORY**

---

**AN INVESTIGATION OF THE DISCRETE  
KARHUNEN-LOEVE TRANSFORM: METHODS  
AND COMPUTATIONAL ASPECTS**

**Michael R. Betker,  
Edward J. Delp and  
T.N. Mudge**

**CRL-TR-35-83**

**DECEMBER 1983**

**Room 1079, East Engineering Building  
Ann Arbor, Michigan 48109  
USA  
Tel: (313) 763-8000**

ABSTRACT

In this report we discuss some of the computational aspects of the discrete Karhunen-Loeve transform. We examine the implementation of the algorithm in an image data compression scheme. We further discuss extension of the algorithm using a high speed floating point computer system.

## 1. Introduction

The Discrete Karhunen-Loeve (K-L) transform has various applications in source coding [9], image processing [10], speech processing [2], and pattern recognition [11]. The K-L transform was first used by Huang and Schultheiss [9] as a block coder to produce uncorrelated transform coefficients. They showed that the performance of this optimal transform coding technique was similar to the optimal block code when a mean square error distortion was used and a Gaussian assumption made on the input. Habibi and Wintz [1] later extended this result and there has been a great deal of interest in transform coding. Recently the Cosine transform has been shown to approximate the K-L transform for a wide class of random processes [2, 12, 13]. The issues to be investigated in this report are methods for implementing the transform, and the speed of these methods. While transforms such as the Fourier transform and Cosine transform have fast algorithms, the K-L transform has no such fast algorithm. The goal of this study is to determine the computational needs of the K-L transform and investigate the attainable speed.

The contents of this report is the following:

- 1) Algorithms for the K-L transform
- 2) Estimating the autocorrelation function of a picture
- 3) Determining the eigenvalues and eigenvectors of the correlation matrix
- 4) Performing the K-L transform
- 5) Results obtained with implementation
- 6) Computational aspects
- 7) Future goals and conclusion

Throughout this report, reference is made to an implementation of the K-L transform using the facilities of the Computer and Image Processing Research Network (CIPRNET) at The University of Michigan.

## 2. Algorithms for the K-L Transform

While the K-L transform has been shown to be the optimal transform for data compression, the amount of computation needed to perform this transform has discouraged the use of the K-L transform. The types of transforms in popular use for data compression have the feature that fast algorithms exist. In general, the K-L transform has no such fast algorithm. This section presents two algorithms for the K-L transform. The first is a method described by Jain [2], and involves making assumptions about the auto-correlation function of the picture. This algorithm has been shown to have a fast implementation similar to the FFT. The second algorithm presented is the algorithm used in an implementation performed for this report. The methods described are general in nature, and can be used on any picture.

### 2.1. A Fast K-L Transform

Jain [2] describes a fast algorithm for implementing the K-L transform. The assumption made is that the correlation function can be represented by a first-order stationary Markov process. The correlation matrix then is a tridiagonal Toeplitz matrix. The eigenvalues and eigenvectors of this matrix are shown to be analytic, and hence the K-L transform can be performed using a fast sine transform algorithm.

### 2.2. A General K-L Transform Algorithm

To determine the computational aspects of the K-L transform a particular implementation was performed using the following steps:

- 1) Estimation of the autocorrelation function
- 2) Determination of the basis pictures
- 3) Performing the transform

These steps are discussed in separate sections, but the assumptions made in the implementation are discussed in this section.

The actual implementation was designed to operate on 8x8 sub-pictures of a 256x256 picture. These values were chosen because of time factors in executing the programs and available pictures. Extensions to sub-pictures and pictures of higher dimensions are easily done. The reasoning behind the chosen dimensions are discussed in the section on computational aspects.

Estimation of the correlation function can be a very complex task, but for this implementation a simple approach was taken. For an 8x8 image the correlation matrix is 64x64 and can be defined by the function  $K(i,j)$  from the auto-correlation function  $R(m,n,p,q)$ , where  $i = mN + n$  and  $j = pN + q$ . The notation is due to Rosenfeld and Kak [10]. The method used takes the 8x8 sub-pictures in the picture and calculates the ensemble average  $E[f(m,n) f(p,q)]$  for each sub-picture to determine an estimate of  $R(m,n,p,q)$ . (See next section.)

The basis pictures are determined by finding the eigenvectors of the correlation matrix  $K$ . The method used in this implementation is the Jacobi method for diagonalizing a matrix. The diagonal elements are the eigenvalues and the columns of the transformation matrix are the eigen vectors. The transformation matrix is determined using an iterative procedure to zero-out all the non-diagonal elements. For the 64x64 auto-correlation matrix used, this process took only 100 seconds on a VAX 11/780. For larger matrices the time would increase as the cube of the dimension of the matrix, but since the basis pictures are only determined once for a class of pictures, the time needed is not that critical.

The time critical factor in the K-L transform is performing the actual transform. The general expression for any matrix transform is  $[F]=[P][f][Q]$  and requires two matrix multiplications [10]. The method used consists of taking the matrix dot product of the picture and each basis picture, which requires

$N^4$  operations. The reason for using the latter method is due to its simplicity. An implementation to be used in an actual system would determine  $[P]$  and  $[Q]$ .

This implementation proved to be a useful tool in investigating the computational aspects of the K-L transform, and we were able to obtain an estimate of processing time. The approximate execution times for the three steps discussed above are as follows:

Determining the auto-correlation function	160 seconds
Determining the basis pictures	100 seconds
Performing the K-L transform	300 seconds
Performing the inverse K-L transform	300 seconds

### 3. Estimating the Auto-Correlation Function of a Image

The first step in investigating the K-L transform is to determine an estimate of the auto-correlation function. The types of methods typically used range from making assumptions about the functional form of the autocorrelation to obtaining a true statistical estimate of the correlation function.

In previous studies of the use of the K-L transform the correlation function was assumed to be that of a Markov process. The image was used to determine the parameters of the process and analytic solutions to the problem of finding the basis pictures are possible. Inherent in these assumptions is the use of sub-pictures for performing the transform. The justification for this model is the fact that a picture is more likely to be only locally Markov and in general is not homogeneous. The reasons for using sub-pictures to perform the K-L transform are: 1) real world images are not homogeneous; 2) the optimal blocklength of the coder discussed by Huang and Schultheiss [9] is small; 3) the computational complexity of finding the eigenvalues and eigenvectors of large matrices. If the image is  $N \times N$  then the correlation matrix is  $N^2 \times N^2$ . Since our original image was  $256 \times 256$ , if sub-pictures were not used, the eigenvalues and eigenvectors of a  $65536 \times 65536$  correlation matrix would have to be determined. These methods,

and variations on them, have been shown to give good results, but it is our feeling that a better estimate of the auto-correlation function is possible, and can only give better results.

As mentioned above, for the implementation of the K-L transform performed for this study the auto-correlation function was estimated using an ensemble average through all the sub-pictures. What we mean here is that each of the 1024  $8 \times 8$  sub-pictures of the original  $256 \times 256$  image are considered to be sample pictures of a discrete parameter random field. We are assuming a finite number (1024 in this case) of sample pictures describe the ensemble for this random field. The correlation matrix is estimated by estimating the ensemble average through the sample pictures [10]. This method of estimating the correlation matrix is another common approach. The results of this estimate do not assume a homogeneous random field, and thus are more general, and hopefully more accurate.

#### 4. Calculating the Basis Pictures

The problem of determining the basis pictures for the K-L transform is reduced to the problem of determining the eigenvectors of the correlation matrix of the picture. Assuming this matrix has been determined, the eigenvector problem must be investigated.

To determine the eigenvectors of a matrix analytically is very complex and time consuming. Since it is very difficult to find the solution in closed form, iterative numerical methods are employed. Examples of such methods are the Power method, Jacobi's method, Given's method, and Householder's method (see [3] and [4] for descriptions). A correlation matrix is a real non-negative definite symmetric matrix, hence there are more techniques available (see [5]). The approach often used is to diagonalize the matrix by a series of transformations, which produce the eigenvalues on the diagonal and the eigenvectors as the

columns of the transformation. The method used in this study was the Jacobi method (see [6] for program), which diagonalizes the matrix through several iterations. The convergence of this method is tested by summing the squares of the diagonals and determining the change after each iteration. When it has converged, the diagonal elements should not change, and the sum of the squares of the diagonal elements should equal the sum of the squares for all elements of the original matrix. The examples run converged in 5 to 6 iterations. If the convergence test value was reduced the number of iterations could be reduced.

While determining the basis pictures for the K-L transform is a complicated task, it need be done only once for a set of pictures with similar second order statistics. The frequency of the operation would depend on the application, but by using sub-pictures the problem is reduced significantly.

Another consideration in determining the basis pictures is whether the picture can be assumed to be homogeneous. If this assumption can be made the correlation matrix is of block Toeplitz form and there is the possibility of a quicker solution to the eigenvector problem. This possibility has been studied but no solution obtained at this time.

### **5. Performing the Karhunen-Loeve Transform**

The problem of performing the K-L transform once the basis pictures have been determined is relatively straight forward. The same can be said for the inverse transform.

The approach using the basis pictures for performing the transform consists of calculating the coefficients needed using a matrix dot product. Each coefficient is equal to the matrix dot product of the original picture and a basis picture. The pixels of the inverse transform are each determined by taking the matrix sum of the basis picture multiplied by their corresponding transform coefficients.



The other approach to performing the forward and inverse transform is to determine the  $[P]$  and  $[Q]$  matrices defined in Section 2 and by Rosenfeld and Kak [10]. This method involves pre and post multiplying the picture by these matrices to determine the transformed picture. The inverse operation is defined similarly using the inverse matrices. The reason for using this approach is that the computation required is an order of the matrix dimension less.

The use of the K-L transform for data compression requires that some of the coefficients be requantized and possibly discarded. The reason the K-L transform is optimal is that the picture can be represented as few coefficients as possible for a given mean square error. To take advantage of this, which coefficients to use and not discard must be determined. A common approach is to order the basis pictures based on their corresponding eigenvalues. This is done since the eigenvalues are a measure of energy of a basis picture as related to this class of picture. The ordering is done with largest eigenvalue first, decreasing from there. This method was used for this study and proved adequate. The ordering determined put the low frequency basis pictures first.

## **6. Results of Karhunen-Loeve Implementation**

The study presented in this report includes an actual implementation of the K-L transform. All the algorithms needed were programmed and used to transform and inverse transform pictures. The results of these tests verified the algorithms, and allowed the analysis of the data compression capabilities.

Data compression using the K-L transform involves two considerations: the number of coefficients to use and the quantization of the coefficients. Tests were run using both these methods and only the number of coefficients. The latter tests did not quantize the coefficients. The number of coefficients needed to reconstruct each sub-picture is 64 and any compression must use only some of these.

Tests run without quantizing the coefficients showed acceptable results with the number of coefficients used ranging from 8 to 64. When all the coefficients were used the difference between the reconstructed picture and the original picture was minimal. The use of 32, 24, 16, and 8 coefficients yielded successively poorer results, but the results were still easily recognizable. One of the problems experienced was the blocking artifact resulting from the use of sub-pictures. This was only noticeable for tests with a small number of coefficients.

Tests run with the quantization of the coefficients showed little difference from the above tests when 8 bits were used, but when fewer bits were used for each coefficient the degradation from the quantization was obvious. The number of coefficients used had the same effect as in the previous set of tests. The results of these tests showed that data compression down to 1-2 bits per pixel are possible, depending on the fidelity required.

The results are shown in Figure 1.

## 7. Computational Aspects of the Karhunen-Loeve Transform

One of the goals of this study was to determine the computational complexity of the K-L transform. This was accomplished by implementing the transform and then analyzing the programs used. In some cases the complexity of a step in the algorithm was determined on paper and a different method used in the implementation. This was done for the actual transform and inverse transform, but the complexity of the method using  $[P]$  and  $[Q]$  is also presented.

The major steps in the K-L transform are estimating the auto-correlation matrix, determining the basis pictures, performing the transform, and reconstructing the picture with a subset of the coefficients. The only step requiring a fixed number of operations is the estimation of the auto-correlation matrix. Determining the basis pictures requires iterations, and the forward and inverse transforms are dependent on the number of coefficients used. The complexity

of these steps is presented below, using the order of the algorithm as a measure (the order is defined as in [7]). The variables used are  $N$  and  $s$ :  $N$ =picture size and  $s$ =subset size.

Estimating auto-correlation  $N^2 \times s^2$

Determining basis pictures  $s^6 \times \text{iteration}$

K-L transform w/o [P],[Q]  $N^2 \times s^2 \times \frac{\text{coef. used}}{s^2}$

K-L transform with [P],[Q]  $N^2 \times s \times \frac{\text{coef. used}}{s^2}$

Inverse K-L transform w/o [P],[Q]  $N^2 \times s^2 \times \frac{\text{coef. used}}{s^2}$

Inverse K-L transform with [P],[Q]  $N^2 \times s^2 \times \frac{\text{coef. used}}{s^2}$

Using the times for performing these steps in the implementation done we can analyze the change in execution time as  $s$  increases. If a 16x16 sub-picture is used the increase in determining the basis pictures is a factor of 64, which translates into an estimated execution time on the order of 1-5 hours, depending on the accuracy needed. This is undoubtedly too long, but by using a faster computer it is felt that this time can be reduced. It is also possible that another method for determining the basis pictures could be used which does not take as long.

Once the basis pictures are determined it is the forward and inverse transforms that must be executed repeatedly. For the version using [P] and [Q] the increase in execution time for a 16x16 sub-picture is only by a factor of 2. For a 16x16 sub-picture and a 256x256 picture the complexity of these steps is 16777216. As an example, the CRAY-1 computer can perform a maximum of 160 million floating point operations per second. If the forward and inverse

transform was such that the number of operations required was equal to the complexity and the CRAY-1 could be fully utilized, approximately 10 pictures could be processed per second. Neither of these assumptions are going to hold so the best we could hope for on the CRAY-1 would be less than 10 pictures per second, but probably greater than 1 picture per second. If the basis pictures can be determined in a reasonable time, this is the computation time we are looking at. Special-purpose architectures designed for these types of operations like PASM [8] could significantly improve the execution time.

### **8. Future Goals and Conclusion**

In this report a study on the Karhunen-Loeve transform has been presented. The focus of this study has been on the computational aspects of the K-L transform. The future goals for this study consist of the following:

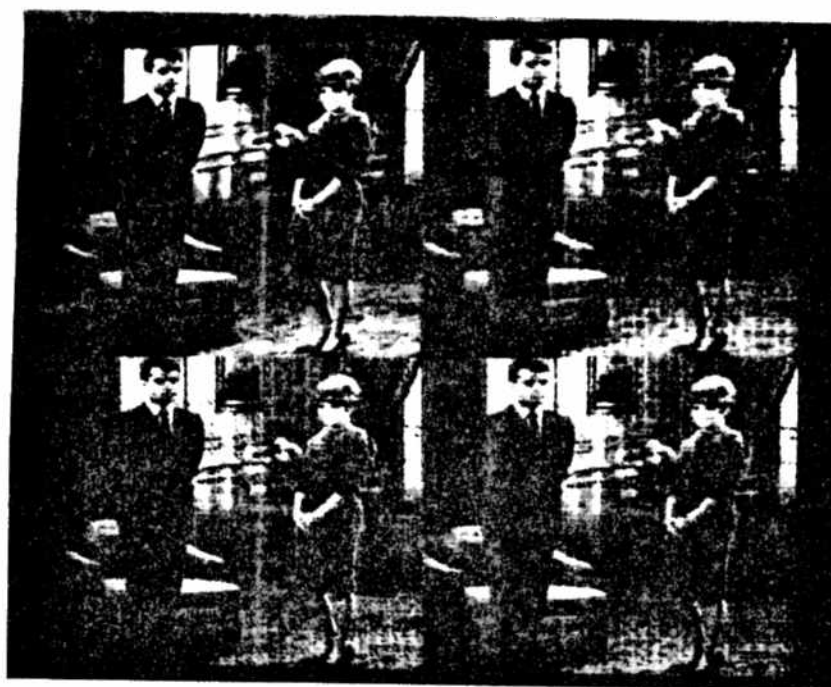
- 1) Implementation of the K-L transform on the CRAY-1 and other high-speed computers
- 2) Investigation of faster methods - eigenvalues and eigenvectors of a block Toeplitz matrix
- 3) A fidelity criteria for selecting the number of coefficients

The first goal consists of benchmarking the K-L transform using a CRAY-1 simulator and analyzing the speed of the algorithm on other architectures, such as PASM. The second goal concerns the determination of the basis pictures with the assumption that the sub-pictures are homogeneous. The resulting auto-correlation matrix is of block Toeplitz form and there is the possibility that faster methods for determining eigen vectors exist. The third goal for a fidelity criteria is motivated by the need for an evaluation of data compression performance, and as an aid in performing the data compression. These goals will be worked on in the future to arrive at a better understanding of the computational complexity of the K-L transform, and to give an estimate of the feasibility of using this transform.

The work done so far on the K-L transform resulted in a working model for performing and evaluating the transform. The results have shown that the methods used are correct, with expected performance in data compression. The goals to be met in the future will refine this work and present results that can be used to better compare the K-L transform with other data compression techniques. This will allow the system designer to evaluate if the Karhunen-Loeve transform will fit his application.

## 9. References

- [1] A. Habibi and P. A. Wintz, "Image coding by linear transformation and block quantization," *IEEE Trans. Communications*, vol. COM-19, Feb. 1971, pp. 50-62.
- [2] A. K. Jain, "A fast Karhunen-Loeve transform for a class of random processes," *IEEE Trans. Communications*, vol. COM-24, Sept. 1976, pp. 1023-1029.
- [3] R. Bellman, *Introduction to Matrix Analysis*, McGraw-Hill, New York, NY, 1960.
- [4] C. E. Froberg, *Introduction to Numerical Analysis*, Addison-Wesley, Reading, MA, 1965.
- [5] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [6] B. Carnahan et al., *Applied Numerical Methods*, John Wiley and Sons, 1964.
- [7] D. J. Kuck, *The Structure of Computers and Computations*, Vol. 1, John Wiley and Sons, New York, NY, 1978.
- [8] H. J. Siegel, L. J. Siegel, F. C. Kremmerer, P. T. Muller, H. E. Smalley, S. D. Smith, "PASM: A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition," *IEEE Trans. Computers*, Vol. C-30, No. 12, Dec. 1981, pp. 934-947.
- [9] T. T. Y. Huang and P. M. Schultheiss, "Block Quantization of Correlated Gaussian Random Variables," *IRE Trans. Communications*, Vol. COM-11, No. 9, Sept. 1963, pp. 289-296.
- [10] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, 1976.
- [11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1972.
- [12] H. Kitajima, "A Symmetric Cosine Transform," *IEEE Trans Computers*, Vol. C-29, No. 4., April 1980, pp. 317-323.
- [13] W-H. Chen and H. C. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Trans. Communications*, Vol. COM-25, Nov. 1977, pp. 1285-1292.



**Figure 1** Upper Left: Original Image.  
Upper Right: Reconstructed Image Using 8 coefficients.  
Lower Right: Reconstructed Image Using 16 coefficients.  
Lower Left: Reconstructed Image Using 24 coefficients.

