DETECTION AND ESTIMATION THEORY
EE 574
PROJECT REPORT

VIDEO CODING AND DECODING USING
THE NON ADAPTIVE ESTIMATION
AND BLIND ADAPTIVE ESTIMATION OF KLT BASIS VECTORS

FATMA OZAR
015360

SUBMITTED TO: Assist. Prof. Dr. Aykut Hocanin
An approach to video encoding and decoding by using three different algorithms techniques is discussed in this project. These methods can be used to estimate the basis vectors used by Karhunen-Loeve Transform (KLT) for encoding and decoding of the frames in a video sequences.

The first method is the blind non adaptive estimation of KLT basis vectors. What does the blind mean? Blind estimation method utilizes minimum information of the basis vectors which are being encoded. In this method, constant basis vectors are repeatedly encoded and transmitted. In this algorithm, the basis vectors are not updated.

The second method is the blind adaptive estimation of KLT basis vectors. KLT Basis vectors are adaptively updated and the encoded transformation vector is transmitted in video sequences. The result indicated that the adaptive method provides better performance.

The third method is the blind adaptive estimation of KLT basis vectors. Using this new approach result obtained by the second method is improved. In this algorithm give best result.
CONTENTS

ABSTRACT .............................................................................................................i
LIST OF FIGURE ...................................................................................................iii
LIST OF TABLES .................................................................................................iv
TABLE OF CONTENT .........................................................................................ii

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION .............................................................................................1

CHAPTER 2: THREE DIFFERENT ALGORITHMS OF KLT BASIS VECTOR

2.1 THE BLIND NON ADAPTIVE ESTIMATION OF KLT BASIS VECTOR ...... 2
2.2 CALCULATING EIGENIMAGES (BASIS VECTOR) ................................. 3
2.3 QUALITY CONSIDERATION ......................................................................... 4
2.4 THE BLIND ADAPTIVE ESTIMATION OF KLT BASIS VECTOR ........... 5
2.5 TRACKING KLT BASIS VECTOR ................................................................. 5
2.6 GENERALIZE EIGENVALUE PROBLEM .................................................... 6

CHAPTER 3 RESULTING AND IMPLEMENTATION

3.1 DISCUSS OF RESULTS ..................................................................................10
3.2 CONCLUSION ...............................................................................................17
REFERENCES ......................................................................................................18
LIST OF FIGURES

Fig 1.1a Original images from 31’st to 34’rd frame.

Fig 1.1b The average image from the training set (Using the first 30 images)

Fig 1.1c The first four difference images

Fig.1.1d it is shown first two highest Eigenvector (Basis Vector)

Fig 1.1e The reconstructed image from 31’st frame to 34’rd

Fig 2.1a Peak Signal to Noise Ratio for reconstructed the image from using first 10, 20, 30, 40 frame by using NAKLT1 algorithm

Fig 2.1b Mean Squared Error for reconstructed the image from using first 10, 20, 30, 40 frame by using NAKLT1 algorithm

Fig 2.1c Peak Signal to Noise Ratio for reconstructed the image from using first 10, 20, 30, 40 frame by using AKLT1 algorithm

Fig 2.1d Mean Squared Error for reconstructed the image from using first 10, 20, 30, 40 frame by using AKLT algorithm

Fig 2.1e Peak Signal to Noise Ratio Error for reconstructed the image from using first 10, 20, 30, 40 frame by using AKLT algorithm
LIST OF TABLES

TABLE 1 Basic Non Adaptive Algorithm NAKLT1 for Blind Non Adaptive Estimation of KLT Basis Vectors
NAKLT1: Non Adaptive algorithm in Karhunen-Loeve Transform (KLT)

TABLE 2 Basic Algorithm AKLT1 for Blind Adaptive Estimation of KLT Basis Vectors
AKLT1: Adaptive algorithm in KLT 1.

TABLE 3 Basic Algorithm AKLT for Blind Adaptive Estimation of KLT Basis Vectors
AKLT: Adaptive algorithm in KLT.
CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Karhunen-Loeve Transform (KLT) is an orthogonal linear transform[2], which is the removing correlation between the pixels or vectors. Also, KLT is known as the Hotelling transform or Principle Component Analyses.

For instance, X is a vector with zero mean. Covariance matrix of X is Rx, the KLT is the A matrix. It is multiplied with X,

\[ Y = AX \]

(1.1.a)

Y is uncorrelated and independently each other. It is simply to verify that, the transform matrix of A is constructed by the basis vector (eigenvector) of covariance matrix. Without loss of generality the rows of A are ordered so that[2]:

\[ R_y = \text{diag} (\bullet_0, \bullet_1, \ldots, \bullet_{N-1}) \quad \text{where} \quad \bullet_0 \geq \bullet_1 \geq \ldots \geq \bullet_{N-1} \geq 0 \]  

(1.1.b)

KLT is data dependent. This means that, the transform matrix is different for different matrices. And also, It is slow because transform matrix is computed every time. Therefore, KLT is only theoretical interest of image compression.

In this research, video sequences are studied to encoding and decoding. Three kinds of algorithms are used to determine the basis vector to the KLT.

In the first algorithm, the initial set is obtained by using video sequences which is called a training set. From the training set, the eigenvector (basis vectors) can be estimated. M images are used for training set. These images correspond to the highest eigenvalues. By using M images, new images can be constructed.

In the second algorithm, training set is studied. Using the training set, eigenvectors and eigenvalues are estimated. These eigenvectors and eigenvalues are updated repeatedly and transmitted to the receiver.

In the third algorithm, the second methods are improved.
CHAPTER 2: THREE DIFFERENT ALGORITHMS OF KLT BASIS VECTOR

2.1 THE BLIND NON ADAPTIVE ESTIMATION OF KLT BASIS VECTOR

Our approach in this method is to coding and decoding using KLT basis vector in video sequences. Why do we use KLT? It is mentioned above that KLT is data dependent. However, KLT efficiently reconstruct image in low bandwidth transmission. Using the minimum information, the image is constructed with minimum lost.

On the other hand, what is blind non adaptive estimation? In non adaptive estimation, the constant basis vector is encoded and transmitted to the receiver which is not updated.

In this project, video sequences are studied. In video, each frame is represented by image. And, these images repeatedly display.

The first aim in this project is to form initial images from video. Eigenvectors of covariance matrix are found from the training set. The eigenvectors can be thought of as a set of features that together characterize the variation between images [3]. The eigenvectors display as like ghostly face, and is called an eigen image. It is shown in Fig.1.1d. Linear combination of the eigen image can construct each individual image. The starting point of finding the eigenvector is the initial set of original images and estimating the best coordinate system for image compression [3]. Each coordinate is shown with image. It’s called an eigenpicture [3]. From the small collection of weights and small set of eigenpictures image is reconstructed.

For each images, the weights are described by projecting the face image onto each eigenpicture [3]. Therefore, each individual image can be characterized by using the small set of eigenpicture.
2.2 CALCULATING EIGENIMAGES (BASIS VECTOR)

In this project, video sequences which contain images (149 images) are used. The training set is set from chosen M images. Let us called these images \( \mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_M \). The average set is defined as [3]:

$$\Psi = \frac{1}{M} \sum_{M=1}^{M} \mathbf{I}_M \quad (2.2.a)$$

Each image difference is calculated from the average by vector [3].

$$\Omega_i = \mathbf{I}_i - \Psi \quad (2.2.b)$$

Training set is shown in Fig.1.1b. The KLT is the subject of the set of very large vector. The set of M orthogonal vectors is found by using KLT, \( \mathbf{u}_k \).

The best distribution of the data is described by \( \mathbf{u}_k \). The k’th vector \( \mathbf{u}_k \) is chosen from eigenvalues (\( \lambda_k \)) of covariance matrix [3]:

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} (\mathbf{u}_k^T \mathbf{I}_n)^2 \quad (2.2.c)$$

Let say the difference matrix [3]:

$$A = [\Phi_0 \Phi_1 \ldots \Phi_M] \quad (2.2.d)$$

From the difference matrix, evaluated the covariance matrix C:

$$C = A A^T \quad (2.2.e)$$

For the basis vectors take first M vectors. The remaining eigenvectors will have associated eigenvalues of zero. Basis coefficient can be calculated using the M orthonormal vector (\( \mathbf{u}_k \)) [3]:

$$W_i = \mathbf{u}_i^T (\Phi_i) \quad (2.2.f)$$

Using basis coefficient, the difference matrix can be reconstructed, (\( \hat{\Phi}_i \)) [3].

$$\hat{\Phi}_i = \sum_{j=1}^{M} \mathbf{u}_j W_i \quad (2.2.g)$$

When the new difference matrix is added to the average matrix, the image is easily reconstructed [3]. It is shown in Fig1.1e.

$$\hat{\mathbf{I}}_i = \hat{\Phi}_i + \Psi \quad (2.2.h)$$
2.3 QUALITY CONSIDERATIONS

Quality of reconstructed image can be evaluated from MSE, PSNR.

\[
\text{MSE} = \frac{1}{MN} \sum_{j=1}^{N} \left( \sum_{i=1}^{M} (X_{i,j} - Y_{i,j}) \right)^2
\]

\[\text{PSNR} = 20 \log(255/\sqrt{\text{MSE}}) \quad (2.3.b)\]

\(X_{i,j}\) = Original signal values from N by M matrix.

\(Y_{i,j}\) = Reconstructed signal values from N by M matrix

In gray scale, original and reconstructed signal values are changing from 0 to 255.

\[\text{PSNR} = \frac{20 \log(255/\sqrt{\text{MSE}})}{\text{MSE}}\]

\[\text{MSE} = \frac{1}{MN} \sum_{j=1}^{N} \left( \sum_{i=1}^{M} (X_{i,j} - Y_{i,j}) \right)^2 \]

\[= \frac{1}{MN} \sum_{j=1}^{N} \left( \sum_{i=1}^{M} (X_{i,j} - Y_{i,j}) \right)^2 \]

\[\text{PSNR} = 20 \log(255/\sqrt{\text{MSE}}) \]

\[\text{Sender} \quad \text{Reciever}\]

\[\hat{\theta}_0 = \hat{\theta}_M ( :,1:m)\]

For \(m = m+1\):frame size

\[W_m = (\hat{\theta}_M)^T \Phi_m\]

Transmit the \(W_m\)

End

\[\Phi_m = \text{Difference Matrix}\]

\[W_m = \text{Transform coefficient}\]

\[\hat{\theta}_M = \text{Eigenectors}\]

\[\hat{\theta}_0 = \hat{\theta}_M ( :,1:m)\]

For \(m = m+1\):frame size

\[\hat{\Phi}_m = \hat{\Phi}_M W_m\]

\[\hat{\Gamma}_M = \hat{\Phi}_M + \text{•}\]

\[\hat{\Gamma}_M = \text{Reconstructed Image}\]

\[\text{•} = \text{Average Image}\]

\[\hat{\Phi}_M = \text{Reconstructed Difference Matrix}\]

\[\text{TABLE 1}\]

Basic Non Algorithm NAKLT1 for Blind Non Adaptive Estimation of KLT Basis Vectors
CHAPTER 3: THE BLIND ADAPTIVE ESTIMATION OF KLT BASIS VECTOR

3.1 INTRODUCTION

In the second method, each image from the video sequence can be represented as a frame. M times frame to be encoded [1].

\[ X_M = \left[ \Gamma_1 \Gamma_2 \ldots \Gamma_M \right] \]  (3.1.a)

\( X_m \) is difference matrices (input vector) [1].

\[ R = X_M X_M^T \] having rank \( r \leq M \)  (3.1.b)

Linear combination of the eigenvectors of covariance matrix is to be form of \( X_m \)

Where \( R \) given by \( q_1, q_2, \ldots, q_r \). Eigenvalues of \( R \) are corresponding as \( 0 \geq \lambda_1 \geq \ldots \geq \lambda_{N-1} \geq 0 \). Let \( \theta \) is eigenvector [1]:

\[ \theta = [q_1, q_2, \ldots, q_r] \]  (3.1.c)

\( \theta \) is \( N \) by \( M \) matrix whose columns are the KLT basis vectors (eigenvectors of \( R \)).

Transform coefficients are [1]:

\[ y_m = (\theta)^T X_M \]  (3.1.d)

\( y_m \) is quantized as \( \hat{y}_m \). \( \hat{y}_m \) can be coded and transmitted. If the receiver has the idea of basis vector \( Q \), then \( X_m \) is found as

\[ \hat{X}_m = \theta \hat{y}_m \]  (3.1.e)

If \( X_m \) is statistically stationary the only thing that can be done is to estimate the eigenvectors and transmit. However, the eigenstructure vary considerably over time. Therefore, it is known that KLT is constantly retransmitted by the eigenvectors of \( R \).

In this project, determining the basis vectors using KLT coefficients is give only the limited images [1].

3.2 TRACKING KLT BASIS VECTOR

The idea from the subspace tracking literature can help us to accomplish a blind estimation of the KLT basis vectors. \( \hat{R}_m \) can be updated as[1]:

\[ \hat{R}_m = \alpha \hat{R}_{m-1} + X_M X_M^T \] where \( 0 < \alpha < 1 \)  (3.2.a)

\( X_M X_M^T = \) Given covariance matrix.
$R_{m-1}$ = Previous covariance matrix.

$\alpha$ = Correlation coefficient.

\[
\hat{\theta}_m = \left[q^1, q^2, \ldots, q^r\right] \quad (3.2.b)
\]

\[
\Lambda_m = \text{Diag}[\lambda_1, \lambda_2, \ldots, \lambda_r] \quad (3.2.c)
\]

$\hat{\theta}_m$, $\Lambda_m$ is estimated from covariance matrix. $\hat{\theta}_m$ are the eigenvectors, $\Lambda_m$ is eigenvalue

$R_m$ is a symmetric matrix.

Covariance matrix can be written as eigenvalues and eigenvectors of that matrix[1]. Therefore [1],

\[
\hat{R}_m = \alpha \hat{\theta}_{\theta_m} \Lambda_{\alpha_m}^{\hat{\theta}_m} + X_M X_M^T \quad (3.2.d)
\]

Using the Generalize Eigenvalue Problem, eigenvectors can be updated.

3.3 GENERALIZE EIGENVALUE PROBLEM

\[
FW_M = GW_M \Pi_M \quad (3.3.a)
\]

\[
F = \left(\hat{\theta}_m\right)^T \hat{R}_m \hat{\theta}_m \quad \text{and} \quad G = \left(\hat{\theta}_m\right)^T \hat{\theta}_m \quad (3.3.b)
\]

$W_M(1:r)$ are the eigenvector corresponding to the maximum r eigenvalues in $\Pi_m$ [1].

$\Pi_m$ is the diagonal matrix which is represented eigenvalues[1].

\[
\overline{\theta}_m = \left[\theta_{m-1}^\uparrow, V_m\right] \quad (3.3.c)
\]

$\overline{\theta}_m$ has dimension $N$ by $(r+1)$

$V_m$ is a search direction vector. It is used to increase the converge speed.

The eigenvalue can be updated as [1]:

\[
\hat{\Lambda}_m = \Pi_m \left(1:r,1:r\right) \quad (3.3.d)
\]
\[ \hat{\theta}_0 = \hat{\theta}_m(:,1:r) \]

For \( m = m+1 \):

\[ \hat{\theta}_m = \left( \hat{\theta}_{m-1} \right)^T V_m \]

\[ y_m = \left( \hat{\theta}_{m-1} \right)^T X_m \]

\[ \hat{\theta}_m = \hat{\theta}_m + y_m \left( y_m \right)^T \]

Transmit \( \hat{y}_m \) to receiver

\[ F = \alpha \hat{\theta}_m \hat{\theta}_{m-1} \Lambda_m \hat{\theta}_{m-1} \left( \hat{\theta}_{m-1} \right)^T + y_m \left( y_m \right)^T \]

\[ G = \left( \hat{\theta}_m \right)^T \hat{\theta}_m \]

solve \( FW_m = GW_m \Pi_M \)

\[ \hat{\theta}_m = \hat{\theta}_m W_m (1:r) \]

End

\[ y_m = \text{Transform Coefficient} \]

\[ W_m = \text{Updated Eigenvector} \]

\[ \hat{\theta}_m = \text{Eigenvector} \]

\[ \hat{\theta}_m = \text{New Eigenvector} \]

\[ \hat{\theta}_0 = \hat{\theta}_m(:,1:m) \]

for \( m = m+1 \):

\[ \hat{\theta}_m = \left( \hat{\theta}_{m-1} \right)^T V_m \]

Wait for \( \hat{y}_m \)

\[ \hat{X}_m = \hat{\theta}_m \hat{y}_m \]

\[ F = \alpha \hat{\theta}_m \hat{\theta}_{m-1} \Lambda_m \hat{\theta}_{m-1} \left( \hat{\theta}_{m-1} \right)^T \hat{\theta}_m + y_m \left( y_m \right)^T \]

\[ G = \left( \hat{\theta}_m \right)^T \hat{\theta}_m \]

solve \( FW_m = GW_m \Pi_M \)

\[ \hat{\theta}_m = \hat{\theta}_m W_m (1:r) \]

end

\[ \hat{\Gamma}_M = \hat{X}_m + \cdot \]

\[ \Gamma_M = \text{Reconstructed Image} \]

\[ \Psi = \text{Average Image} \]

\[ V_m = \text{Search Direction Vector} \]

\[ \hat{X}_m = \text{Reconstructed Difference} \]

**TABLE 2**

Basic Non Algorithm AKLT1 for Blind Adaptive Estimation of KLT Basis Vectors

The Search Direction Vector \( V_m \) is Assumed Known To

Both The Sender And Receiver; \( \Delta(.) \) is Quantizer

\[ \hat{y}_m \] is Transform Coefficient

Using the same initial conditions, above algorithm is worked by both sender and receiver. The search direction vectors are assumed to be known by the sender and receiver [1]. Therefore, knowing the KLT coefficients and additional scalar coefficients, \( V_m \hat{X}_m \), the receiver can be tracked to the KLT basis vectors.

In this method, the algorithm is ‘Blind’ because the receiver has no information about the \( X_m \) for tracking the KLT basis vectors [1].
\[ \hat{\theta}_0 = \hat{\theta}_m(:,1:r) \]

\( r = m - 1 \)

For \( m = m + 1 \):

Frame size

\[ y_M = \left( \hat{\theta}_M \right)^T X_M \quad ; \quad X_M = \hat{\theta}_M Y_M \]

\[ V_M = \begin{bmatrix} \left( \hat{\theta}_M \right)^T X_M \left( \hat{\theta}_M \right)^T \end{bmatrix}^T \]

\[ \bar{\theta}_m = \begin{bmatrix} \hat{\theta}_{m-1} \\ V_m \end{bmatrix} \]

\[ y_m = \left( \hat{\theta}_{m-1} \right)^T X_m \]

\[ \hat{y}_m = y_M \]

Transmit \( \hat{y}_m \) to receiver

\[ F = \bar{\theta}_m^T \bar{\theta}_m + y_m \left( \hat{y}_m \right)^T \]

\[ G = \begin{bmatrix} \bar{\theta}_m^T \\ \bar{\theta}_m \end{bmatrix} \]

solve \( FW_M = GW_M \Pi_M \)

\[ \hat{\theta}_m = \bar{\theta}_m W_m (1:r) \]

End

\( y_m \) = Transform Coefficient

\( W_m \) = Updated Eigenvector

\( \bar{\theta}_m \) = Eigenvector

\( \hat{\theta}_m \) = New Eigenvector

---

**TABLE 3**

Basic Non Algorithm AKLT for Blind Adaptive Estimation of KLT Basis Vectors

The Search Direction Vector \( V_m \) is Assumed Known To Both The Sender and Receiver; \( y_m \) is Transform Coefficient.
The third algorithm is improved the second method. In the reference paper [1], the search direction vector is assumed to be noise. However, this information is not enough to efficiently implement the algorithm AKLT1. In AKLT search direction vector in input vector which contain information about the image. Otherwise, generalize eigen vector equation is simplify and uses. The result is obtained the best performance.
In this project, NAKLT1 algorithm is implemented by using the video sequences. 149 image frame is contained in this video sequences. In Fig 1.1a show that the 31’st to 34’rd frame of the original images. The NAKLT1 is implemented by using first 10, 20, 30 and 40 images. The average image for the first 30 images is shown in Fig 1.1b. It is easy to verify that it is similar to the original image. The average image is the more blurred with compared to the 31’st to 34’rd frames. For example, the woman, which is seen the video sequences, teeth is not seen.

On the other hand, I obtained the difference matrices from subtracting the original images from the average image which is shown in Fig 1.1c. In this figures, the differences are the face and body. Faces are more cleared because the image changes occur in region of center (Face area).

The first two eigenvectors are shown in Fig 1.1d. These eigenvectors like a ghostly image. The most important eigenvector is the first one, which has the highest corresponding eigenvalue, which contains the most important information of the image. I understand from the eigenvectors (eigen images) that the changes between the frames are at the center of the image. This happened because the movement is occurred in the center region.

From the first algorithm implementation, I get the reconstructed image which is similar to the original images. It can be seen in Fig 1.1e. In the reconstructed images eyes are not clear. To understand the quality of reconstructed image, I evaluated the mean square error (MSE) and peak signal to noise ratio (PSNR).

It can be seen in Fig. 2.1a and in Fig.2.1b that the error is increased between the 80’th to 100’st. Why does this happen? Because, in the initial training set eigenvectors can not represent these images well.

On the other hand, I can realize that when increase the frame size of the training set, the performance gets better. Because, the training set describes more information in video sequences, which contain more frames.
I studied the algorithm AKLT1. However, in this algorithm search direction vector is used. Search direction vector is known as the additive white Gaussian noise (AWGN) with zero mean and variance is greater than zero [1]. The problem is the variance is not known. Therefore, when the implementation is done, search direction vector chosen randomly zero mean AWGN with having variance 1 which is not efficient. When the AKLT1 algorithm is done, the algorithm gives very bad performance. It is shown in Fig2.d. As it can be seen that the mean square error is fluctuation. However, when I use AKLT algorithm is give better result which is shown in Fig 2.1d and in Fig2.1e. If I compare In Fig 2.1e with In Fig2.1a, it is simply to understand that error is decrease. As you can see minimum values of the peak signal to noise ratio is 193. However, in Fig 2.1a the maximum point is 193. In this case, still the error increase between the 80’st to 100’rd frames. Used eigenvectors can not be represented well in these images.
Fig 1.1a Original images from 31’st to 34’rd frame

Fig 1.1b The average image from the training set (Using the first 30 images)
Fig. 1.1c The first four difference images

Fig. 1.1d It is shown first two highest Eigenvector (Basis Vector)
Fig 1.1e The reconstructed image from 31’st frame to 34’rd frame.

Fig 2.1a Peak Signal to Noise Ratio for reconstructed The image from using first 10, 20, 30, 40 frame by using NAKLT1 algorithm.
Fig 2.1b Mean Squared Error for reconstructed
The image from using first 10, 20, 30, 40 frame by using NAKLT1 algorithm

Fig 2.1c Peak Signal to Noise Ratio for reconstructed
The image from using first 10, 20, 30, 40 frame by using AKLT1 algorithm
Fig 2.1d Mean Squared Error for reconstructed
The image from using first 10, 20, 30, 40 frame by using AKLT algorithm

Fig 2.1e Peak Signal to Noise Ratio Error for reconstructed
The image from using first 10, 20, 30, 40 frame by using AKLT algorithm
4.2 CONCLUSION

The aim of in this project is to reconstruct the video sequences by using minimum transformation coefficient information at the initial basis vectors. Three different algorithms are described in this project. The principal differences between the blind non adaptive and the blind adaptive estimation of KLT basis vectors is the updated basis vectors. In the first methods, the constant eigenvectors are used to reconstruct the images. The first algorithm, which is shown in TABLE 1, is give good performance. However when I see the video sequences, slow movement is occurred. In the second algorithm, which is seen in TABLE 2, the video sequences are not clear because of unknown search direction vector. Search direction vectors are used for increasing the converge speed. It is assumed to be additive white Gaussian noise [1]. Because, the difference between the original image and reconstructed image is similar to the noise. This algorithm should give better performance. However, the search direction vector is not clear explained in the reference paper [1]. Therefore, implementation of that algorithm is not give better performance. On the other hand, I improved the AKLT algorithm, which is shown in TABLE 3, using different search direction vector. I assumed that, search direction vector is input frame. The input data include much more information than noise that’s why I used it. In AKLT algorithm, the reconstructed images are better quality than the implementation of the other two algorithms.

REFERENCES