Module
3

LOSSY IMAGE COMPRESSION SYSTEMS

Lesson
8

Transform Coding & K-L Transforms

Instructional Objectives

At the end of this lesson, the students should be able to:

1. Distinguish between spatial and transform-domain image compression systems.
2. State the objectives of transform coding.
3. Write the general expressions for forward and inverse transforms.
4. Define separable and symmetric transforms.
5. Define basis images.
6. Determine the covariance matrix of image block.
7. Represent a covariance matrix in terms of its eigenvectors and eigenvalues.
8. Define K-L transform.
9. Show that K-L transform is optimal in terms of mean-square truncation error.
10. State why K-L transforms are difficult to implement in practice.

8.0 Introduction

The lossy image compression techniques discussed in lesson-7 work in the spatial domain, since we are predicting the pixel values and the prediction errors also correspond to the pixels in the original space. It is seen that although the linear prediction mechanism essentially tries to exploit the inherent spatial redundancy, the compression ratios of Differential Pulse Code Modulation (DPCM) encoded images are not always very high. This is primarily due to the fact that in presence of sharp changes in intensity values, which are always expected in any natural image due to the presence of objects of varying intensities, prediction suffers and encoding large prediction errors in those regions lead to high consumption of bits. In terms of compression, performance is seen to be better in transform-domain approaches, in which the pixel intensities are first mapped into a set of linear, reversible transform coefficients, which are subsequently quantized and encoded. The transform coefficients are de-correlated and tend to pack most of the energy within few coefficients only. Thus, it is possible to achieve significant compression by either discarding the coefficients which do not carry much of the energy or, at least coarsely quantizing them.

In this lesson, we shall first introduce the basic concepts of transform coding techniques in a generic sense. Subsequently, we are going to discuss *Karhunen-Loeve transforms (KLT)* which is an optimal transformation in terms of the retained transform coefficients. We shall study that despite optimal performance, KLT is often not the preferred transform coding technique, since the process of transformation is heavily image dependent and the computational cost is high. It is for this reason that KLT has not been recommended in the international multimedia standards for image or video compression.
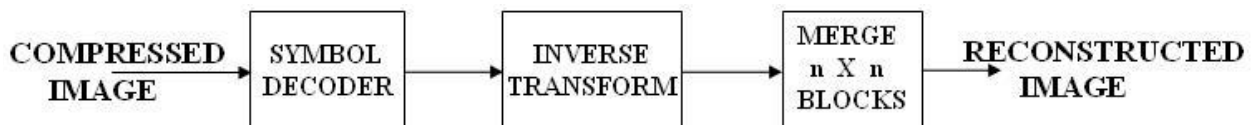
8.1 Transform Coding

The basic principle of transform coding is to map the pixel values into a set of linear transform coefficients, which are subsequently quantized and encoded. By applying an inverse transformation on the decoded transform coefficients, it is possible to reconstruct the image with some loss. It must be noted that the loss is not due to the process of transformation and inverse transformation, but due to quantization alone. Since the details of an image and hence it's spatial frequency content vary from one local region to the other, it leads to a better coding efficiency if we apply the transformation on local areas of the image, rather than applying global transformation on the entire image. Such local transformations require manageable size of the hardware, which can be replicated for parallel processing. For transform coding, the first and foremost step is to subdivide the image into non-overlapping blocks of fixed size. Without loss of generality, we can consider a square image of size N x N pixels and divide it into n^2 number of blocks, each of size (N/n) x (N/n), where $n<<N$ and is a factor of N.
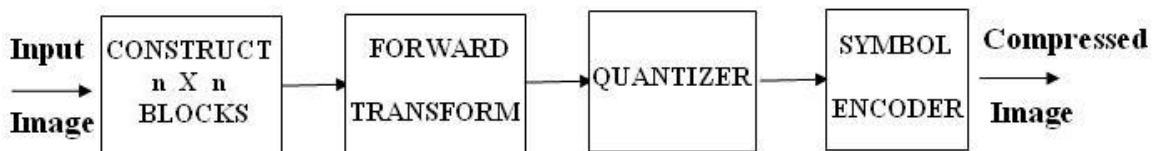
Fig 8.1: *Block Diagram of Transform Coding System Encoder.*

Fig.8.1 shows the block diagram of a transform coding system and fig.8.2 shows the corresponding decoder. Although transformation does not directly achieve any compression, it prepares the input signal to compression in the transformed domain.

Fig 8.2: *Block Diagram of Transform Coding System Decoder*

A transformation must necessarily fulfill the following properties –

    (i)   The coefficients in the transformed space should be de-correlated.

    (ii)   Only a limited number of transform coefficients should carry most of the signal energy (in other words, the transformation should possess *energy compaction* capabilities) and most of the coefficients should carry insignificant energy. Only then the quantization process can coarsely quantize those coefficients to achieve compression, without much of perceptible degradation.

A number of transformation techniques, such as Discrete Fourier Transforms (DFT), Discrete Cosine Transforms (DCT), Discrete Wavelet Transforms (DWT), K-L Transforms (KLT), Discrete Haar Transforms, and Discrete Hadamard Transforms etc. exist that fulfill the above properties, although their energy packing capabilities vary. In terms of energy packing, KLT is optimal and we are going to study KLT in the latter part of this lesson.

8.2 Generalized forward and inverse transforms

Several transformation techniques are available, but the choice of the technique depends on the amount of reconstruction error that can be available and the computational resources available.

Let us consider an image block of size n x n whose pixel intensities are represented by $s(n_1, n_2)$ $(n_1, n_2 = 0,1,\cdots\cdots n-1)$ where n_1 and n_2 are the row and the column indices of the array. Its general expression for transformation is given by

$$S(k_1, k_2) = \sum_{n_1=0}^{n-1}\sum_{n_2=0}^{n-1} s(n_1, n_2) g(n_1, n_2, k_1, k_2) \qquad k_1, k_2 = 0,1,\cdots\cdots n-1 \ldots\ldots\ldots\ldots\ldots\text{(8.1)}$$

where $S(k_1, k_2)$ $(k_1, k_2 = 0,1,\cdots\cdots n-1)$ represents the transform coefficients of the block with k_1 and k_2 as the row and the column indices in the transformed array and $g(n_1, n_2, k_1, k_2)$ is the transformation kernel that maps the input image pixels into the transform coefficients. Given the transform coefficients $S(k_1, k_2)$, the input image $s(n_1, n_2)$ may be obtained as

$$s(n_1, n_2) = \sum_{k_1=0}^{n-1}\sum_{k_2=0}^{n-1} S(k_1, k_2) h(n_1, n_2, k_1, k_2) \qquad n_1, n_2 = 0,1,\cdots\cdots n-1 \ldots\ldots\ldots\ldots\ldots\text{(8.2)}$$

In the above equation, $h(n_1, n_2, k_1, k_2)$ represents the inverse transformation kernel.

8.2.1 Separable kernel

A transformation kernel is said to be separable if it can be expressed as a product of two kernels along the row and the column, i.e.

$$g(n_1,n_2,k_1,k_2) = g_1(n_1,k_1)g_2(n_2,k_2) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.3)$$

where $g_1(.)$ and $g_2(.)$ represent the transformation kernels along the row and the column directions respectively. By a similar way, the inverse transformation kernel too can be separable. Separable transforms are easier to implement in hardware, since the transformation can first be applied along the rows (or the columns) and then along the columns (or the rows).

8.2.2 Symmetric kernel

A separable transform is symmetric, if the kernels along the row and the column have the identical function, i.e. if

$$g(n_1,n_2,k_1,k_2) = g_1(n_1,k_1)g_1(n_2,k_2) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.4)$$

Most of the transformations that we deal with have separable, symmetric kernels. For example, the forward and the inverse transformation kernels of Discrete Fourier Transform (DFT) for n x n image block is given by

$$g(n_1,n_2,k_1,k_2) = \frac{1}{n^2}\exp\left[-j\frac{2\pi(n_1k_1 + n_2k_2)}{n}\right] \dots\dots\dots\dots\dots\dots\dots\dots(8.5)$$

and

$$h(n_1,n_2,k_1,k_2) = \exp\left[j\frac{2\pi(n_1k_1 + n_2k_2)}{n}\right] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.6)$$

are separable and symmetric. The students can easily derive the row and the column transformation kernels by expressing the kernel of equation (8.5) as a product of two kernels. This is left as an exercise.

8.2.3 Basis Images

Equation (8.2) relates the pixel intensities of the image block on an element by element basis to the transformation coefficients $S(k_1,k_2)$ $(k_1,k_2 = 0,1,\dots\dots n-1)$ and there are n^2 number of similar equations, defined for each pixel element. These equations can be combined and written in the matrix form

$$\mathbf{s} = \sum_{k_1=0}^{n-1}\sum_{k_2=0}^{n-1} S(k_1,k_2)\mathbf{H}_{k_1,k_2} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.7)$$

where s is an n x n matrix containing the pixels of $s(n_1, n_2)$ and

$$\mathbf{H}_{k_1,k_2} = \begin{bmatrix} h(0,0,k_1,k_2) & h(0,1,k_1,k_2) & \cdots & \cdots & h(0,n-1,k_1,k_2) \\ h(1,0,k_1,k_2) & \vdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \cdots & \vdots \\ h(n-1,0,k_1,k_2) & h(n-1,1,k_1,k_2) & \cdots & \cdots & h(n-1,n-1,k_1,k_2) \end{bmatrix} \quad \text{............(8.8)}$$

$\mathbf{H}_{k_1,k_2}$ is an n x n matrix defined for (k_1, k_2). The image block s can therefore be realized by a weighted summation of n^2 images, each of size n x n, defined by equation (8.8) and the weights are provided by the transform coefficients $S(k_1, k_2)$. The matrix $\mathbf{H}_{k_1,k_2}$ is known as a basis image corresponding to (k_1, k_2). There are n^2 such basis images, each of size n x n, corresponding to each (k_1, k_2). Some examples of basis images for typical transforms will be shown later.

8.3 Covariance Matrix

Since transforms are applied on a block-by-block basis, each block of an image may be treated as a random field. A block may be represented by a n^2 - dimensional random variable vector x, whose elements are composed by the lexicographic ordering of pixel intensity values. We define a vector b, such that

$$\mathbf{b} = \mathbf{x} - E[\mathbf{x}] \quad \text{...(8.9)}$$

where, $E[.]$ is the expectation operator. The expectation of x can be obtained from the
mean of the random variable x over all the blocks present in the image. Thus,

$$\mathbf{b} = \mathbf{x} - \mu \quad \text{...(8.10)}$$

where,

$$\mu = \frac{1}{N_B} \sum_{\forall i} \mathbf{x}_i \quad \text{,...(8.11)}$$

i is the block index and N_B is the total number of blocks.
The covariance matrix $\mathbf{R}_b$ computed over blocks of size n x n is defined by

$$\mathbf{R}_b = E\big[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^T\big] = E\big[\mathbf{bb}^T\big] \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.12)$$

where, as before, the expectation is calculated by averaging over all the blocks. Since b is an n^2-dimensional vector, its outer product realizes an n^2 x n^2 – dimensional matrix, which is the size of $\mathbf{R}_b$. The matrix $\mathbf{R}_b$ is real and symmetric and it is possible to find a set of n^2 orthonormal eigenvectors.

Let $\mathbf{e}_i$ and λ_i, $i = 1,2,\dots\dots,n^2$ be the eigenvectors and the corresponding eigenvalues, arranged in non-increasing order, such that $\lambda_j \ge \lambda_{j+1}$ for $j = 1,2,\dots\dots,n^2-1$.

By the basic definition of eigenvectors,

$$\mathbf{R_b}\mathbf{e}_j = \lambda_j\mathbf{e}_j \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\dots(8.13)$$

Pre-multiplying both the sides of equation (8.13) by $\mathbf{e}_j^T$ and noting that $\mathbf{e}_j^T\mathbf{e}_j = 1$ for orthonormal eigenvectors, it follows that

$$\mathbf{e}_j^T\mathbf{R_b}\mathbf{e}_j = \lambda_j \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.\dots\dots.. \ (8.14)$$

We now compose a matrix Γ of dimension $n^2 \times n^2$, whose rows are formed from the eigenvectors of $\mathbf{R}_b$, ordered such that the first row of Γ is the eigenvector corresponding to the largest eigenvalue and the last row is the eigenvector corresponding to the smallest eigenvalue. Considering all the eigenvectors, we can write equation (8.14) in matrix form as

$$\Gamma\mathbf{R_b}\Gamma^T = \Lambda \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ (8.15)$$

where, Λ is a diagonal matrix of ordered eigenvalues, defined as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \cdots & \lambda_{n^2} \end{bmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ (8.16)$$

Pre-multiplying equation (8.15) by Γ^T, post-multiplying by Γ and noting the orthonormal properties of matrix Γ, i.e., $\Gamma^T = \Gamma^{-1}$, we obtain

$$\mathbf{R_b} = \Gamma^T\Lambda\Gamma \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots(8.17)$$

8.4 K-L Transforms

If we use the matrix Γ to map the block of n^2 -dimensional vector b into a transformed block of n^2 -dimensional vector y, defined by

$$\mathbf{y} = \Gamma\mathbf{b} \quad\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(8.18)}$$

the transformation is called *Karhunen-Loeve transforms (KLT)*.

The covariance matrix $\mathbf{R}_y$ of the y's is given by

$$\begin{aligned}
\mathbf{R_y} &= E\left[\mathbf{yy}^T\right] \\
&= E\left[(\Gamma\mathbf{b})(\Gamma\mathbf{b})^T\right] \\
&= \Gamma E\left[\mathbf{bb}^T\right]\Gamma^T \\
&= \Gamma\mathbf{R}_b\Gamma^T
\end{aligned} \quad\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(8.19)}$$

The pre-multiplication of $\mathbf{R}_b$ by Γ and post-multiplication by Γ^T diagonalizes $\mathbf{R}_b$ into a diagonal matrix Λ of eigenvectors and the matrix $\mathbf{R}_y$ can be written as

$$\mathbf{R_y} = \Lambda \quad\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\text{(8.20)}$$

The covariance matrix $\mathbf{R}_y$ has the same eigenvectors and eigenvalues as that of $\mathbf{R}_b$, but its off-diagonal elements are zero, which signifies that the elements of the transform-domain vectors y are uncorrelated. Using equation (8.18), it is possible to recover vector b as

$$\mathbf{b} = \Gamma^{-1}\mathbf{y} \quad\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\text{(8.21)}$$

Using equation (8.10) and orthonormality property of Γ, it is possible to reconstruct the original block x as

$$\mathbf{x} = \Gamma^T\mathbf{y} + \mu \quad\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(8.22)}$$

The above equation leads to exact reconstruction. Suppose that instead of using all the n^2
eigenvectors of $\mathbf{R}_b$, we use only *k* eigenvectors corresponding to the *k* largest eigenvalues and form a transformation matrix Γ_k of order $k \times n^2$. The resulting transformed vector $\hat{\mathbf{y}}$ therefore becomes *k*-dimensional and the reconstruction

given in equation (8.22) will not be exact. The reconstructed vector $\hat{\mathbf{x}}$ is then given by

$$\hat{\mathbf{x}} = \Gamma_k^T \hat{\mathbf{y}} + \mu \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (8.23)$$

8.5 Optimality of K-L Transform

To show that K-L Transform is optimal in the least square error sense, we first establish a relation between the variance of the original data vector x and the eigenvalues.

If we project the mean-removed vector b, defined in equation (8.10) into any of the eigenvectors $\mathbf{e}_j \left(j = 1,2,\dots\dots,n^2\right)$, the projection is defined by the inner product of the vectors b and $\mathbf{e}_j$ is given by

$$A = \mathbf{b}^T \mathbf{e}_j = \mathbf{e}_j^T \mathbf{b} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots 8.24)$$

The variance σ^2 of the projection is therefore given by

$$\begin{aligned}
\sigma^2 &= E\left[A^2\right] \\
&= E\left[\mathbf{e}_j^T \mathbf{b}\mathbf{b}^T \mathbf{e}_j\right] \\
&= \mathbf{e}_j^T E\left[\mathbf{b}\mathbf{b}^T\right]\mathbf{e}_j \\
&= \mathbf{e}_j^T \mathbf{R_b} \mathbf{e}_j
\end{aligned} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots 8.25)$$

By projecting the vector b into all the n^2 eigenvectors and using equation (8.14), we obtain the total variance as

$$\sigma_{n^2}^2 = \sum_{j=1}^{n^2} \mathbf{e}_j^T \mathbf{R}_b \mathbf{e}_j = \sum_{j=1}^{n^2} \lambda_j \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots 8.26)$$

By considering only the first k eigenvectors out of n^2, the variance of the approximating signal in the projected space is given by

$$\sigma_k^2 = \sum_{j=1}^{k} \lambda_j \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (8.27)$$

Thus, the mean-square error e_{ms} in the projected space by considering only the first k components can be obtained by subtracting equation (8.27) from equation (8.26)

$$e_{ms} = \sigma_{n^2}^2 - \sigma_k^2$$

$$= \sum_{j=1}^{n^2} \lambda_j - \sum_{j=1}^{k} \lambda_j \quad \text{...} \quad 8.28)$$

$$= \sum_{j=k+1}^{n^2} \lambda_j$$

Since, the transformation is energy-preserving, the same mean-square error exists between the original vector x and its approximation $\hat{\mathbf{x}}$. It is evident from the above equation that the mean square error is zero when $k = n^2$, i.e., if all the eigenvectors are used in the transformation. Since the λ_j's decrease monotonically, the error can be minimized by selecting the first k eigenvectors are associated with the largest eigenvalues. Thus, K-L transform is optimal in the sense that it minimizes the mean-square error between the original input vectors x and their approximations $\hat{\mathbf{x}}$.

8.6 Practical limitations of K-L Transforms

Despite the optimal performance of K-L transforms, it is rarely used in practice because of the following limitations:

(i)   The transformation matrix for a block of image is derived from the covariance matrix, which needs to be computed for every block. This makes the transformation data dependent and involves non-trivial computations.

(ii)   Perfect de-correlation in transform domain is not possible, since rarely, the image blocks can be modeled as a random field.

(iii)   No fast computational algorithms are available for its implementation.

Other transform-domain approaches, such as DFT, DCT etc. on the other hand are not image dependent and work on fixed basis images. Moreover, fast computational algorithms and efficient VLSI architectures are available for these transforms. It is seen that the sinusoidal transforms, such as the DFT or the DCT more closely approximate the information packing capability of the optimal K-L transforms.

Questions

NOTE: The students are advised to thoroughly read this lesson first and then answer the following questions. Only after attempting all the questions, they should click to the solution button and verify their answers.

PART-A

A.1. Distinguish between spatial domain and transform-domain compression approaches.

A.2. State the basic objectives of transform coding.

A.3. Write the general expressions for forward and inverse transforms.

A.4. Define separable transforms with an example.

A.5. Define symmetric transforms with an example.

A.6. Define K-L transform and its inverse, applied on a block of image.

A.7. Express the covariance matrix of a block in terms of its eigenvalues and eigenvectors.

A.8. Show that K-L transforms are optimal in least square error sense when a limited number of non-decreasingly ordered eigenvalues and the corresponding eigenvectors are considered.

A.9. Why are transforms like DCT, DFT etc. are preferred over K-L transforms from practical implementation considerations.

PART-B: Multiple Choice

In the following questions, click the best out of the four choices.
B.1 A transformation kernel for an *N x N* image block, as given

by $g(n_1, n_2, k_1, k_2) = \dfrac{1}{2}\cos\dfrac{\pi(2n_1 k_1 + 2n_2 k_2 + k_1 + k_2)}{2N} + \dfrac{1}{2}\cos\dfrac{\pi(2n_1 k_1 - 2n_2 k_2 + k_1 - k_2)}{2N}$ is

  (A) neither separable, nor symmetric.

  (B) separable, but not symmetric.

  (C) not separable, but symmetric.

  (D) both separable and symmetric.

B.2 The 2x2 basis images in an image transform are given by

$$\mathbf{H}_{0,0} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{H}_{0,1} = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{H}_{0,0} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad \mathbf{H}_{0,1} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The transform-domain coefficients $S(k_1, k_2)$ are given by

$$S(k_1, k_2) = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix}$$

The spatial-domain image is

(A) $\begin{bmatrix} 7 & 2 \\ 2 & 1 \end{bmatrix}$ (B) $2\begin{bmatrix} 4 & 1 \\ 1 & 0 \end{bmatrix}$

(C) $4\begin{bmatrix} 3 & 0 \\ 2 & 0 \end{bmatrix}$ (D) $4\begin{bmatrix} 3 & 2 \\ 0 & 0 \end{bmatrix}$

B.3 The inverse transformation kernel for a 2x2 image block is given by

$$h(n_1, n_2, k_1, k_2) = \exp\left[j\frac{2\pi(n_1 k_1 + n_2 k_2)}{4} \right]$$

(A) $\begin{bmatrix} 1 & j \\ j & -1 \end{bmatrix}$ (B) $\begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix}$

(C) $\begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}$ (D) $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

B.4 A 2x2 block image is represented by the vector $\mathbf{x} = \begin{bmatrix} 5 & 2 & 2 & 5 \end{bmatrix}^T$. The mean vector computed over the entire image is given by $\mu = 3\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$. The covariance matrix for the given block is

(A) $\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$

(B) $\begin{bmatrix} 25 & 10 & 10 & 25 \\ 10 & 4 & 4 & 10 \\ 10 & 4 & 4 & 10 \\ 25 & 10 & 10 & 25 \end{bmatrix}$

(C) $\begin{bmatrix} 4 & -2 & -2 & 4 \\ -2 & 1 & 1 & -2 \\ -2 & 1 & 1 & -2 \\ 4 & -2 & -2 & 4 \end{bmatrix}$

(D) $\begin{bmatrix} 16 & 4 & 4 & 16 \\ 4 & 1 & 1 & 4 \\ 4 & 1 & 1 & 4 \\ 16 & 4 & 4 & 16 \end{bmatrix}$

B.5 Which of the following matrices can qualify to be a Λ matrix ?

(i) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$

(ii) $\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(iii) $\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(iv) $\begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(A) Only matrix-(i)

(B) Only matrix-(iv)

(C) Matrices (ii) and (iv)

(D) All four of them.

B.6 The number of eigenvalues of the covariance matrix for an 8x8 image block will be

(A) 8

(B) 64

(C) 512

(D) 4096

B.7 For an 8x8 image block, the number of elements in the Γ matrix will be

    (A) 8

    (B) 64

    (C) 512

    (D) 4096

B.8 The Γ matrix must necessarily fulfill the following condition:

    (A) $\Gamma^T = \Gamma^{-1}$

    (B) $\Gamma^T = \Gamma$

    (C) $\Gamma = \Gamma^{-1}$

    (D) It is a diagonal matrix.

B.9 A 2x2 block image has the following eigenvalues for its covariance matrix:

$$\lambda_1 = 8, \ \lambda_2 = 4, \lambda_3 = 2, \lambda_4 = 1$$

The eigenvector corresponding to the smallest eigenvalue is dropped while performing K-L transform. The ratio of mean-square reconstruction error to the signal variance is

    (A) 1:15

    (B) 2:15

    (C) 8:15

    (D) 14:15.

PART-C: Problems

C-1.
(a) Write a computer program to lexicographically order an 4x4 block into a 16-dimensional vector and compute the mean vector and covariance matrix by considering non-overlapping 4x4 blocks over an image from the archive.

(b) Apply K-L transformation on the image after retaining only top 4 eigenvalues and the corresponding eigenvectors of the covariance matrix.

(c) Apply inverse K-L transformation on the above and reconstruct the image. Compute the PSNR of the reconstructed image.

C-2.

(a) Consider the first six frames of the video sequence "Foreman". Compose 6-element vectors by picking up pixel values at the same spatial position over six consecutive frames.

(b) Determine the mean of these 6-element vectors, considering all spatial positions. Compute the 6x6 covariance matrix and determine its eigenvalues and the corresponding eigenvectors. Retain only top two of these eigenvalues and the corresponding eigenvectors

(c) Obtain the top two principal component images by projecting the vectors (obtaining dot-products) on the two principal eigenvectors and display the results.

(d) Apply inverse K-L transformation and obtain the reconstructed frames. Compute the PSNR of each reconstructed frame.

<p style="text-align:center; color:red;">SOLUTIONS</p>

<div style="text-align:center;">

A.1

A.2

A.3

A.4

A.5

A.6

A.7

A.8

A.9

</div>

<div style="text-align:center;">

B.1 (D) B.2 (B) B.3 (A) B.4 (C) B.5 (C)

B.6 (B) B.7 (D) B.8 (A) B.9 (A).

C.1

C.2

</div>