Lecture 7

Lossy Compression

Multimedia System

Spring 2020

Introduction

- Lossless compression algorithms do not deliver *compression ratios* that are high enough. Hence, most multimedia compression algorithms are *lossy*.

- What is *lossy compression* ?
  - The compressed data is not the same as the original data, but a close approximation of it.
  - Yields a much higher compression ratio than that of lossless compression.

Distortion Measures

- The three most commonly used distortion measures in image compression are:
  - *mean square error* (MSE) $\sigma_d{}^2$,

$$\sigma_d{}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

    where x_n, y_n, and N are the input data sequence, reconstructed data sequence, and length of the data sequence respectively.
  - *signal to noise ratio* (SNR), in decibel units (dB),

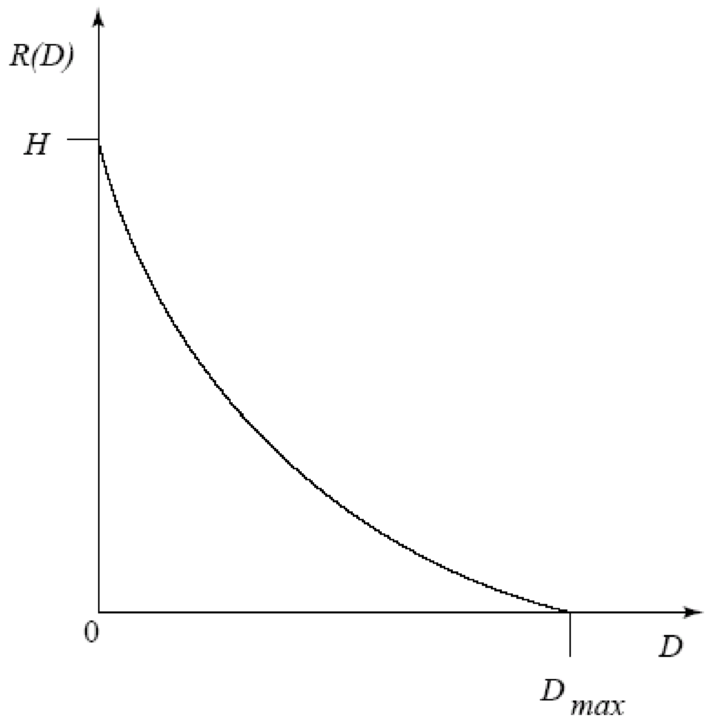
$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$

    where σ_x^2 is the average square value of the original data sequence and σ_d^2 is the MSE.
  - *peak signal to noise ratio* (PSNR),

$$PSNR = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2}$$

The Rate-Distortion Theory

- Provides a framework for the study of tradeoffs between Rate and Distortion.



Typical Rate Distortion Function.

Rate is the average number of bits required to represent each source symbol.

Transform Coding

- If **Y** is the result of a linear transform **T** of the input vector **X** in such a way that the components of **Y** are much less correlated, then **Y** can be coded more efficiently than **X**.
- If most information is accurately described by the first few components of a transformed vector, then the remaining components can be coarsely quantized, or even set to zero, with little signal distortion.
- Discrete Cosine Transform (DCT) will be studied first. In addition, we will examine the Karhunen-Loève Transform (KLT) which *optimally* decorrelates the components of the input **X**.

Spatial Frequency and DCT

- *Spatial frequency* indicates how many times pixel values change across an image block.

- The DCT formalizes this notion with a measure of how much the image contents change in correspondence to the number of cycles of a cosine wave per block.

- The role of the DCT is to *decompose* the original signal into its DC and AC components; the role of the IDCT is to *reconstruct* (re-compose) the signal.

- 1D Inverse Discrete Cosine Transform (1D IDCT):

$$f(i) = \sum_{u=0}^{M-1} \frac{C(u)}{2} \cos \frac{(2i+1)\cdot u\pi}{2M} F(u)$$

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if} \quad \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

- 1D Discrete Cosine Transform (1D DCT):

$$F(u) = \frac{C(u)}{2} \sum_{i=0}^{M-1} \cos \frac{(2i+1)\cdot u\pi}{2M} \cdot f(i)$$

Definition of DCT:

- Given an input function $f(i, j)$ over two integer variables i and j (a piece of an image), the 2D DCT transforms it into a new function $F(u, v)$, with integer u and v running over the same range as i and j. The general definition of the transform is:

$$F(u,v) = \frac{2C(u)C(v)}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \cos\frac{(2i+1)\cdot u\pi}{2M} \cdot \cos\frac{(2j+1)\cdot v\pi}{2N} \cdot f(i,j)$$

where $i, u = 0,1, \ldots, M-1$; $j, v = 0,1, \ldots, N-1$; and the constants $C(u)$ and $C(v)$ are determined by

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } \xi = 0, \\ 1 & \text{otherwise.} \end{cases}$$

- **2D Discrete Cosine Transform (2D DCT):**

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1) \cdot u\pi}{16} \cdot \cos \frac{(2j+1) \cdot v\pi}{16} \cdot f(i,j)$$

where $i, j, u, v = 0, 1, \ldots, 7$, and the constants $C(u)$ and $C(v)$ are determined by Eq. (8.5.16).

- **2D Inverse Discrete Cosine Transform (2D IDCT):** The inverse function is almost the same, with the roles of $f(i, j)$ and $F(u, v)$ reversed, except that now $C(u)C(v)$ must stand inside the sums:

$$\tilde{f}(i,j) = \sum_{u=0}^{7} \sum_{v=0}^{7} \frac{C(u)C(v)}{4} \cos \frac{(2i+1) \cdot u\pi}{16} \cdot \cos \frac{(2j+1) \cdot v\pi}{16} \cdot F(u,v)$$

where $i, j, u, v = 0, 1, \ldots, 7$.

- **1D Discrete Cosine Transform (1D DCT):**

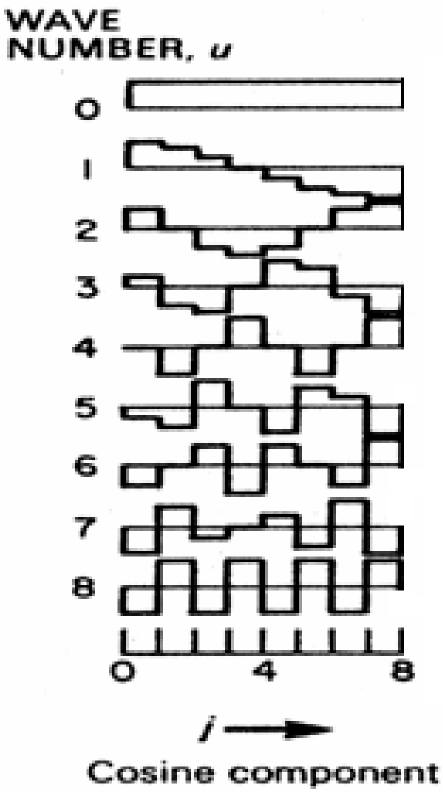
$$F(u) = \frac{C(u)}{2} \sum_{i=0}^{7} \cos \frac{(2i+1) \cdot u\pi}{16} \cdot f(i)$$

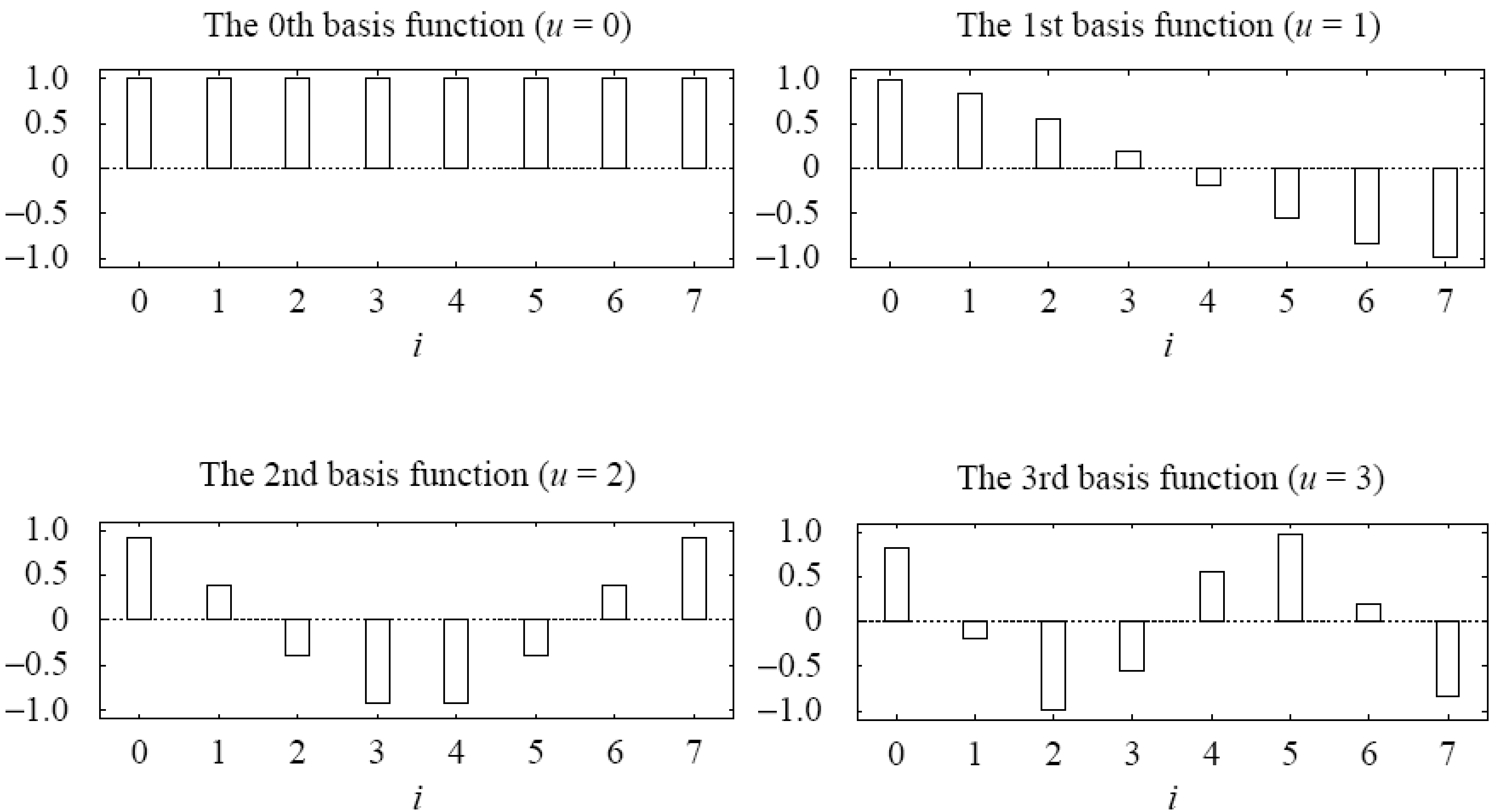
where $i, u = 0,1, \dots , 7$.

- **1D Inverse Discrete Cosine Transfo**
(1D IDCT):

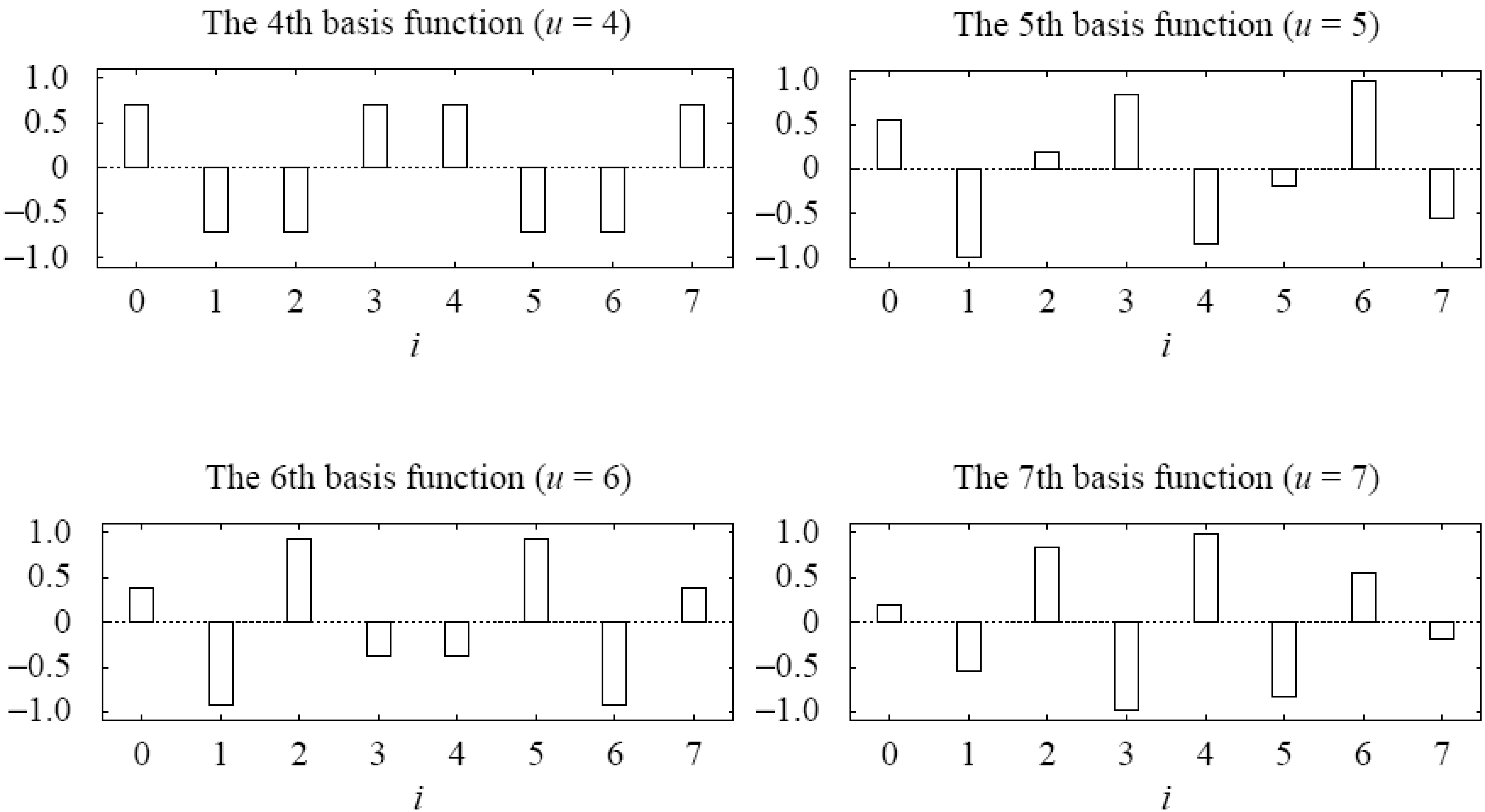
$$\tilde{f}(i) = \sum_{u=0}^{7} \frac{C(u)}{2} \cos \frac{(2i+1) \cdot u\pi}{16} F(u)$$

where $i, u = 0,1, \dots , 7$.

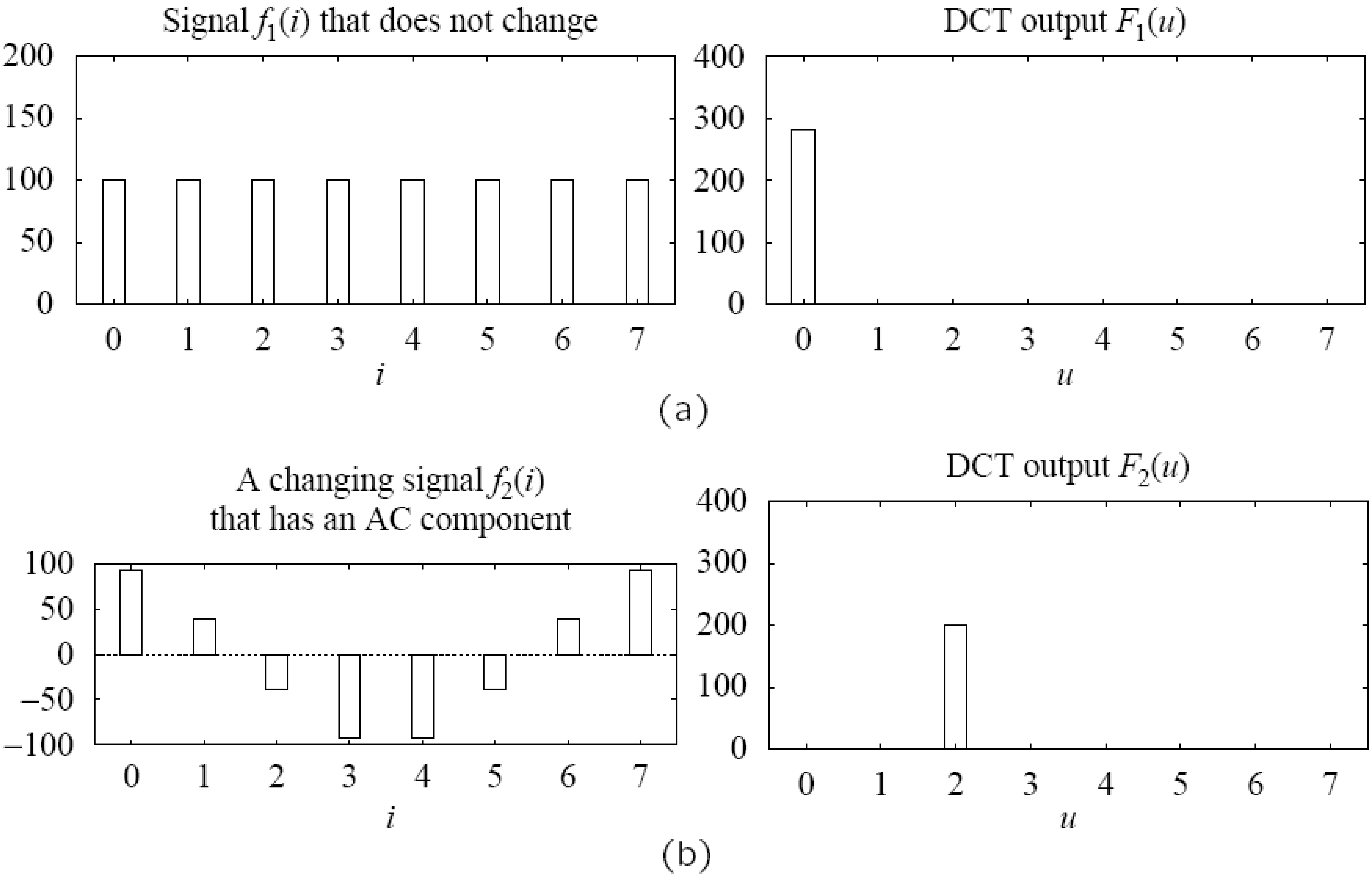




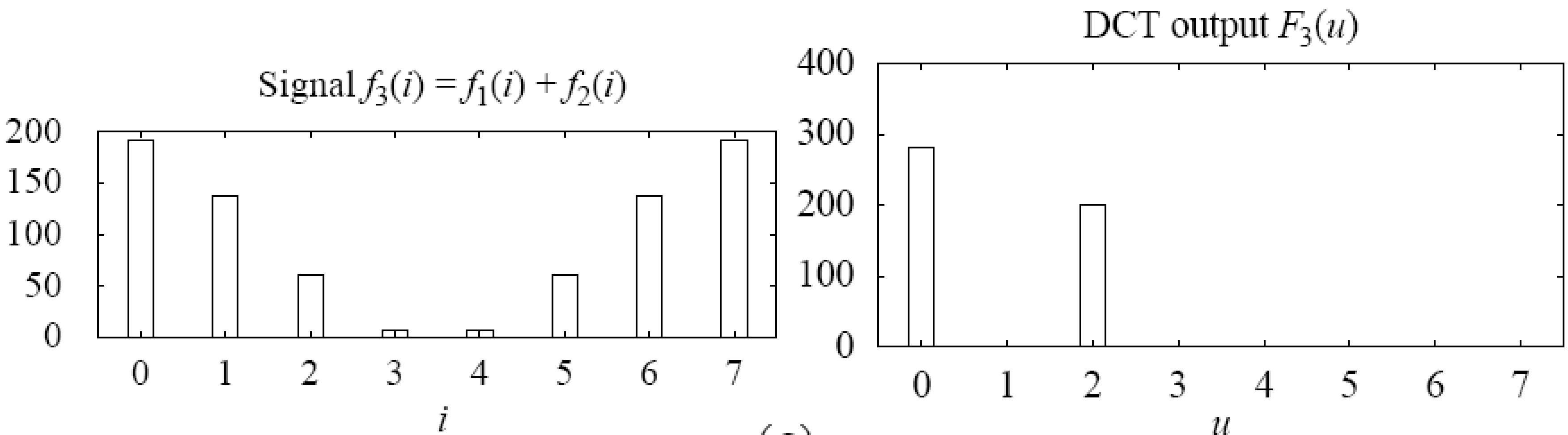
The 1D DCT basis functions.



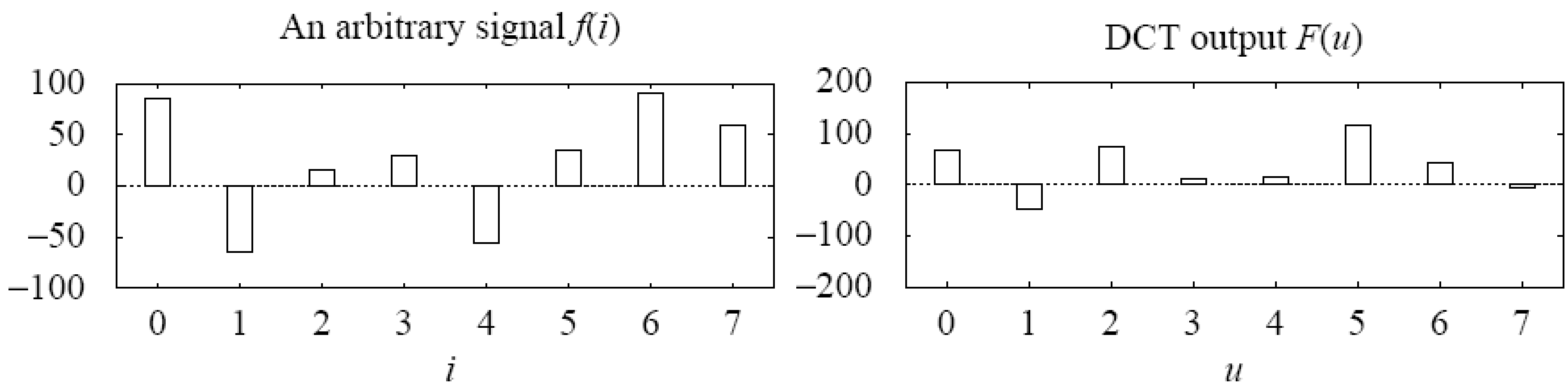
The 1D DCT basis functions.



Examples of 1D Discrete Cosine Transform: (a) A DC signal $f_1(i)$, (b) An AC signal $f_2(i)$.

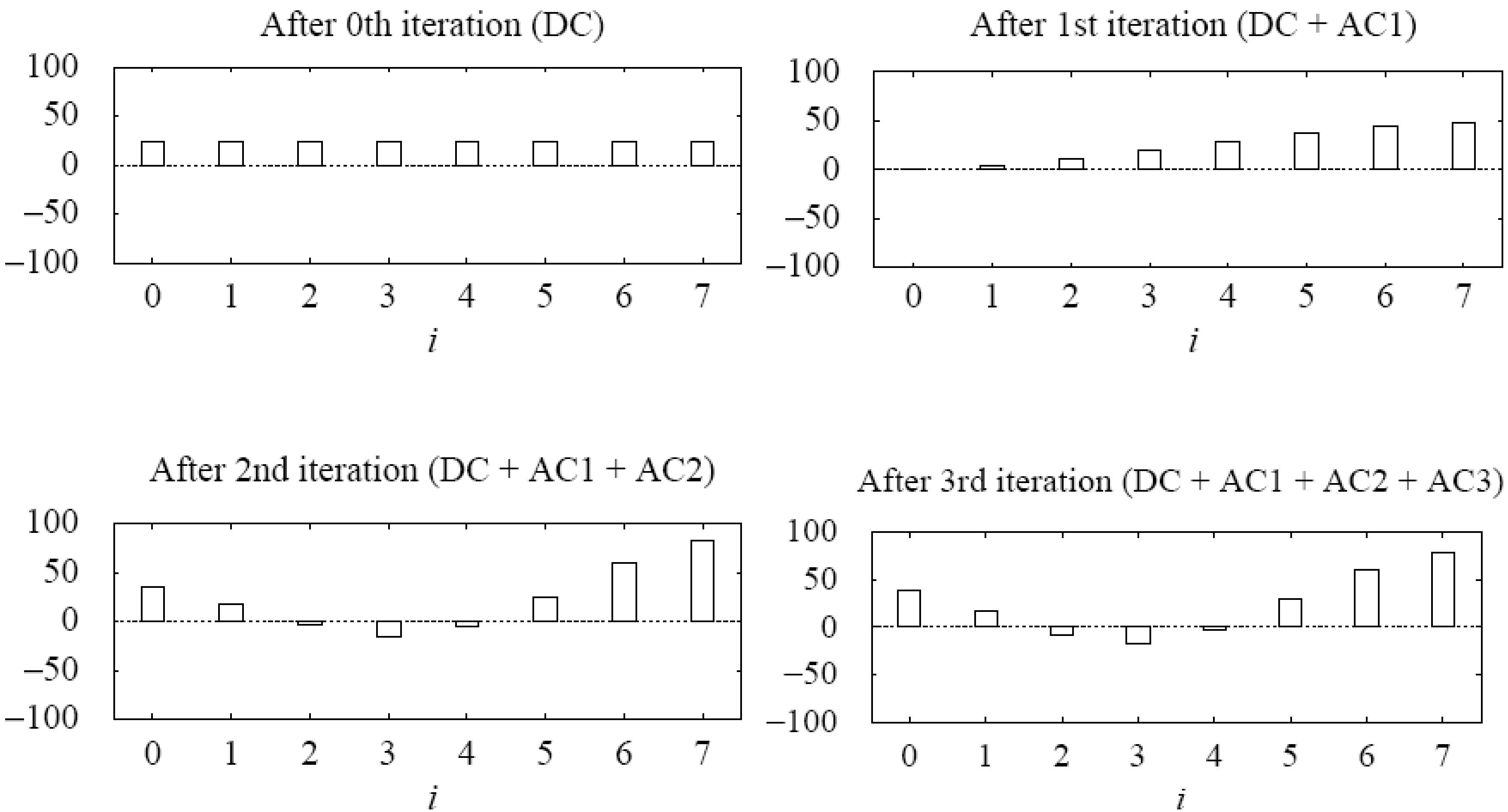


(c)

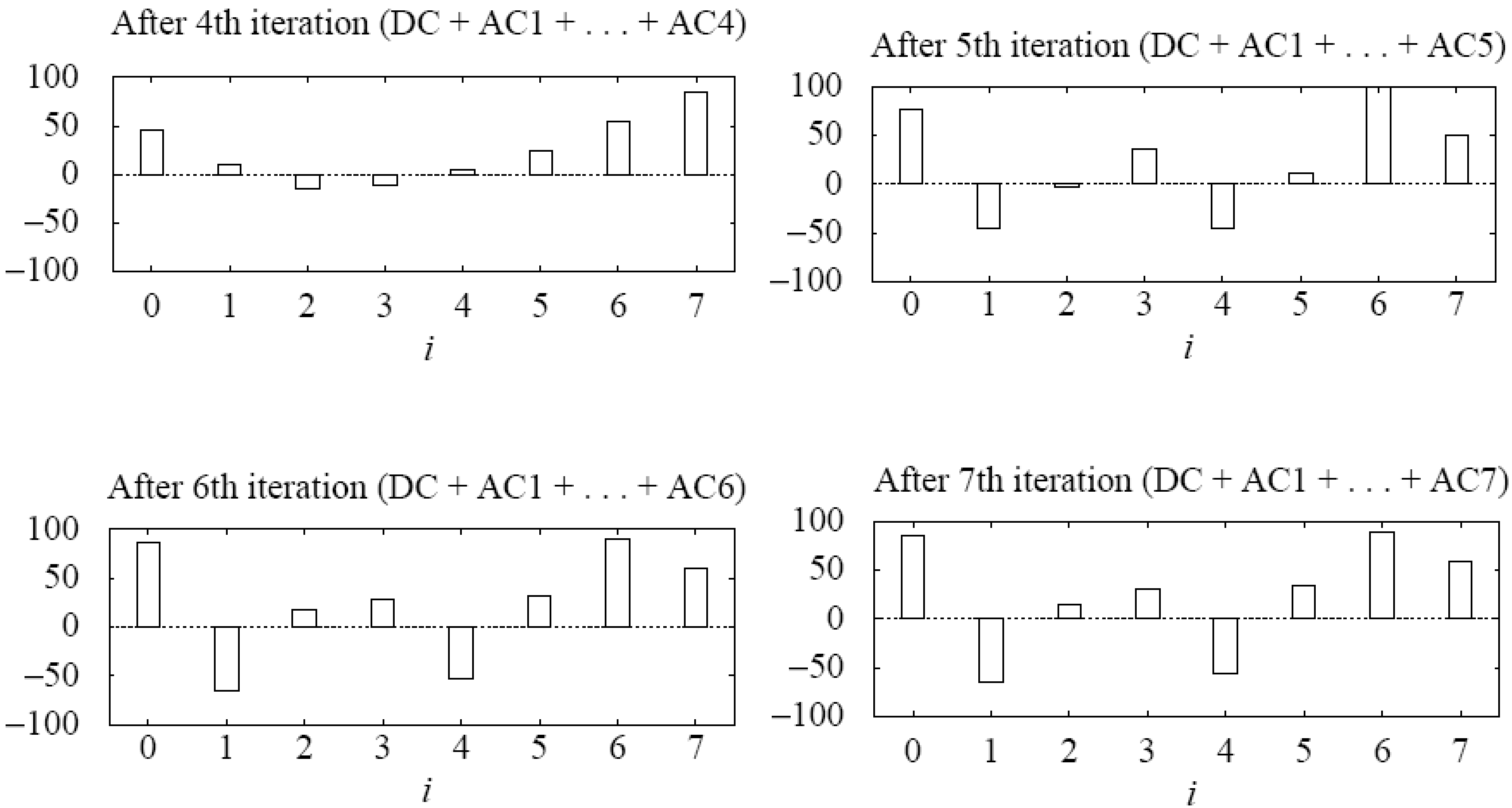


(d)

Examples of 1D Discrete Cosine Transform: (c) $f_3(i) = f_1(i) + f_2(i)$, and (d) an arbitrary signal $f(i)$.



An example of 1D IDCT.



An example of 1D IDCT.

The DCT is a linear transform:

- In general, a transform T (or function) is *linear*, iff

$$\mathrm{T}(\alpha p+\beta q)=\alpha\mathrm{T}(p)+\beta\mathrm{T}(q)$$

  where α and β are constants, p and q are any functions, variables or constants.

- From the definition in Eq. 8.17 or 8.19, this property can readily be proven for the DCT because it uses only simple arithmetic operations.

The Cosine Basis Functions

- Function $B_p(i)$ and $B_q(i)$ are *orthogonal*, if

$$\sum_i [B_p(i) \cdot B_q(i)] = 0 \quad \text{if } p \neq q \qquad (8.22)$$

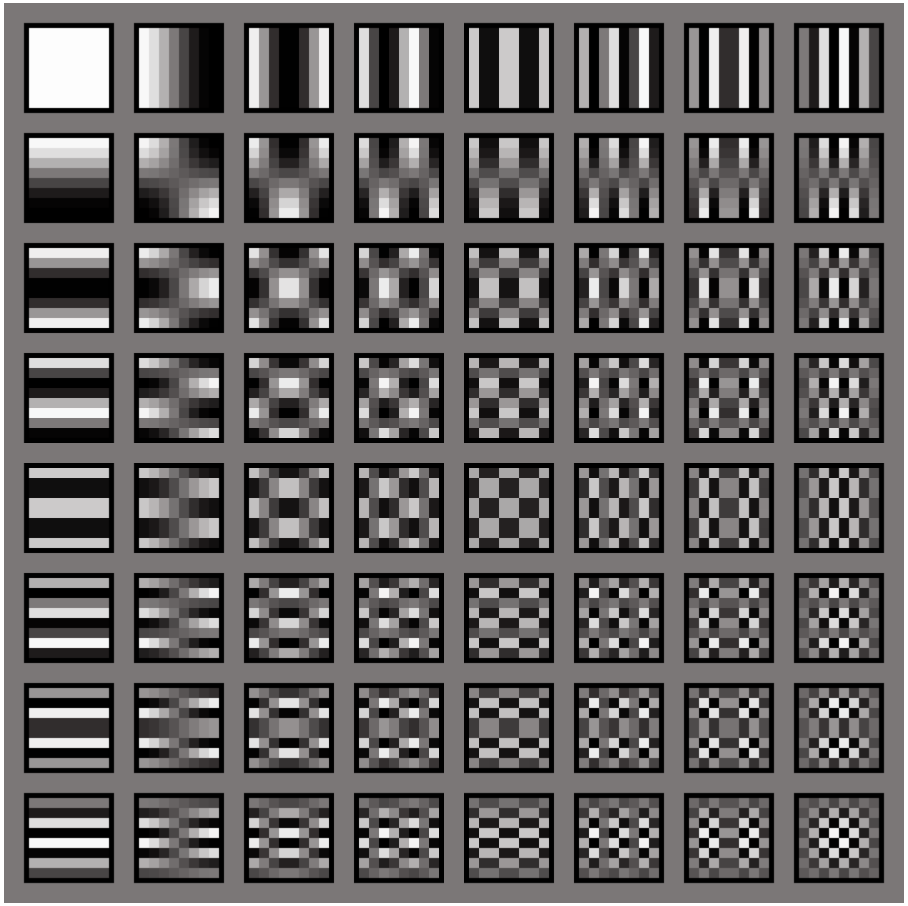
- Function $B_p(i)$ and $B_q(i)$ are *orthonormal*, if they are orthogonal and

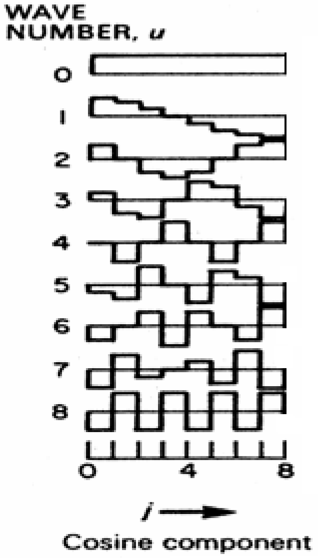
$$\sum_i [B_p(i) \cdot B_q(i)] = 1 \quad \text{if } p = q \qquad (8.23)$$

- It can be shown that:

$$\sum_{i=0}^{7} \left[\cos\frac{(2i+1)\cdot p\pi}{16} \cdot \cos\frac{(2i+1)\cdot q\pi}{16} \right] = 0 \quad \text{if } p \neq q$$

$$\sum_{i=0}^{7} \left[\frac{C(p)}{2} \cos\frac{(2i+1)\cdot p\pi}{16} \cdot \frac{C(q)}{2} \cos\frac{(2i+1)\cdot q\pi}{16} \right] = 1 \quad \text{if } p = q$$





Graphical Illustration of 8x8 2D DCT basis.

2D Separable Basis

- The 2D DCT can be *separated* into a sequence of two, 1D DCT steps:

$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos \frac{(2i+1)\cdot u\pi}{16} \cdot \cos \frac{(2j+1)\cdot v\pi}{16} \cdot f(i,j)$$

$$G(i,v) = \frac{C(v)}{2} \sum_{j=0}^{7} \cos \frac{(2j+1)\cdot v\pi}{16} \cdot f(i,j)$$

$$F(u,v) = \frac{C(u)}{2} \sum_{i=0}^{7} \cos \frac{(2i+1)\cdot u\pi}{16} \cdot G(i,v)$$

- It is straightforward to see that this simple change saves many arithmetic steps. The number of iterations required is reduced from 8x8 to 8+8.

Comparison of DCT and DFT

- The discrete cosine transform is a close counterpart to the Discrete Fourier Transform (DFT). DCT is a transform that only involves the real part of the DFT.
- For a continuous signal, we define the continuous Fourier transform F as follows:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

  Using Euler's formula, we have

$$e^{ix} = \cos(x) + i\sin(x)$$

- Because the use of digital computers requires us to discretize the input signal, we define a DFT that operates on 8 samples of the input signal $\{f_0, f_1, \dots, f_7\}$ as:

$$F_\omega = \sum_{x=0}^{7} f_x \cdot e^{-\frac{2\pi i \omega x}{8}}$$

- Writing the sine and cosine terms explicitly, we have

$$F_\omega = \sum_{x=0}^{7} f_x \cos\left(\frac{2\pi i \omega x}{8}\right) - i \sum_{x=0}^{7} f_x \sin\left(\frac{2\pi i \omega x}{8}\right)$$

- The formulation of the DCT that allows it to use only the cosine basis functions of the DFT is that we can cancel out the imaginary part of the DFT by making a symmetric copy of the original input signal.

- DCT of 8 input samples corresponds to DFT of the 16 samples made up of original 8 input samples and a symmetric copy of these, as shown in Fig. 8.10.

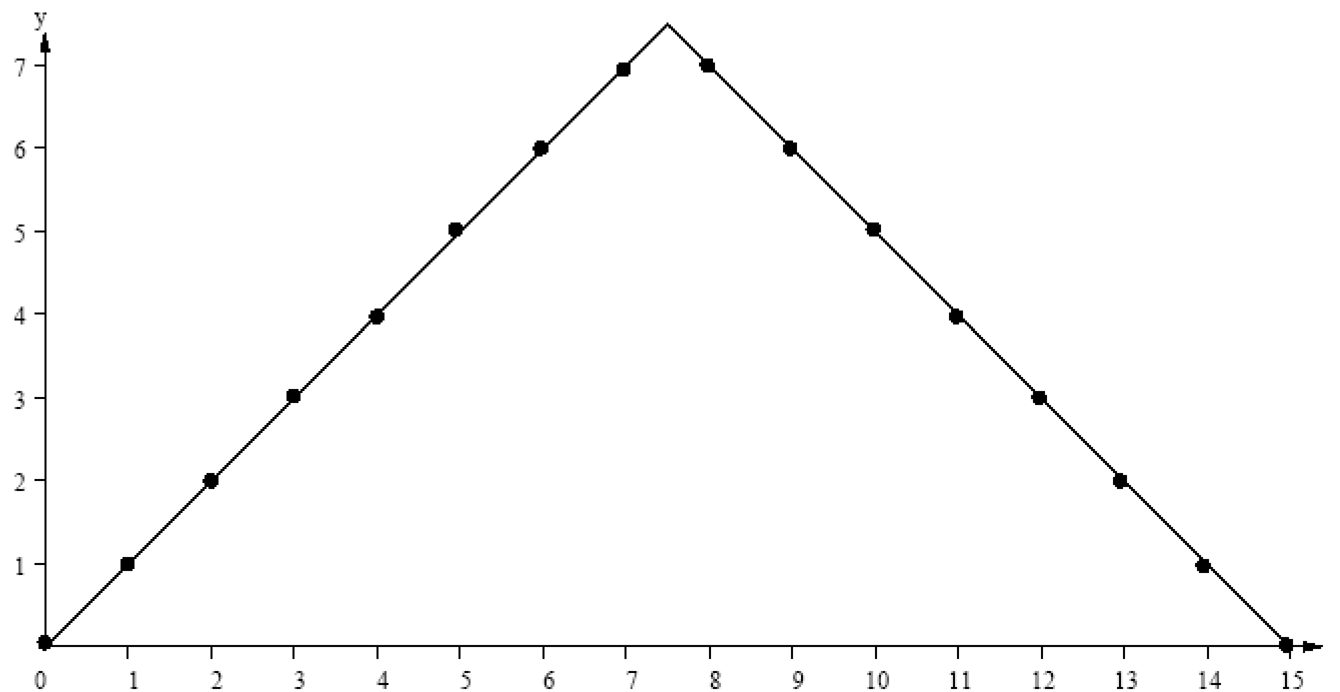


Fig. 8.10 Symmetric extension of the ramp function.

A Simple Comparison of DCT and DFT

- Table 8.1 and Fig. 8.11 show the comparison of DCT and DFT on a ramp function, if only the first three terms are used.

Table 8.1 DCT and DFT coefficients of the ramp function

Ramp	DCT	DFT
0	9.90	28.00
1	-6.44	-4.00
2	0.00	9.66
3	-0.67	-4.00
4	0.00	4.00
5	-0.20	-4.00
6	0.00	1.66
7	-0.51	-4.00

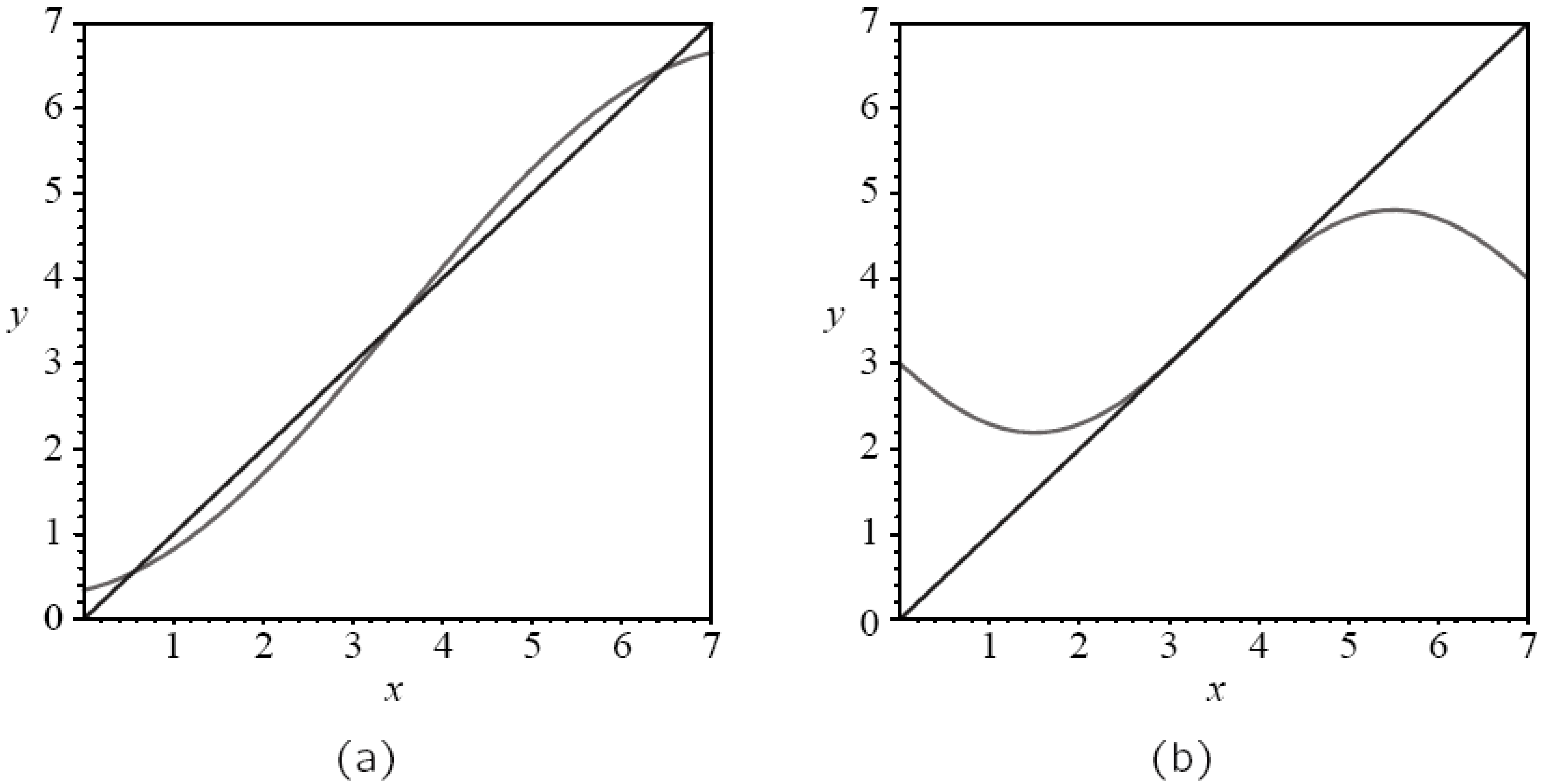


Fig. 8.11: Approximation of the ramp function: (a) 3 Term DCT Approximation, (b) 3 Term DFT Approximation.

Karhunen-Loève Transform (KLT)

- The Karhunen-Loève transform is a reversible linear transform that exploits the statistical properties of the vector representation.
- It optimally decorrelates the input signal.
- To understand the optimality of the KLT, consider the autocorrelation matrix $\mathbf{R_X}$ of the input vector $\mathbf{X}$ defined as

$$\mathbf{R_X} = E\left[\mathbf{X}\mathbf{X}^T\right]$$

$$= \begin{bmatrix} R_X(1,1) & R_X(1,2) & \cdots & R_X(1,k) \\ R_X(2,1) & R_X(2,2) & \cdots & R_X(2,k) \\ \vdots & \vdots & \ddots & \vdots \\ R_X(k,1) & R_X(k,2) & \cdots & R_X(k,k) \end{bmatrix}$$

- Our goal is to find a transform $\mathbf{T}$ such that the components of the output $\mathbf{Y}$ are uncorrelated, i.e $E[Y_t Y_s] = 0$, if $t \neq s$. Thus, the autocorrelation matrix of $\mathbf{Y}$ takes on the form of a positive diagonal matrix.
- Since any autocorrelation matrix is symmetric and non-negative definite, there are k orthogonal eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, … , $\mathbf{u}_k$ and k corresponding real and nonnegative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 0$.
- If we define the Karhunen-Loève transform as

$$\mathbf{T} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\right]^T$$

- Then, the autocorrelation matrix of $\mathbf{Y}$ becomes

$$\mathbf{R}_{\mathbf{Y}} = E\left[\mathbf{Y}\mathbf{Y}^T\right] = E\left[\mathbf{T}\mathbf{X}\mathbf{X}^T\mathbf{T}^T\right] = \mathbf{T}\mathbf{R}_{\mathbf{X}}\mathbf{T}^T$$

$$= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k \end{bmatrix}$$

KLT Example

- To illustrate the mechanics of the KLT, consider the four 3D input vectors $x_1 = (4,4,5)$, $\mathbf{x}_2 = (3,2,5)$, $\mathbf{x}_3 = (5,7,6)$, and $\mathbf{x}_4 = (6,7, 7)$.
  - Estimate the mean:

$$\mathbf{m}_x = \frac{1}{4}\begin{bmatrix} 18 \\ 20 \\ 23 \end{bmatrix}$$

  - Estimate the autocorrelation matrix of the input:

$$\mathbf{R_X} = \frac{1}{M}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T - m_x m_x^T$$

$$= \begin{bmatrix} 1.25 & 2.25 & 0.88 \\ 2.25 & 4.50 & 1.50 \\ 0.88 & 1.50 & 0.69 \end{bmatrix}$$

- The eigenvalues of $\mathbf{R}_X$ are $\lambda_1 = 6.1963$, $\lambda_2 = 0.2147$, and $\lambda_3 = 0.0264$. The corresponding eigenvectors are

$$\mathbf{u}_1 = \begin{bmatrix} 0.4385 \\ 0.8471 \\ 0.3003 \end{bmatrix}, \mathbf{u}_2 = \begin{bmatrix} 0.4460 \\ -0.4952 \\ 0.7456 \end{bmatrix}, \mathbf{u}_3 = \begin{bmatrix} -0.7803 \\ 0.1929 \\ 0.5949 \end{bmatrix}$$

- The KLT is given by the matrix

$$\mathbf{T} = \begin{bmatrix} 0.4385 & 0.8471 & 0.3003 \\ 0.4460 & -0.4952 & 0.7456 \\ -0.7803 & 0.1929 & 0.5949 \end{bmatrix}$$

- Subtracting the mean vector from each input vector and apply the KLT

$$y_1 = \begin{bmatrix} -1.2916 \\ -0.2870 \\ -0.2490 \end{bmatrix}, y_2 = \begin{bmatrix} -3.4242 \\ 0.2573 \\ 0.1453 \end{bmatrix}, y_3 = \begin{bmatrix} 1.9885 \\ -0.5809 \\ 0.1445 \end{bmatrix}, y_4 = \begin{bmatrix} 2.7273 \\ 0.6107 \\ -0.0408 \end{bmatrix}$$

- Since the rows of $\mathbf{T}$ are orthonormal vectors, the inverse transform is just the transpose: $\mathbf{T}^{-1} = \mathbf{T}^T$, and

$$\mathbf{x} = \mathbf{T}^T \mathbf{y} + \mathbf{m}_x$$

- In general, after the KLT most of the "energy" of the transform coefficients are concentrated within the first few components. This is the "energy compaction" property of the KLT.