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),

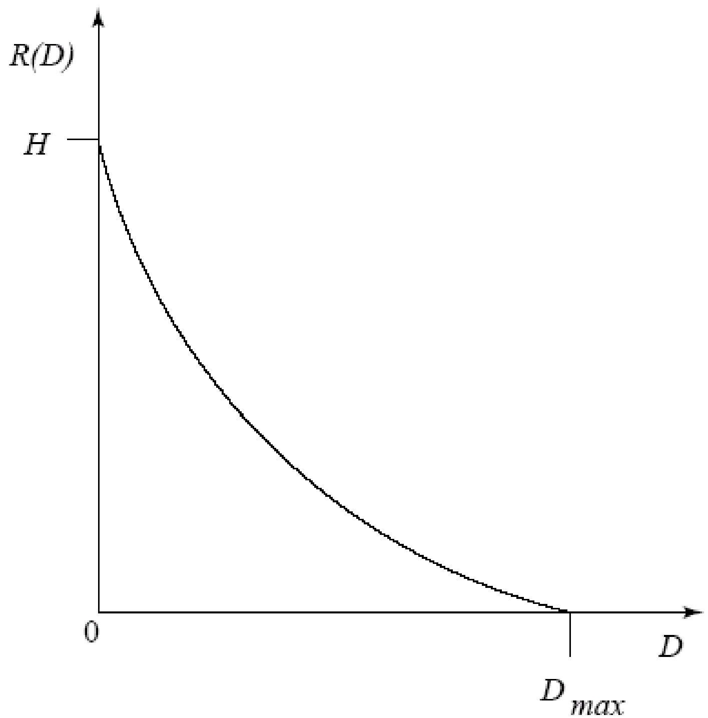
$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$

    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.



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

Typical Rate Distortion Function.

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):

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

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

- 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):**

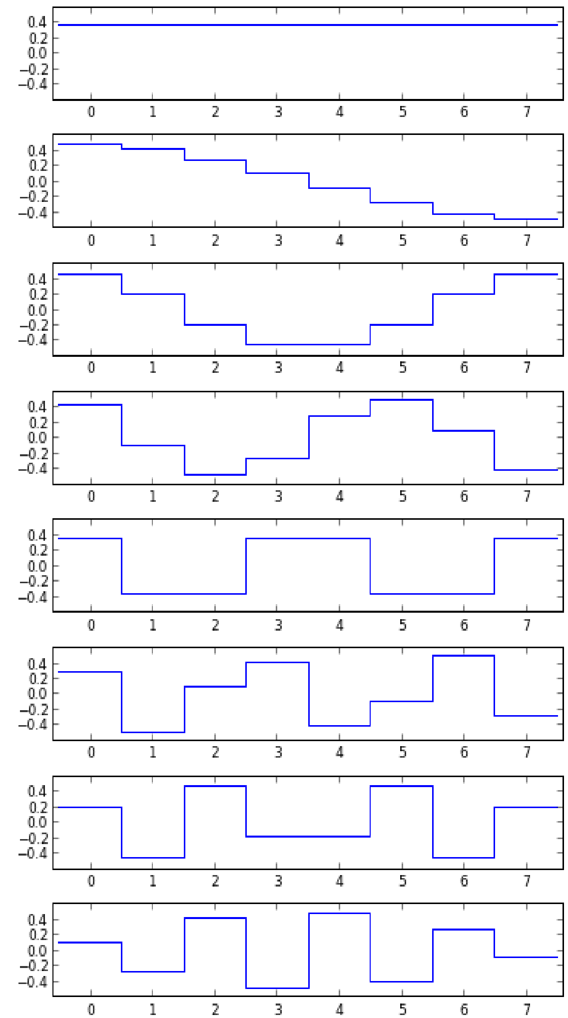
$$F(u) = \frac{C(u)}{2} \sum_{i=0}^{7} \cos \frac{(2i+1) \cdot u\pi}{16} \cdot f(i)$$

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

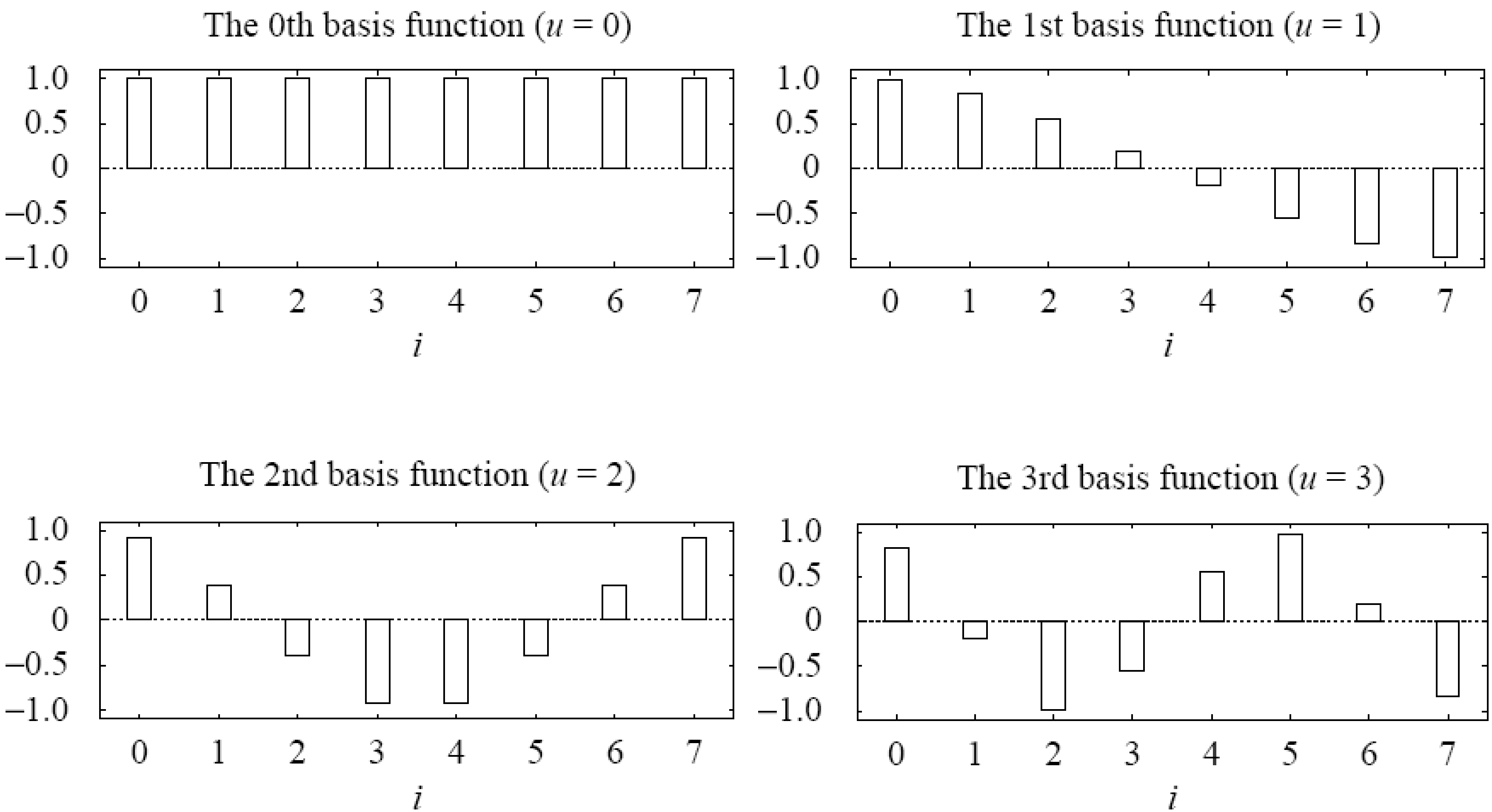
- **1D Inverse Discrete Cosine Transform (1D IDCT):**

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

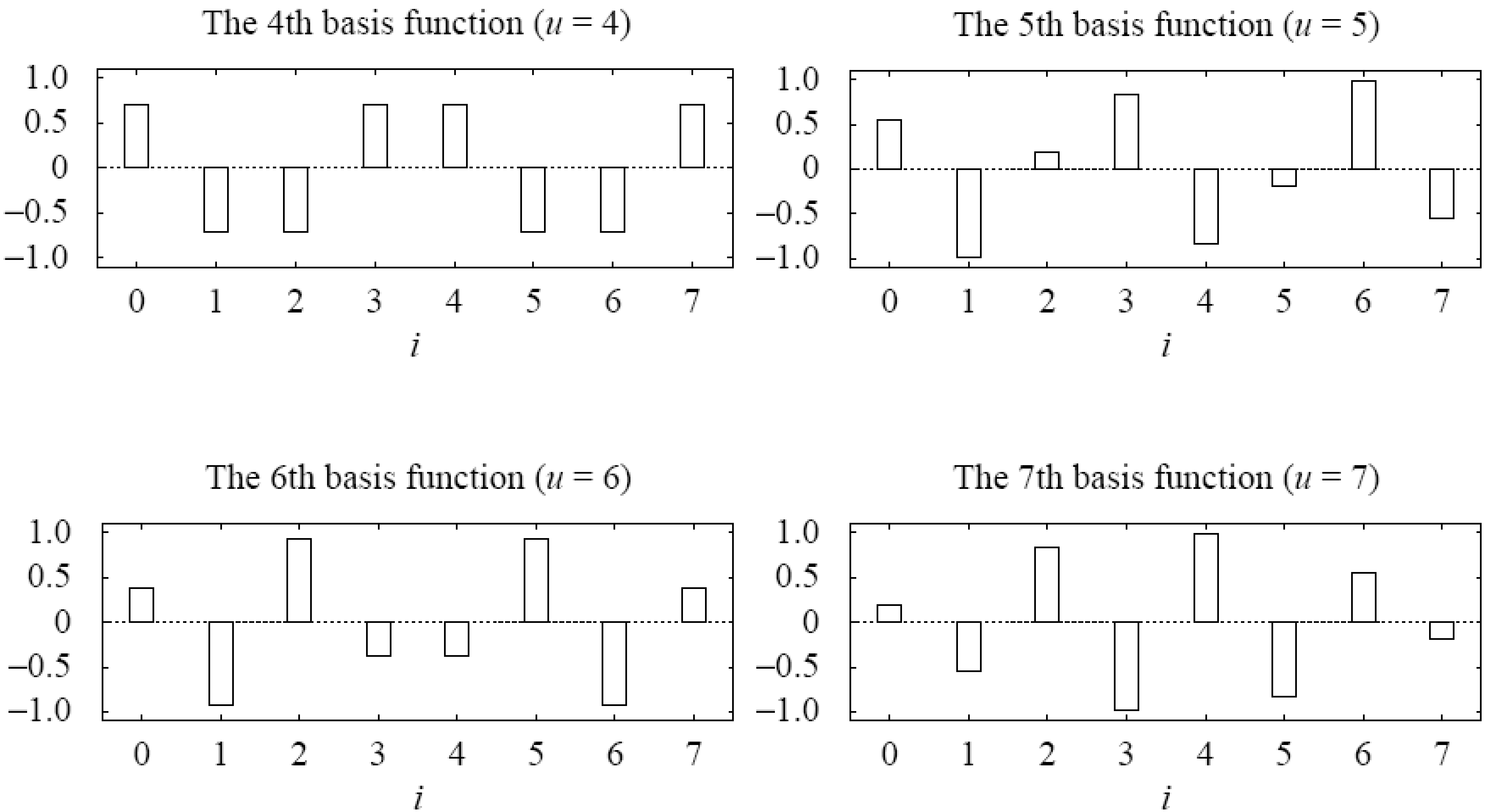
where $i, u = 0,1, \ldots, 7$.



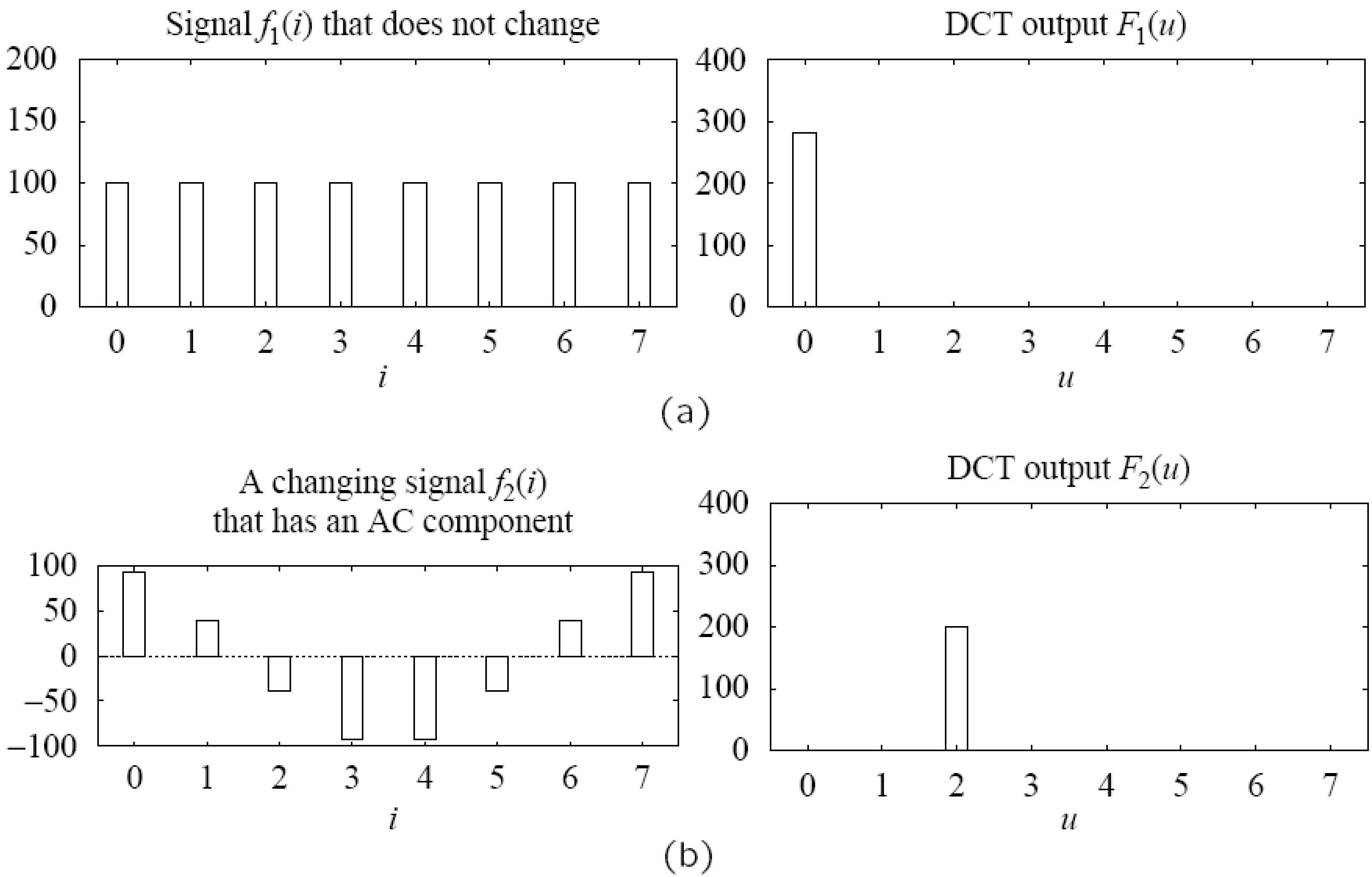
DCT basis function (M=8)



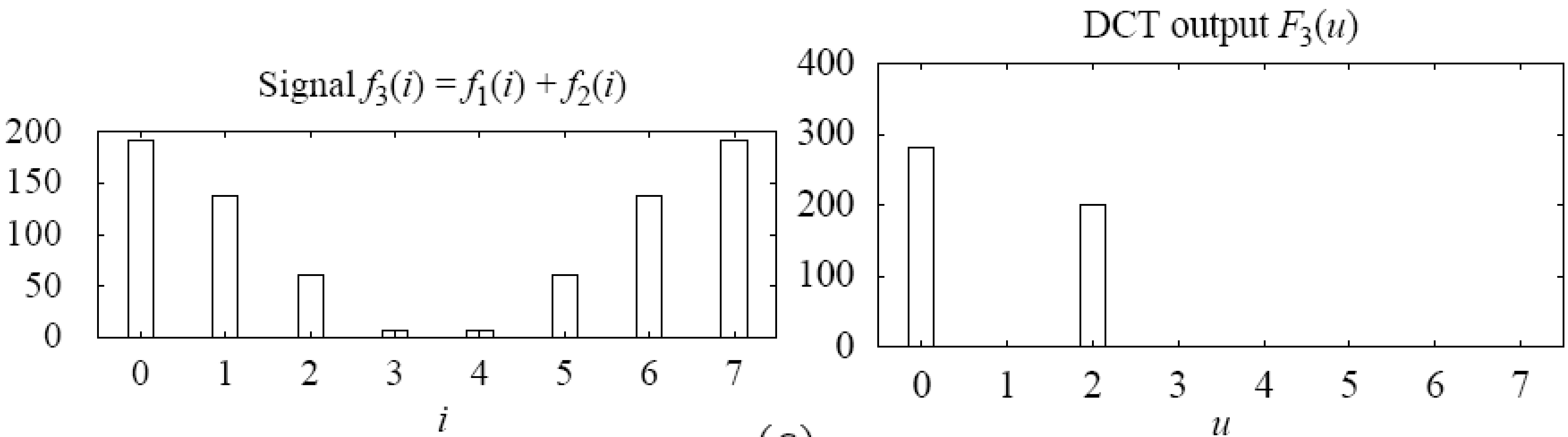
The 1D DCT basis functions.



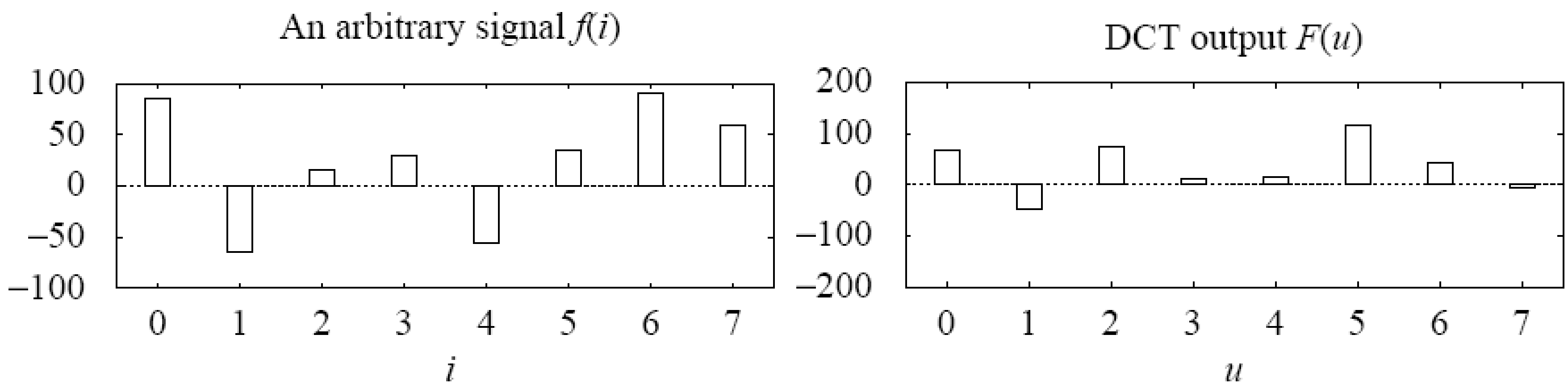
The 1D DCT basis functions.



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

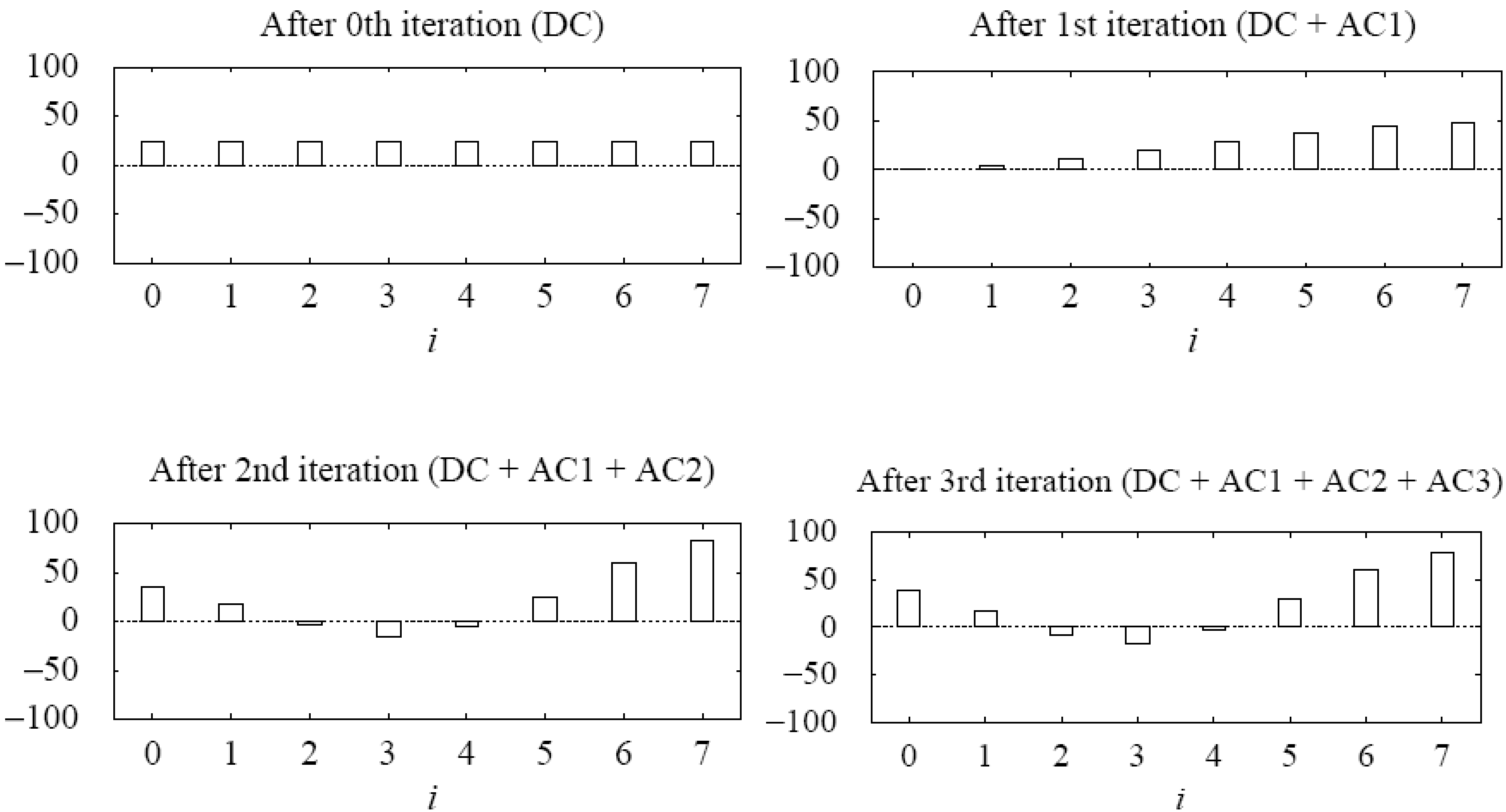


(c)

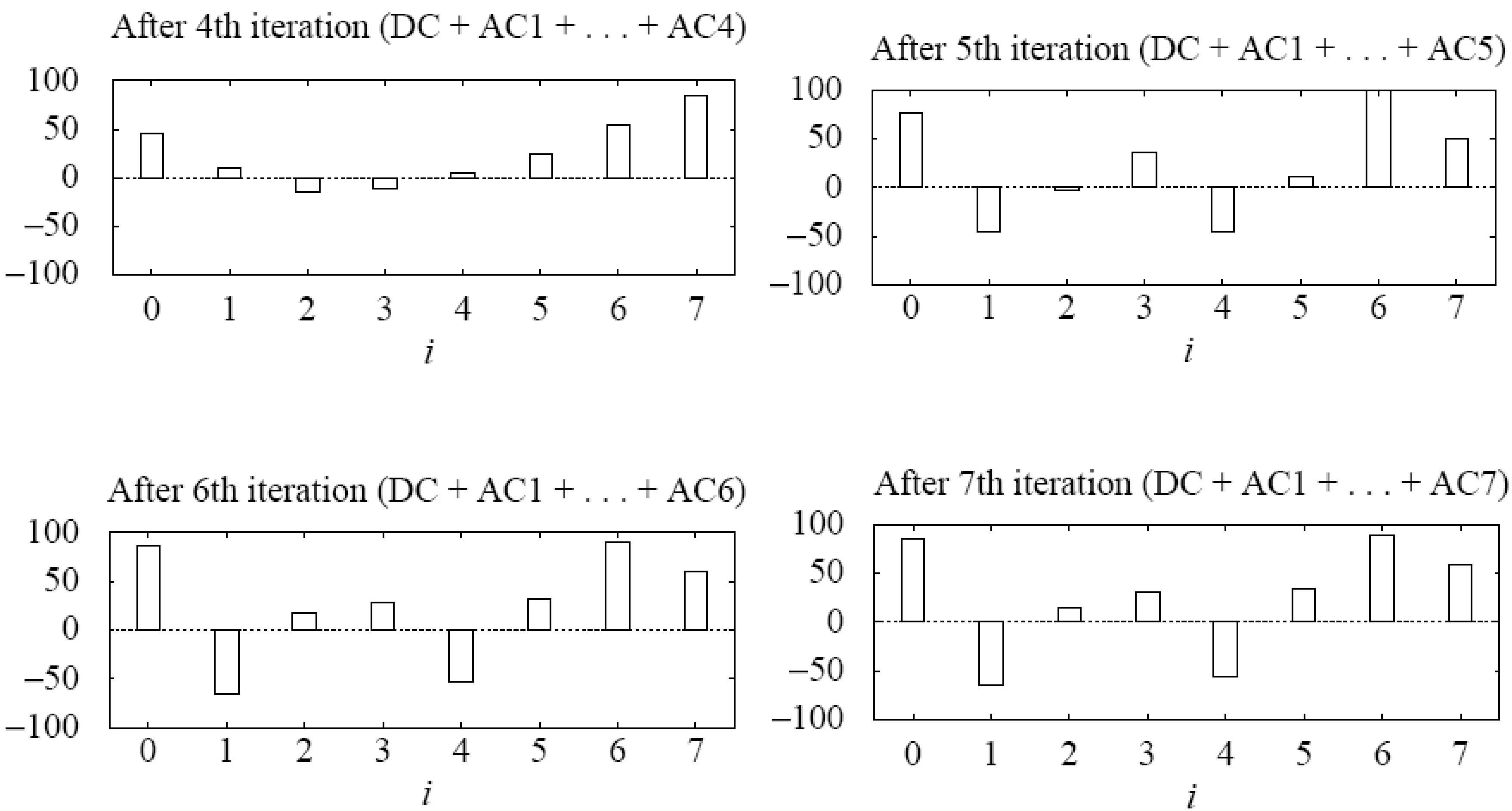


(d)

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

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

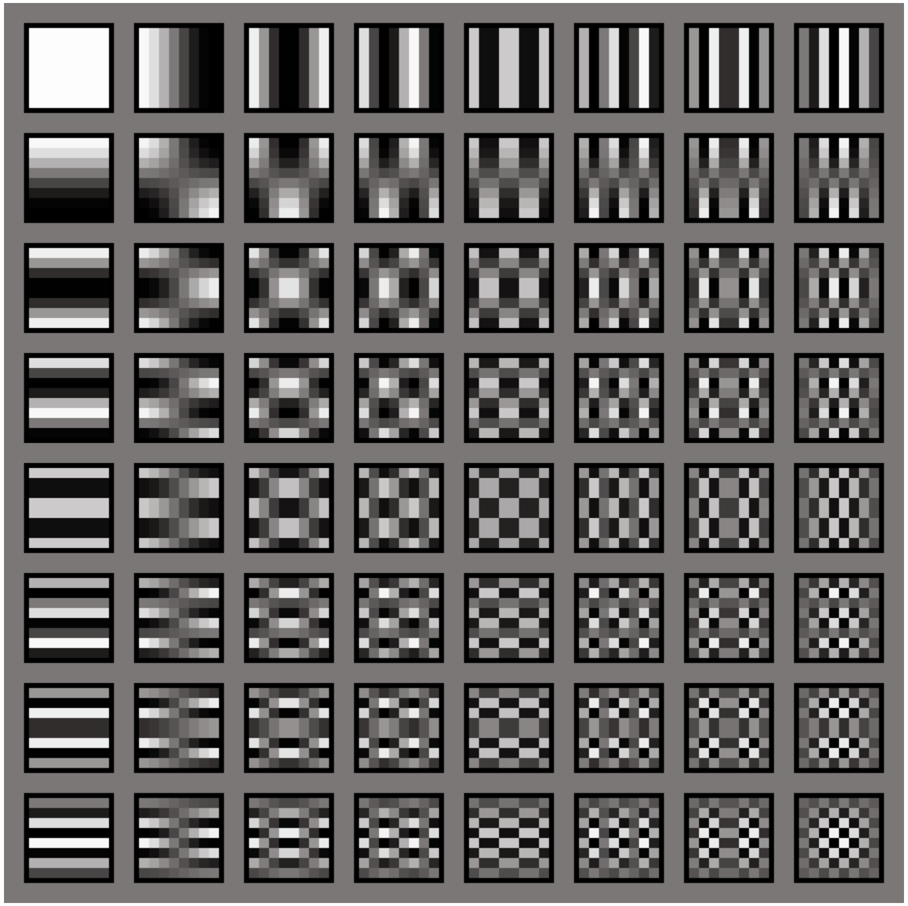
- Function $B_p(i)$ and $B_q(i)$ are *orthonormal*, if they are orthogonal and

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

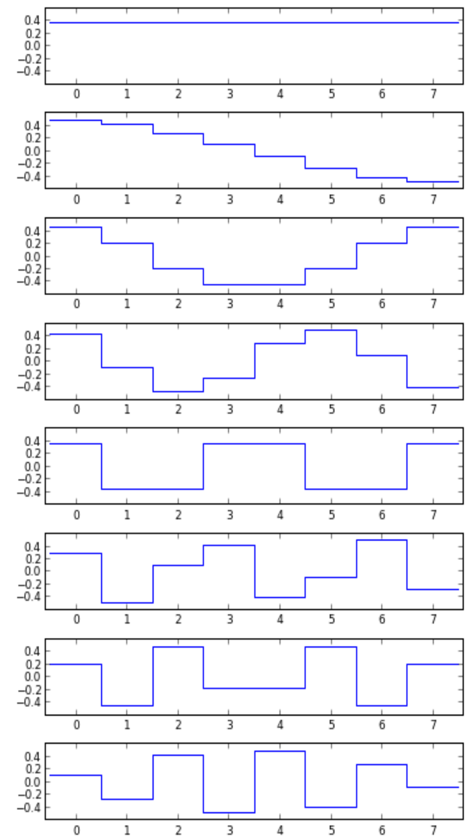
- It can be shown that:

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

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



Graphical Illustration of 8x8 2D DCT basis.



1D DCT basis

DCT on a Real Image Block

>>imblock = lena256(128:135,128:135)

imblock=

182	196	199	201	203	201	199	173
175	180	176	142	148	152	148	120
148	118	123	115	114	107	108	107
115	110	110	112	105	109	101	100
104	106	106	102	104	95	98	105
99	115	131	104	118	86	87	133
112	154	154	107	140	97	88	151
145	158	178	123	132	140	138	133

>>dctblock =dct2(imblock)

dctblock=1.0e+003*

1.0550	0.0517	0.0012	-0.0246	-0.0120	-0.0258	0.0120	0.0232
0.1136	0.0070	-0.0139	0.0432	-0.0061	0.0356	-0.0134	–0.0130
0.1956	0.0101	-0.0087	-0.0029	-0.0290	-0.0079	0.0009	0.0096
0.0359	-0.0243	-0.0156	-0.0208	0.0116	-0.0191	-0.0085	0.0005
0.0407	-0.0206	-0.0137	0.0171	-0.0143	0.0224	-0.0049	-0.0114
0.0072	-0.0136	-0.0076	-0.0119	0.0183	-0.0163	-0.0014	–0.0035
-0.0015	-0.0133	-0.0009	0.0013	0.0104	0.0161	0.0044	0.0011
-0.0068	-0.0028	0.0041	0.0011	0.0106	-0.0027	-0.0032	0.0016

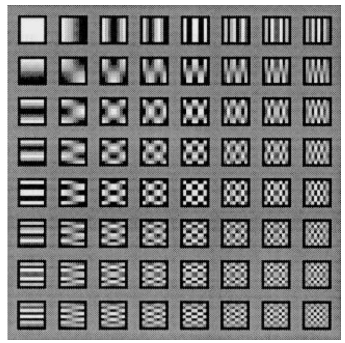
Low frequency coefficients (top left corner) are much larger than the rest!

DCT on a Real Image Block

imblock=

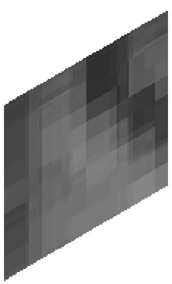
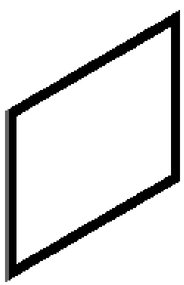
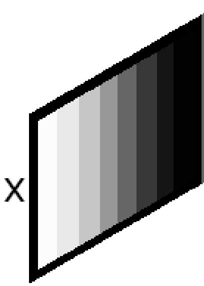
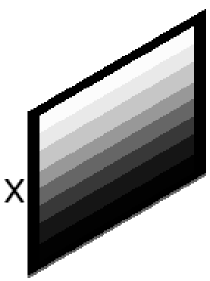
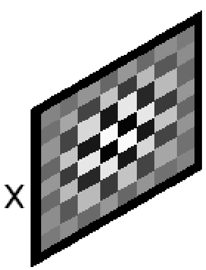
182	196	199	201	203	201	199	173
175	180	176	142	148	152	148	120
148	118	123	115	114	107	108	107
115	110	110	112	105	109	101	100
104	106	106	102	104	95	98	105
99	115	131	104	118	86	87	133
112	154	154	107	140	97	88	151
145	158	178	123	132	140	138	133

X



dctblock=1.0e+003*

1.0550	0.0517	0.0012	-0.0246	-0.0120	-0.0258	0.0120	0.0232
0.1136	0.0070	-0.0139	0.0432	-0.0061	0.0356	-0.0134	-0.0130
0.1956	0.0101	-0.0087	-0.0029	-0.0290	-0.0079	0.0009	0.0096
0.0359	-0.0243	-0.0156	-0.0208	0.0116	-0.0191	-0.0085	0.0005
0.0407	-0.0206	-0.0137	0.0171	-0.0143	0.0224	-0.0049	-0.0114
0.0072	-0.0136	-0.0076	-0.0119	0.0183	-0.0163	-0.0014	-0.0035
-0.0015	-0.0133	-0.0009	0.0013	0.0104	0.0161	0.0044	0.0011
-0.0068	-0.0028	0.0041	0.0011	0.0106	-0.0027	-0.0032	0.0016


= F(0,0) x

+ F(1,0) x

+ F(0,1) x

+............+ F(7,7) x


Original image block

DC basis function

AC basis functions

Reconstructed Block

Original block

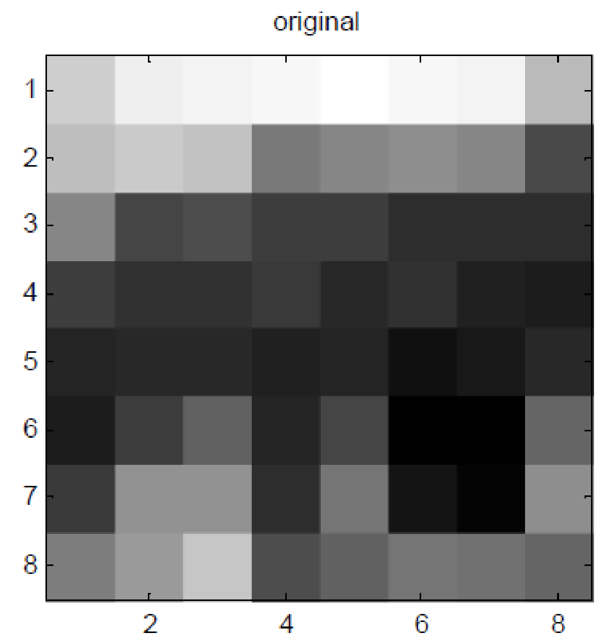
182	196	199	201	203	201	199	173
175	180	176	142	148	152	148	120
148	118	123	115	114	107	108	107
115	110	110	112	105	109	101	100
104	106	106	102	104	95	98	105
99	115	131	104	118	86	87	133
112	154	154	107	140	97	88	151
145	158	178	123	132	140	138	133

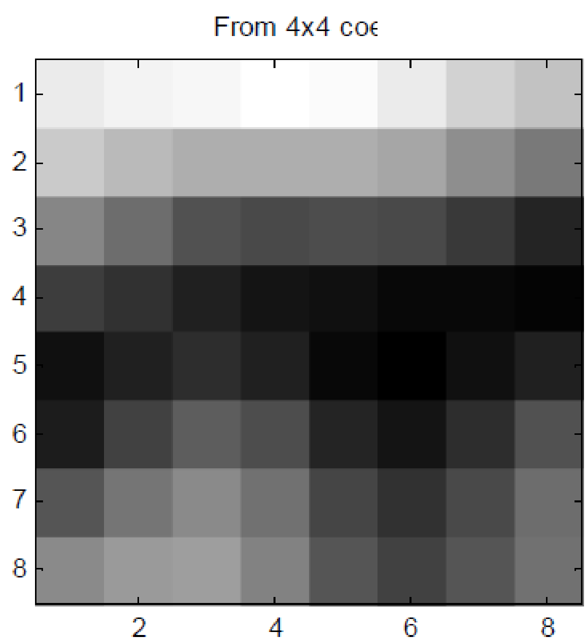
Reconstructed using top 4x4 coefficients

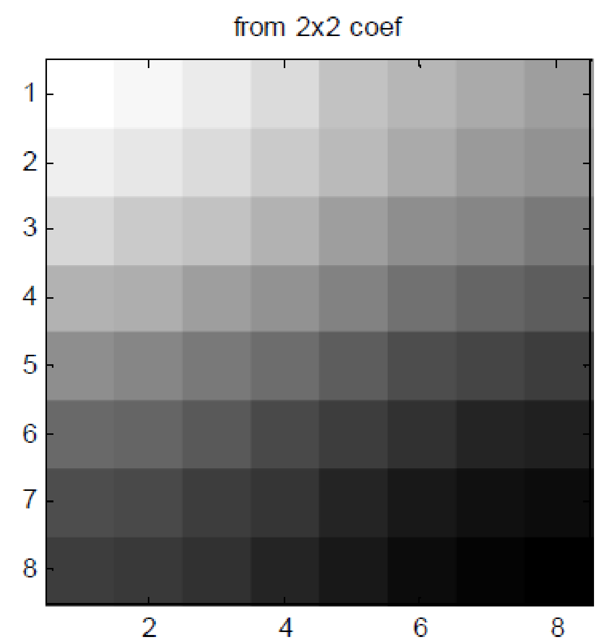
190	192	195	197	196	189	179	172
175	169	163	163	164	160	150	141
147	136	124	120	122	121	114	106
115	110	103	98	96	94	93	92
96	104	109	104	93	89	96	104
102	117	130	123	106	98	109	124
126	139	148	138	119	111	121	136
148	155	156	144	125	118	126	138

Reconstructed using top 2x2 coefficients only

162	161	158	154	149	146	143	141
159	157	154	151	147	143	140	138
153	151	149	145	141	137	135	133
145	144	141	138	134	131	128	126
137	135	133	130	126	123	121	119
129	128	125	122	119	116	114	113
123	122	119	117	114	111	109	108
119	118	116	114	111	108	106	105


original
1
2
3
4
5
6
7
8
2
4
6
8

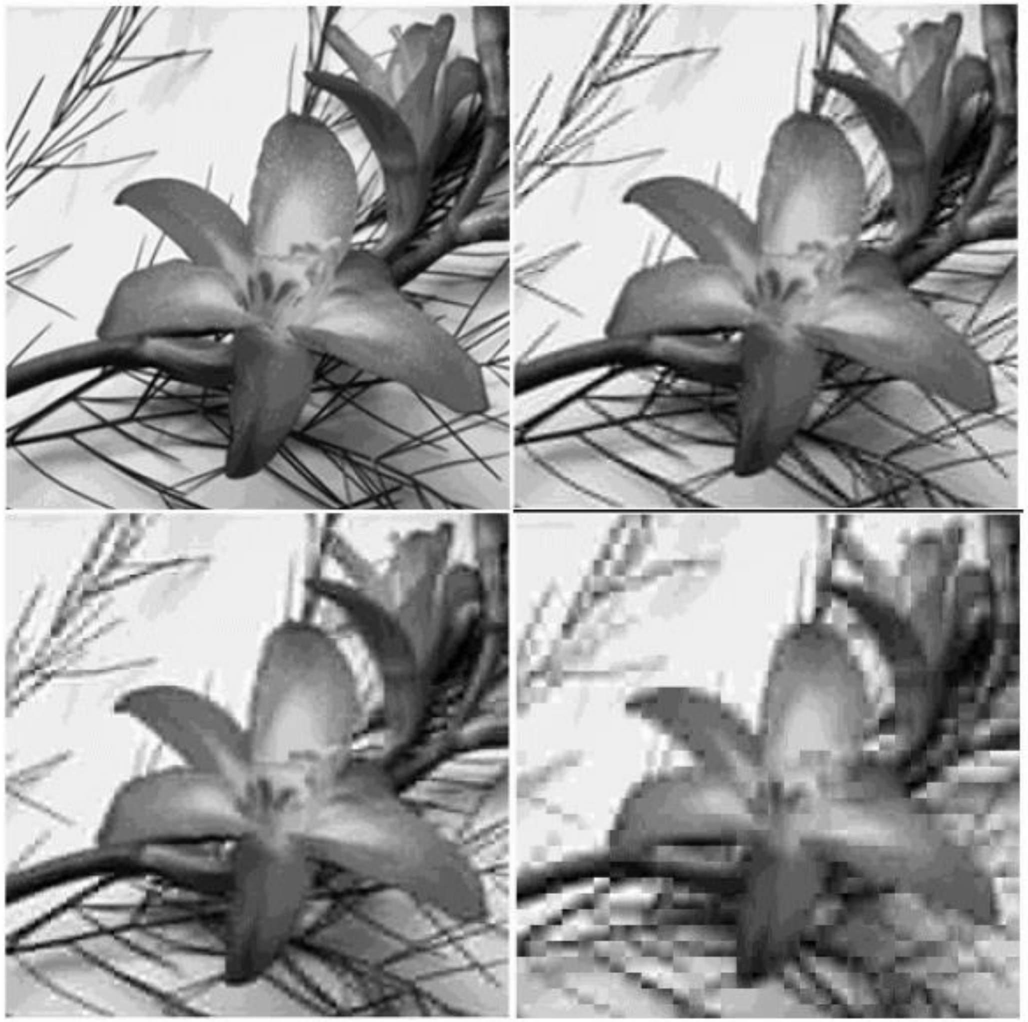

From 4x4 coe
1
2
3
4
5
6
7
8
2
4
6
8


from 2x2 coef
1
2
3
4
5
6
7
8
2
4
6
8

Approximation by DCT Basis

Original

With 16/64 Coefficients



With 8/64 Coefficients

With 4/64 Coefficients

Yao Wang, 2006

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:

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

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

- 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.