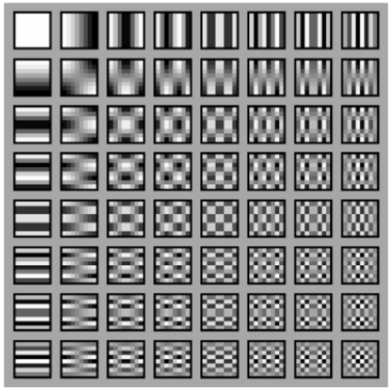
Multimedia Computing

Image Compression:

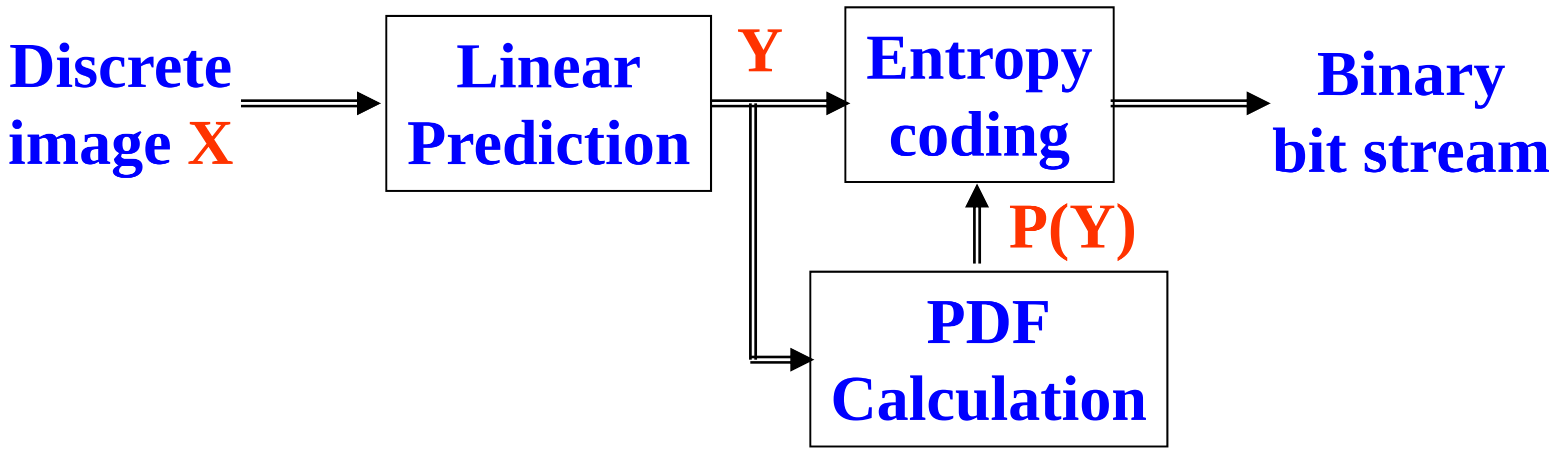
Part 1



Topics

- Lossless image compression
- Lossy image compression
  - Distortion and Quantization
  - Transform based coding
  - Wavelet based coding
- JPEG Standard

Lossless Predictive Coding (LPC)

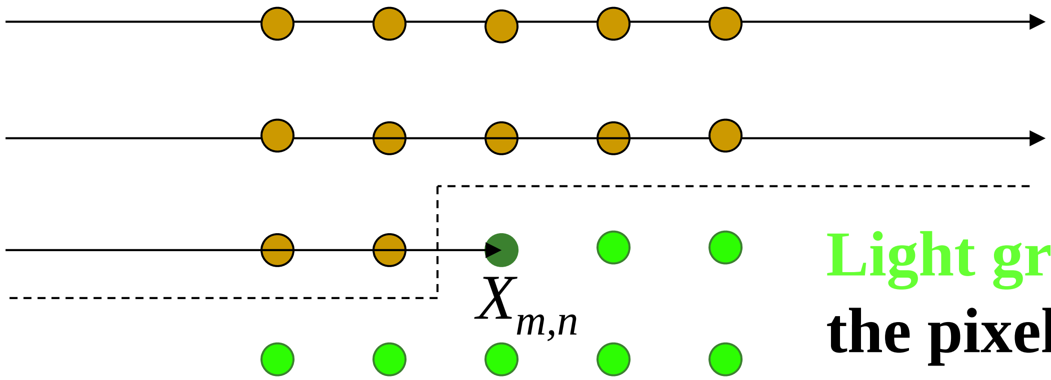


Prediction error sequence Y usually contains less entropy than the original sequence X

2D image LPC

Raster scanning order: left → right, top → bottom

Brown points:
available pixels



Light green points:
the pixels to be coded

Dark green point: the pixel being coded

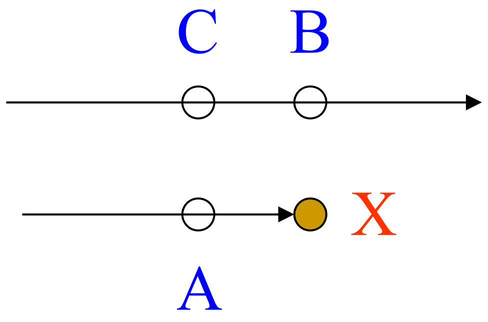
Predictor

$$\hat{X}_{m,n} = \sum_{k=1}^{K} a_k X_k, \forall m,n$$

Difference

$$Y_{m,n} = X_{m,n} - \hat{X}_{m,n}$$

Predictors in Lossless JPEG (JPEG-LS)

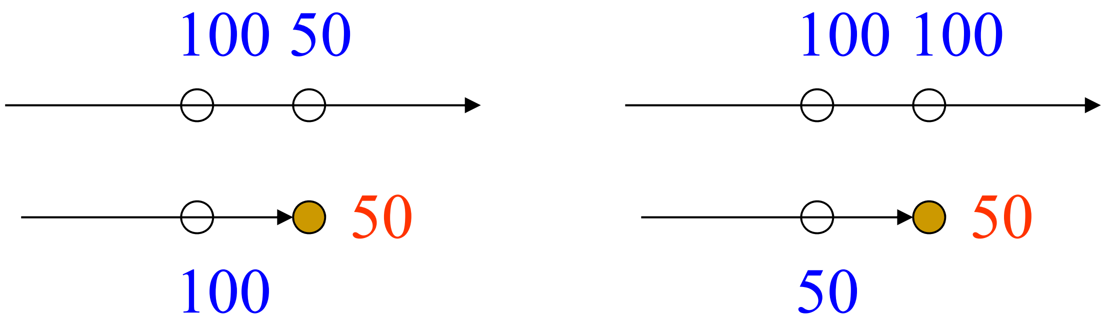


- JPEG-LS is a special case of JPEG image compression.
- We predict X using A, B and C.

Predictors in JPEG-LS

P1	A (horizontal predictor)
P2	B (vertical predictor)
P3	C
P4	A+B-C
P5	Median{A,B,A+B-C}
P6	$\lfloor A+(B-C)/2 \rfloor$
P7	$\lfloor B+(A-C)/2 \rfloor$
P8	$\lfloor (A+B)/2 \rfloor$

Comparing P1, P2 and P5: example



Case 1: vertical edge **Case 2: horizontal edge**

- With P1, the prediction errors for the two cases are -50 and 0, respectively.
- With P2, the prediction errors for the two cases are 0 and -50, respectively.
- With P5, the prediction errors for the two cases are both 0.
- Usually P5 works better than P1 and P2, of course the price is more computation.

A simple example

- Suppose we have the following image

10	12	13	14
12	15	14	15
11	16	18	13
13	10	11	12

- What is the entropy of this image?

A simple example

- The PDF of the image is

10	11	12	13	14	15	16	18
2/16	2/16	3/16	3/16	2/16	2/16	1/16	1/16

- The entropy is

$$\eta = -\sum_i p_i \log_2 p_i = 2.9056$$

A simple example

- We code the image as follows
  - Code the first row using P1
  - Code the first column using P2
  - Code the other pixels using P5
- Then the prediction error image is

10	2	1	1
2	3	-1	1
-1	2	3	-5
2	-6	-1	1

A simple example

- The PDF of the prediction error image is

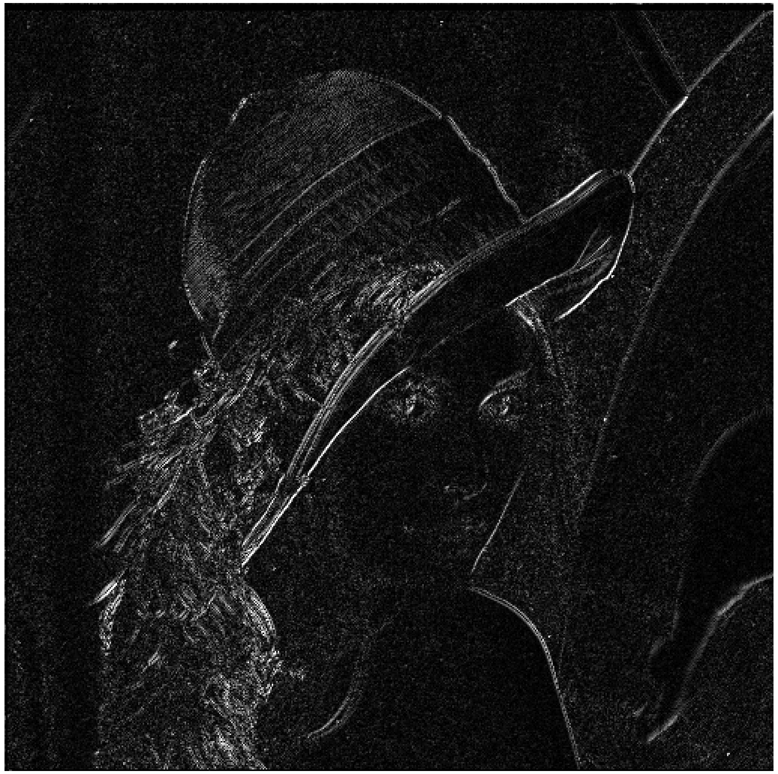
1	2	3	10	-1	-5	-6
4/16	4/16	2/16	1/16	3/16	1/16	1/16

- The entropy is

$$\eta = -\sum_i p_i \log_2 p_i = 2.5778$$

- The entropy of the prediction error is smaller.

Example: Lena



Vertical predictor P2
$\boldsymbol{\eta}$ = 4.67 bits

Predictor P5
$\boldsymbol{\eta}$ = 4.55 bits

Lab exercise

- Write a Matlab program to calculate the histogram (i.e. PDF) of the grey level image Lena, and then calculate its entropy.
- Use the P1 and P5 predictors in JPEG-LS, write a Matlab program to calculate the prediction error, then calculate the PDF and entropy of the prediction error.
- Plot the two PDFs, compare their shapes and the associated entropy values.

Topics

- Lossless image compression
- Lossy image compression
  - Distortion and Quantization
  - Transform based coding
  - Wavelet based coding
- JPEG Standard

Lossy Compression

- 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:

  1. Mean Square Error (MSE)

$$\sigma_e^2 = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(I(i,j) - \hat{I}(i,j) \right)^2$$

I : *The original image*

$\hat{I}$: *The reconstructed image after compression*

Distortion Measures

2. Signal to Noise Ratio (SNR)

$$SNR = 10\log_{10}\frac{\sigma_I^2}{\sigma_e^2}$$

σ_I^2 : *The mean square value of the original image*

$$\sigma_I^2 = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M} I^2(i, j)$$

σ_e^2: *The MSE of the reconstructed image after compression*

Distortion Measures

2. Peak Signal to Noise Ratio (PSNR)

$$SNR = 10\log_{10}\frac{x_I^2}{\sigma_e^2}$$

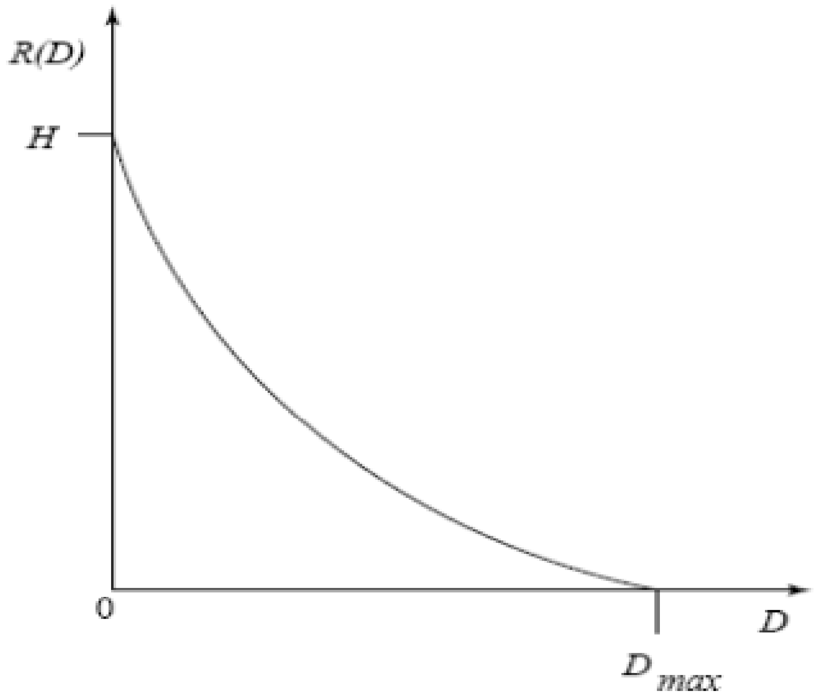
x_I : *The peak value of the original image I*

Usually we let $x_I = 255$ *for digital images*

σ_e^2: *The MSE of the reconstructed image after compression*

The Rate-Distortion Theory

- Provides a framework for the study of tradeoffs between code Rate and signal/image Distortion.



A typical rate-distortion function

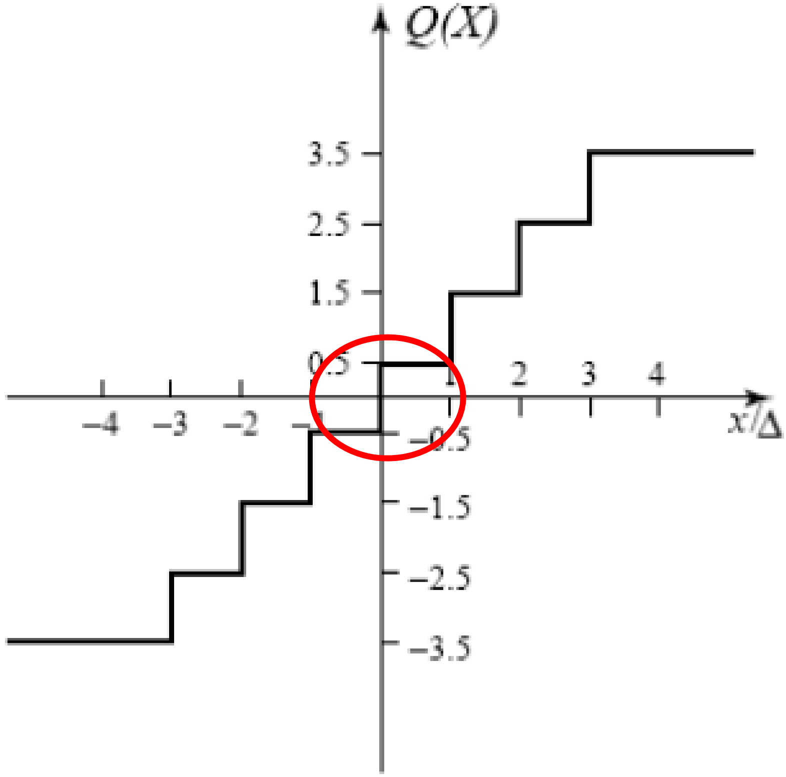
Quantization

- Reduce the number of distinct output values to a much smaller set.
- The main source of the "loss" in lossy compression.
- Three different forms of quantization.
  - Uniform Quantization
  - Non-uniform Quantization
  - Vector Quantization

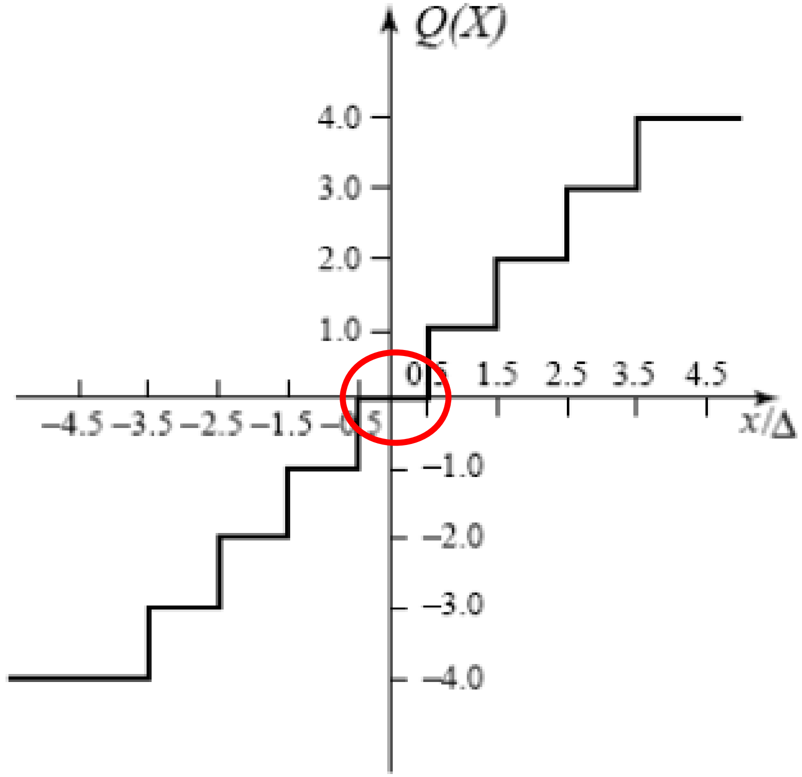
Uniform Scalar Quantization

- A uniform scalar quantizer partitions the domain of input values into equally spaced intervals, except possibly at the two outer intervals.
  - The output or reconstruction value corresponding to each interval is taken to be the midpoint of the interval.
  - The length of each interval is referred to as the step size, denoted by the symbol Δ.
- Two common types of uniform scalar quantizers:
  - Midrise quantizers have even number of output levels.
  - Midtread quantizers have odd number of output levels, including zero as one of them.

Midrise and Midtread Quantizers



Midrise



Midtread

Some special cases (*optional*)

- For the special case where $\Delta = 1$, we can simply compute the output values for these quantizers as:

$$Q_{midrise}(x) = \lceil x \rceil - 0.5$$

$$Q_{midtread}(x) = \lfloor x + 0.5 \rfloor$$

- For M level quantizer, suppose the input is uniformly distributed in the interval $[-X_{max}, X_{max}]$. The rate of the quantizer is:

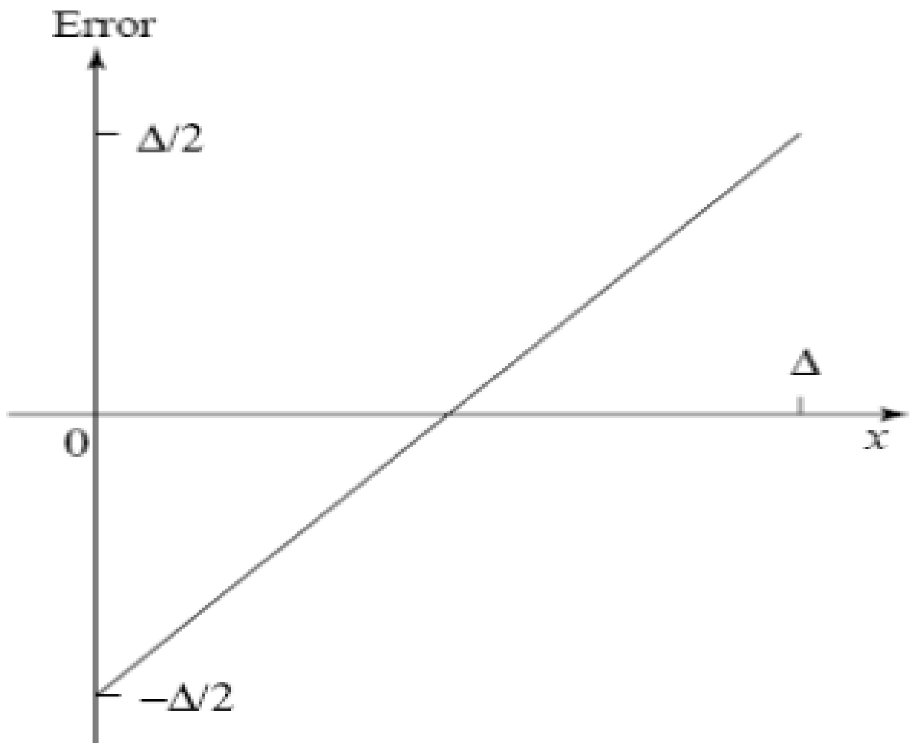
$$R = \lceil \log_2 M \rceil$$

Quantization Error of Uniformly Distributed Source (*optional*)

- Granular distortion: quantization error caused by the quantizer for bounded input.
  - To get an overall figure for granular distortion, notice that decision boundaries b_i for a midrise quantizer are $[(i − 1)\Delta, i\Delta]$, $i = 1,2,\dots,M/2$, covering positive data X (and another half for negative X values).
  - Output quantized values y_i are the midpoints $i\Delta-\Delta/2$, $i = 1,2,\dots,M/2$, again just considering the positive data. The total distortion is twice the sum over the positive data, or

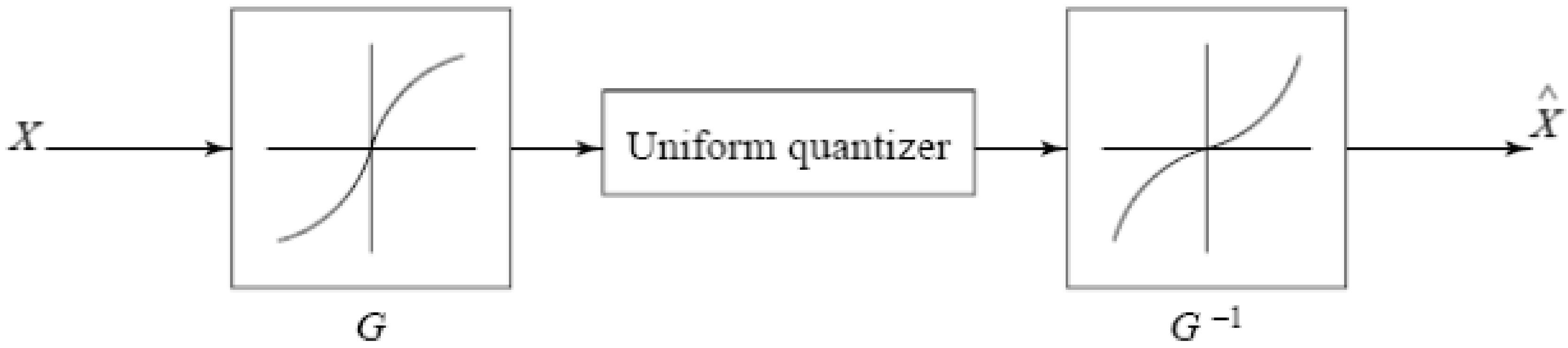
$$D_{gran} = 2\sum_{i=1}^{\frac{M}{2}} \int_{(i-1)\Delta}^{i\Delta} \left(x - \frac{2i-1}{2}\Delta\right)^2 \frac{1}{2X_{max}}\ dx$$

- Since the reconstruction values y_i are the midpoints of each interval, the quantization error must lie within the values $[-\Delta/2, \Delta/2]$.
- For a uniformly distributed source, the graph of the quantization error is as follows



Companded quantization (*optional*)

- Companded quantization is nonlinear.
- A *compander* consists of a *compressor function G*, a uniform quantizer, and an *expander function* G^{-1}.
- The two commonly used companders are the μ-law and *A*-law companders.



Vector Quantization (*optional*)

- According to Shannon's original work on information theory, any compression system performs better if it operates on vectors or groups of samples rather than individual symbols or samples.
- Form vectors of input samples by simply concatenating a number of consecutive samples into a single vector.
- Instead of single reconstruction values as in scalar quantization, in vector quantization (VQ) code vectors with *n* components are used. A collection of these code vectors form the *codebook*.

Topics

- Lossless image compression
- Lossy image compression
  - Distortion and Quantization
  - Transform based coding
  - Wavelet based coding
- JPEG Standard

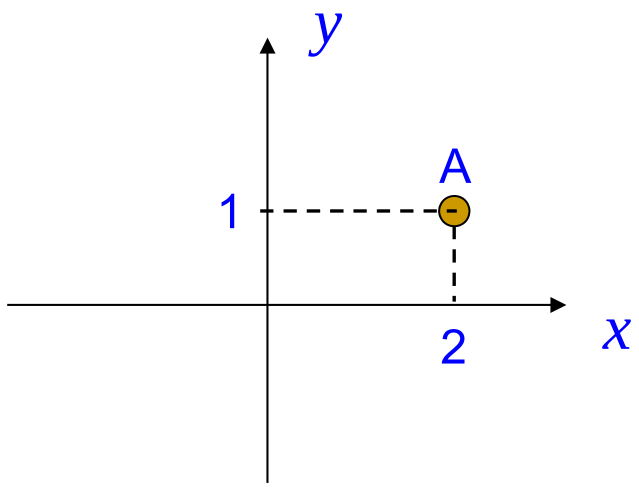
Transform Coding

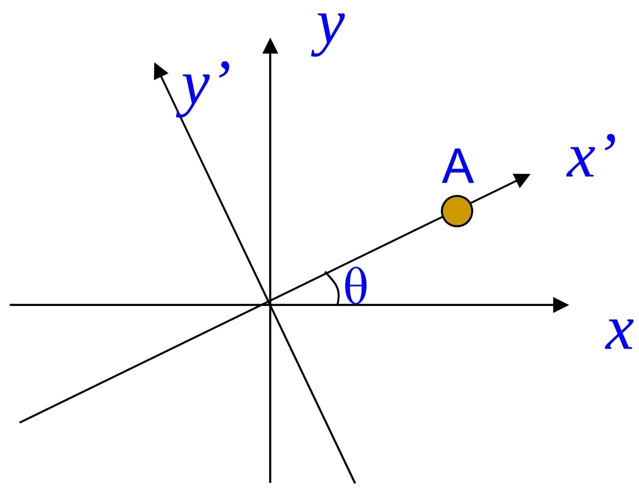
- Why transform?
  - If we transform the input signal X into Y using a linear transform T such that the components of Y are much less correlated, then Y can be coded more efficiently than X.
  - If most information can be accurately described by only few components of a transformed vector, then the remaining components can be coarsely quantized, or even set to zero, with little signal distortion.
  - We will first study Discrete Cosine Transform (DCT) and Karhunen-Loève Transform (KLT), and then study Wavelet Transform (WT).

How does transformation works?

- By transformation, we can view the same thing in different worlds (domains), e.g. from the Yang (阳) domain to Yin (阴) domain in Chinese philosophy.
- We can view the ordinary representation (time/spatial/temporal) of signals/images/videos as in the Yang domain, and the representation in the transformed world as in the Yin domain.
- Many times, an event can be better represented in the other domains with less cost.
- The predictive coding is actually a transformed coding because we transform the original signal into the difference domain.
- One key point in transform based coding is that the original data can be transformed back from the transformed domain.

An example





- Suppose in the (x, y) coordinate world, the point A is represented by (2,1), we need two numbers to index it.
- We can transform the "world" by using a rotation transformation as follows, and then in the new (x', y') world the point A can be represented as (sqrt(5), 0) → only one number is needed.

$$\begin{bmatrix} A_{x'} \\ A_{y'} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} A_x \\ A_y \end{bmatrix}$$

Discrete Cosine Transform (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 (direct current) and AC (alternative current) components; the role of the Inverse DCT (IDCT) is to *reconstruct* (re-compose) the signal.

Definition of DCT

- Given an input function *f*(*i*,*j*) over two integer variables *i* and *j*, 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 DCT 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* = 0,1,2,…, *M*−1; *j, v* = 0,1,2,…,*N* − 1; and the constants *C*(*u*) and *C*(*v*) are determined by

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

2D DCT and 2D IDCT

2D DCT

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

where i, j, u, v = 0,1,…,7.

2D Inverse DCT (2D IDCT)

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

where i, j, u, v = 0,1,…,7.

1D DCT and 1D IDCT

1D DCT

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

where *i, u* = 0,1,…,7.

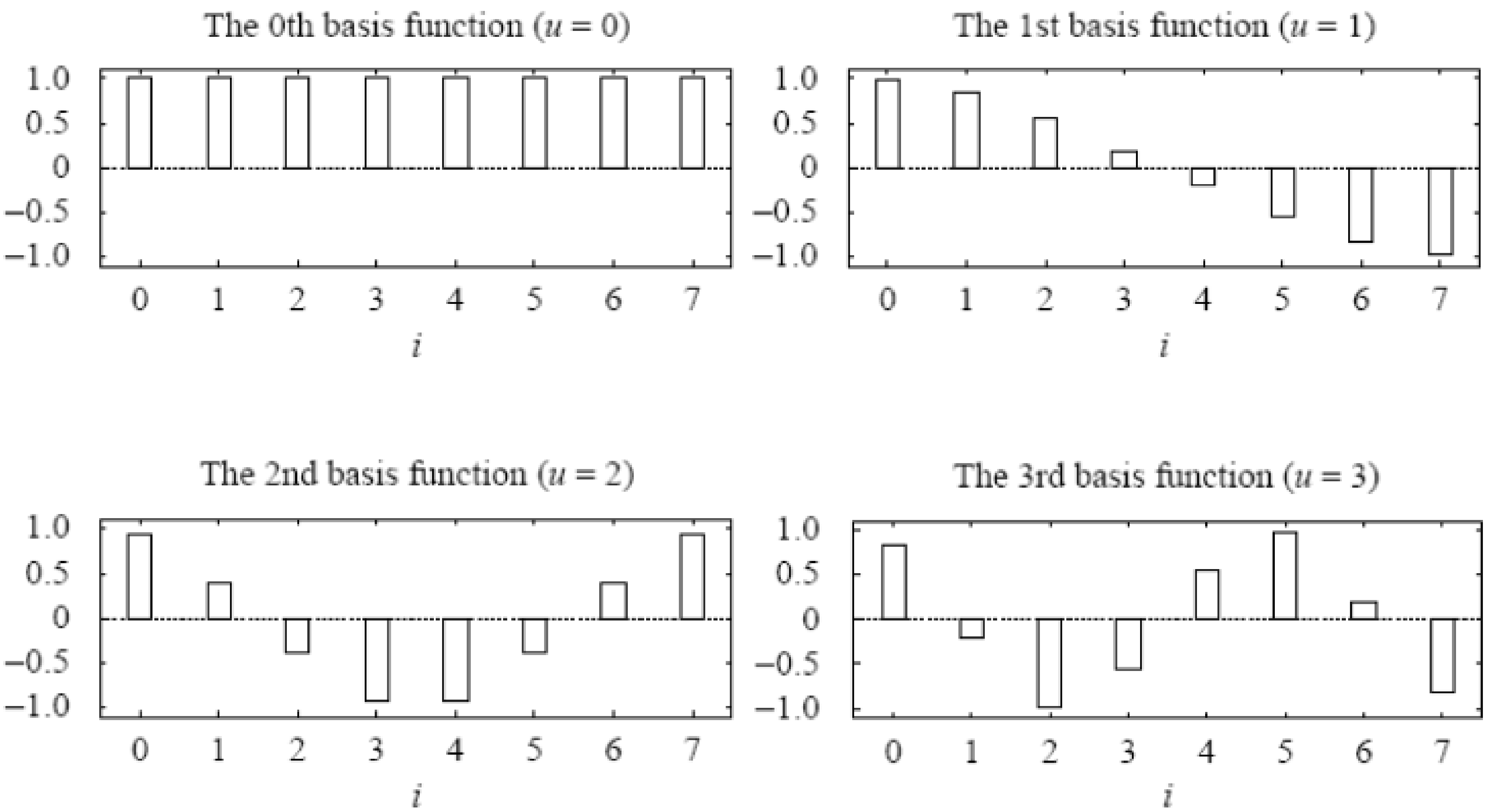
1D Inverse DCT (1D IDCT)

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

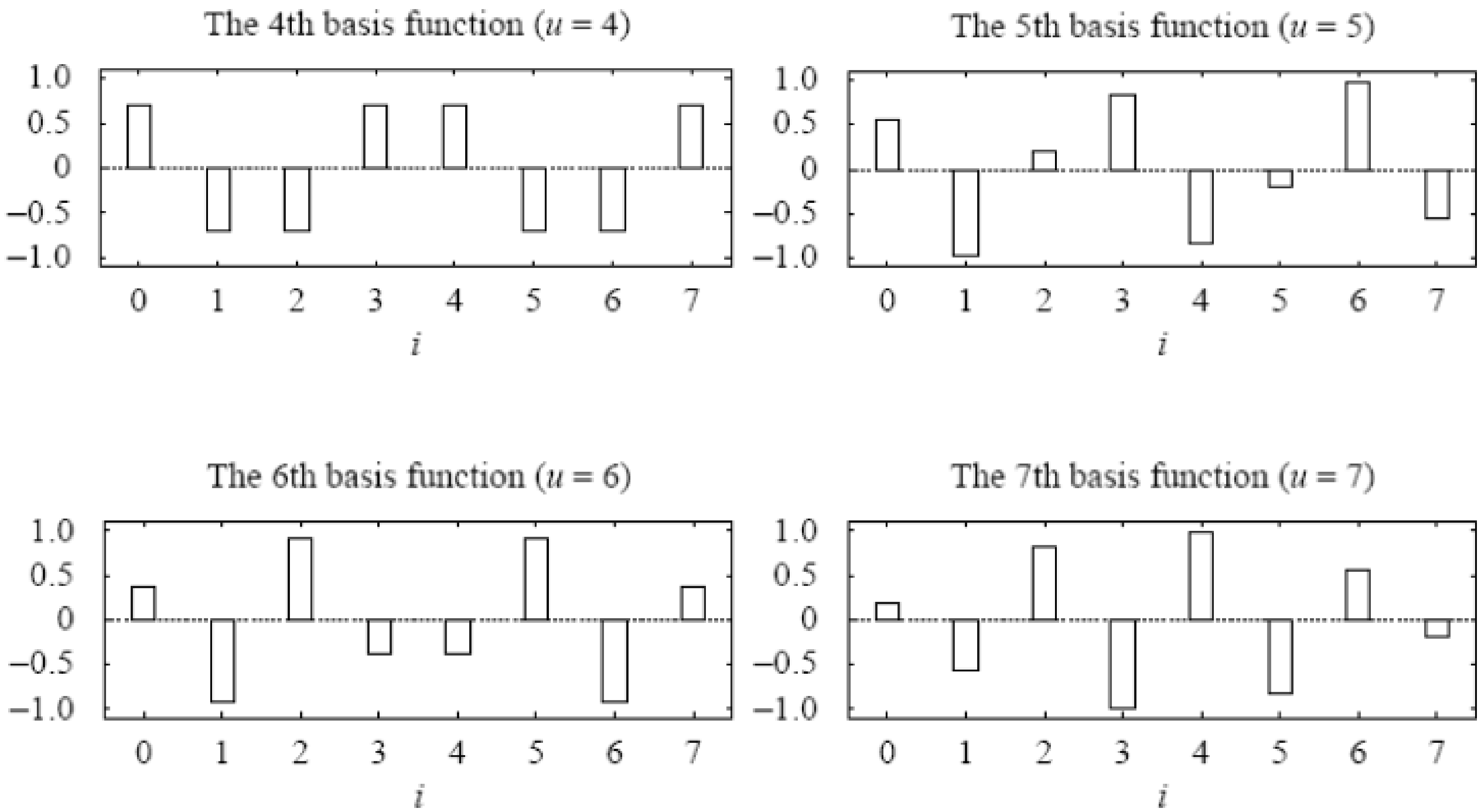
where *i, u* = 0,1,…,7.

Almost all properties of 1D DCT can be readily extended to 2D DCT.

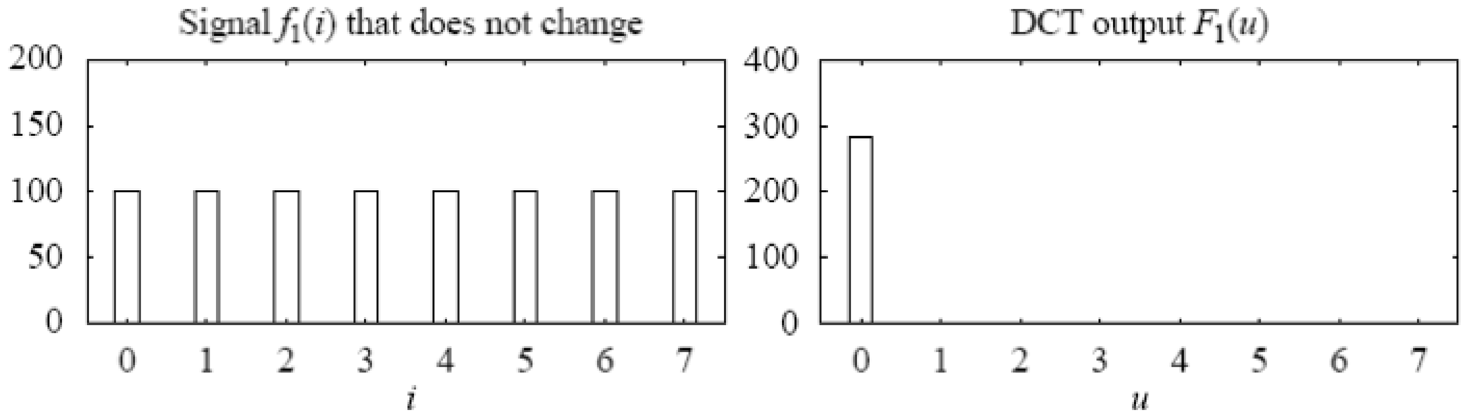
1D DCT basis functions: cos(•)



1D DCT basis functions: cos(•)



Example



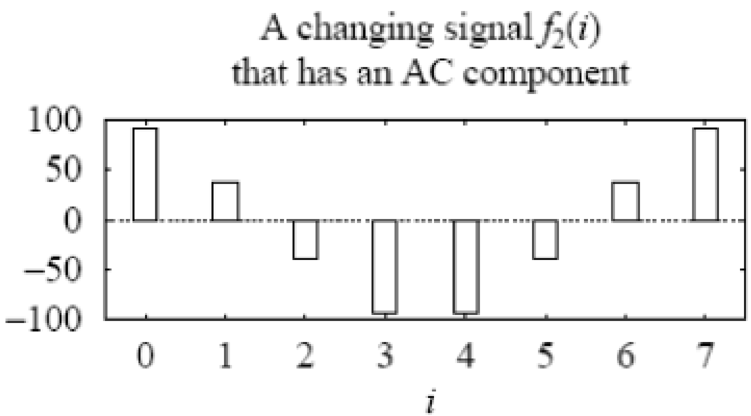
Constant signal f_1=[100, 100, …,100]. Its DCT is

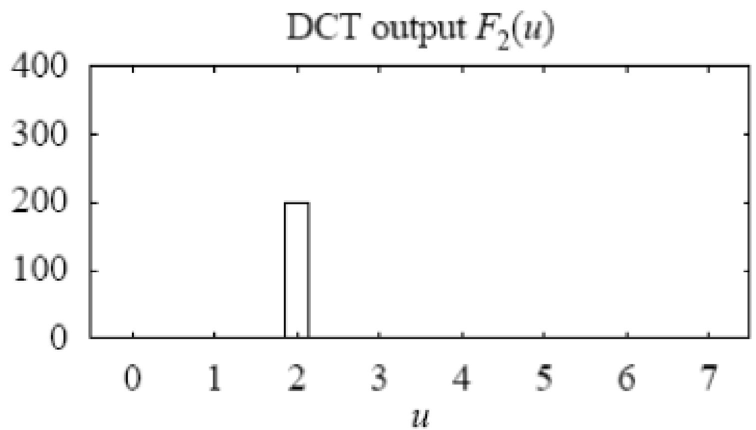
$$F_1(0) = \frac{\sqrt{2}}{2\cdot 2}\left(1\cdot 100 + 1\cdot 100 + \ldots + 1\cdot 100\right) \approx 283$$

$$F_1(1) = \frac{1}{2}\left(\cos\frac{\pi}{16}\cdot 100 + \cos\frac{3\pi}{16}\cdot 100 + \ldots + \cos\frac{15\pi}{16}\cdot 100\right) = 0$$

$$F_1(2) = F_1(3) = \ldots = F_1(7) = 0$$

Example





Signal f_2 is a discrete cosine signal

$$f_2(i) = 100\cos\left((2i+1)\pi/8\right)$$

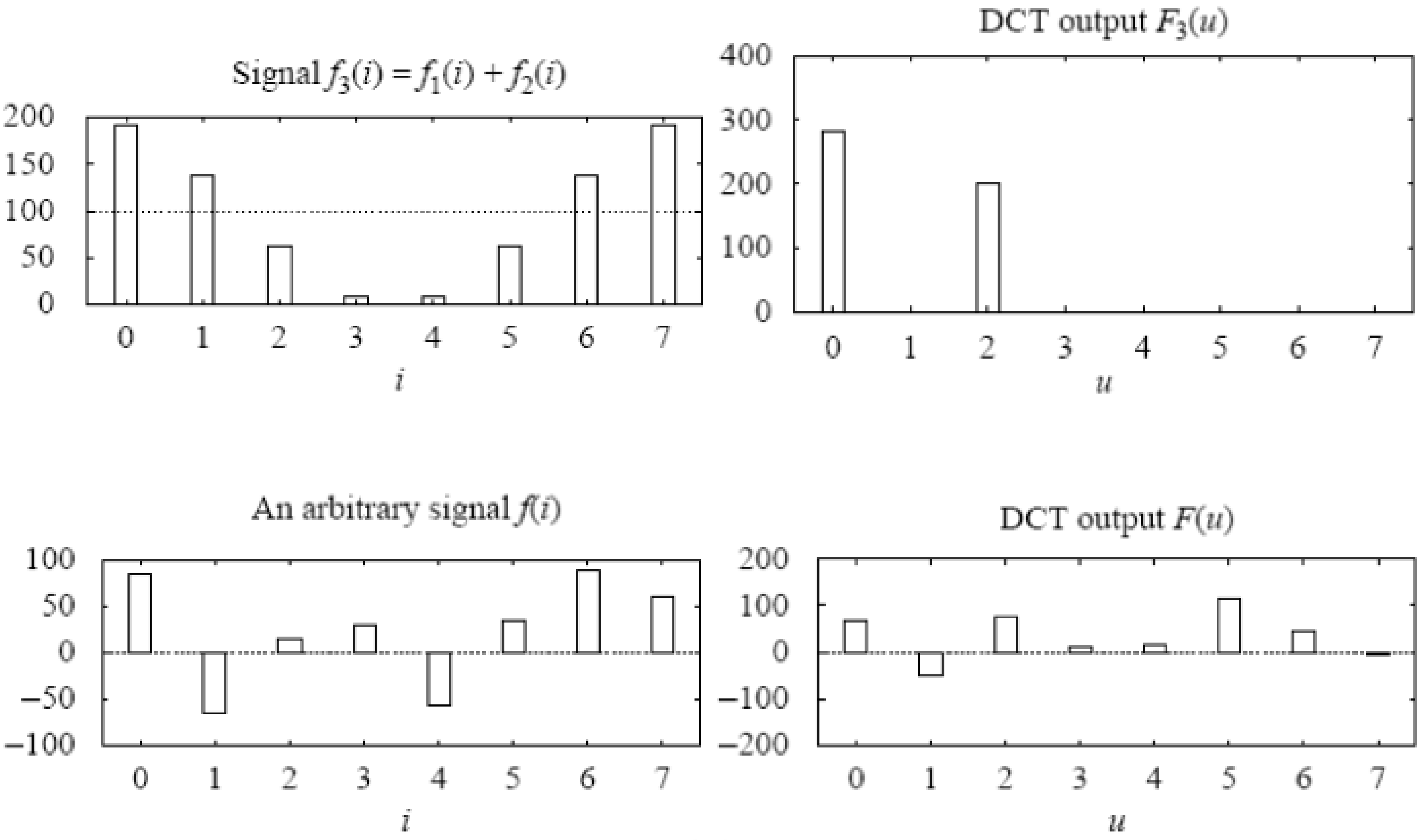
Its DCT is

$$F_2(0) = \frac{\sqrt{2}}{2\cdot 2}\left(100\cos\frac{\pi}{8} + 100\cos\frac{3\pi}{8} + \ldots + 100\cos\frac{15\pi}{8}\right) = 0$$

$$F_2(1) = F_2(3) = F_2(4) = \ldots = F_2(7) = 0$$

$$F_2(2) = \frac{1}{2}(\ldots) = 200$$

Example



Signal $f_3(i) = f_1(i) + f_2(i)$
DCT output $F_3(u)$
An arbitrary signal $f(i)$
DCT output $F(u)$

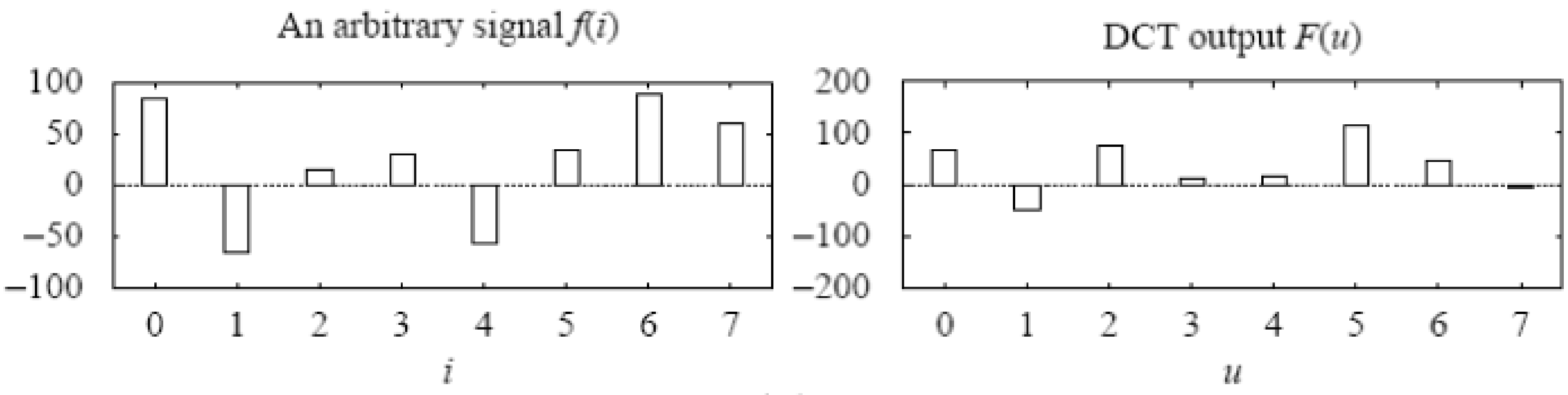
DCT is a linear transform

- In general, a transform T (or function) is *linear*, if and only if

$$T(\alpha p + \beta q) = \alpha T(p) + \beta T(q)$$

  where α and β are constants, p and q are any functions, variables or constants.
- From the definition of DCT, we can easily prove that DCT is a linear transform because it uses only simple arithmetic operations.

Example of 1D IDCT



f(i): 85 -65 15 30 -56 35 90 60

F(u): 69 -49 74 11 16 117 44 -5

Can we recover the original signal f(i) from its DCT F(u)?

Example of 1D IDCT

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

We can see that IDCT can be implemented as a loop with eight iterations.

Iteration 0 (DC)

$$\tilde{f}(i) = \frac{C(0)}{2} \cdot \cos(0) \cdot F(0) = \frac{\sqrt{2}}{2 \cdot 2} \cdot 1 \cdot 69 \approx 24.3$$

Iteration 1

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

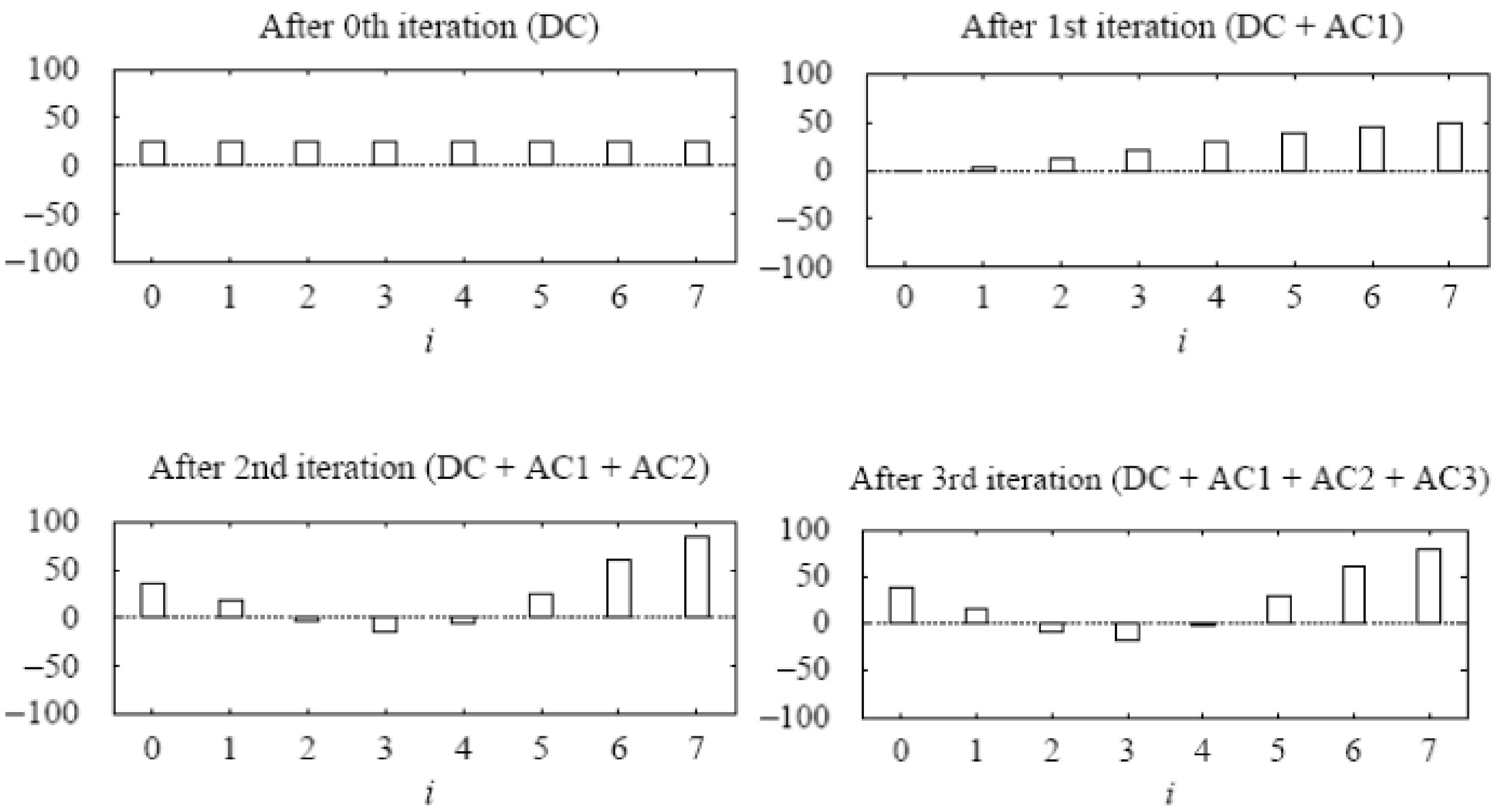
$$\approx 24.3 - 24.5 \cos \frac{(2i+1)\pi}{16}$$

Iteration 2

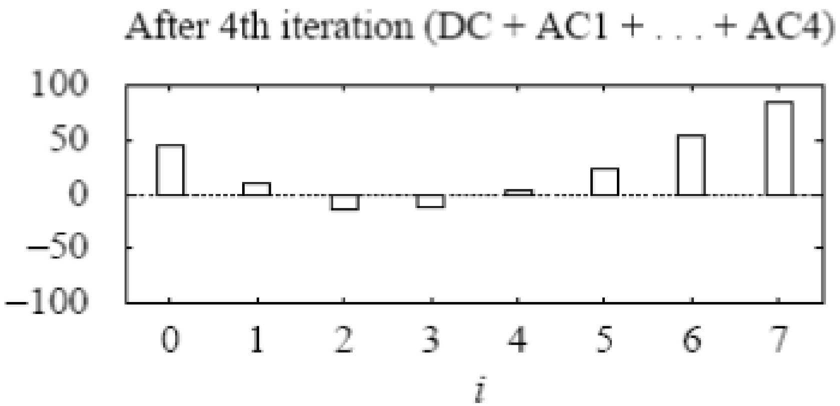
$$\tilde{f}(i) \approx 24.3 - 24.5 \cos \frac{(2i+1)\pi}{16} + 37 \cos \frac{(2i+1)\pi}{8}$$

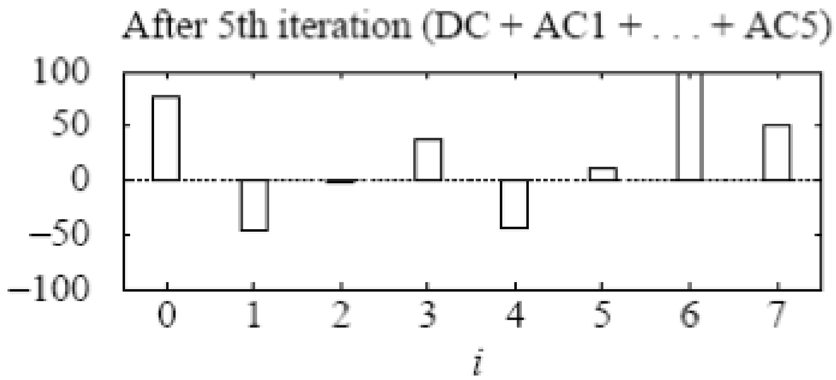
……

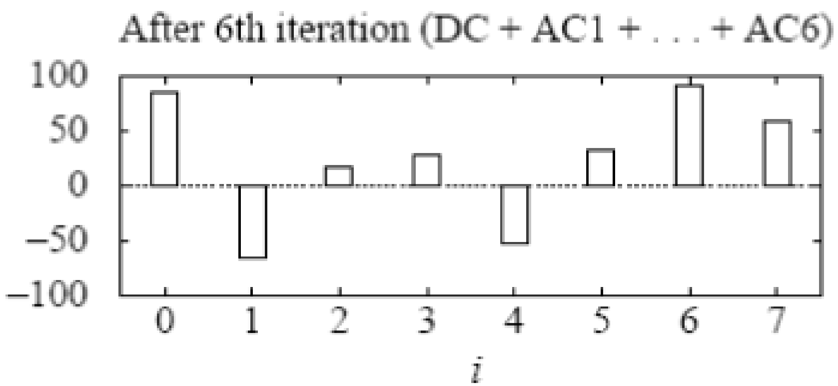
Example of 1D IDCT

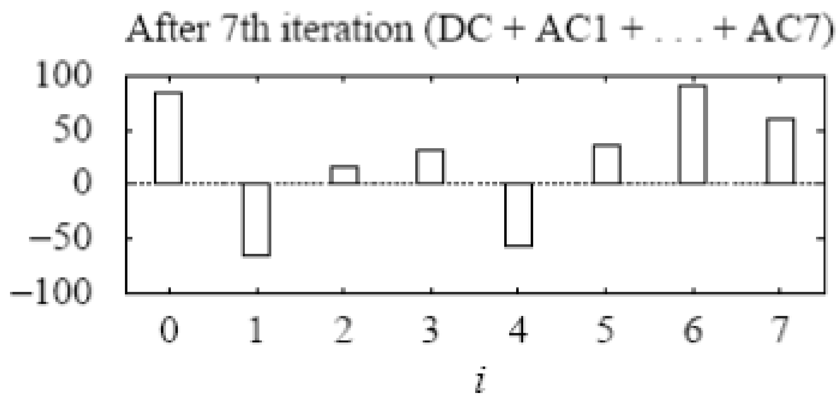


Example of 1D IDCT









Orthonormality of Cosine Bases

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

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

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

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

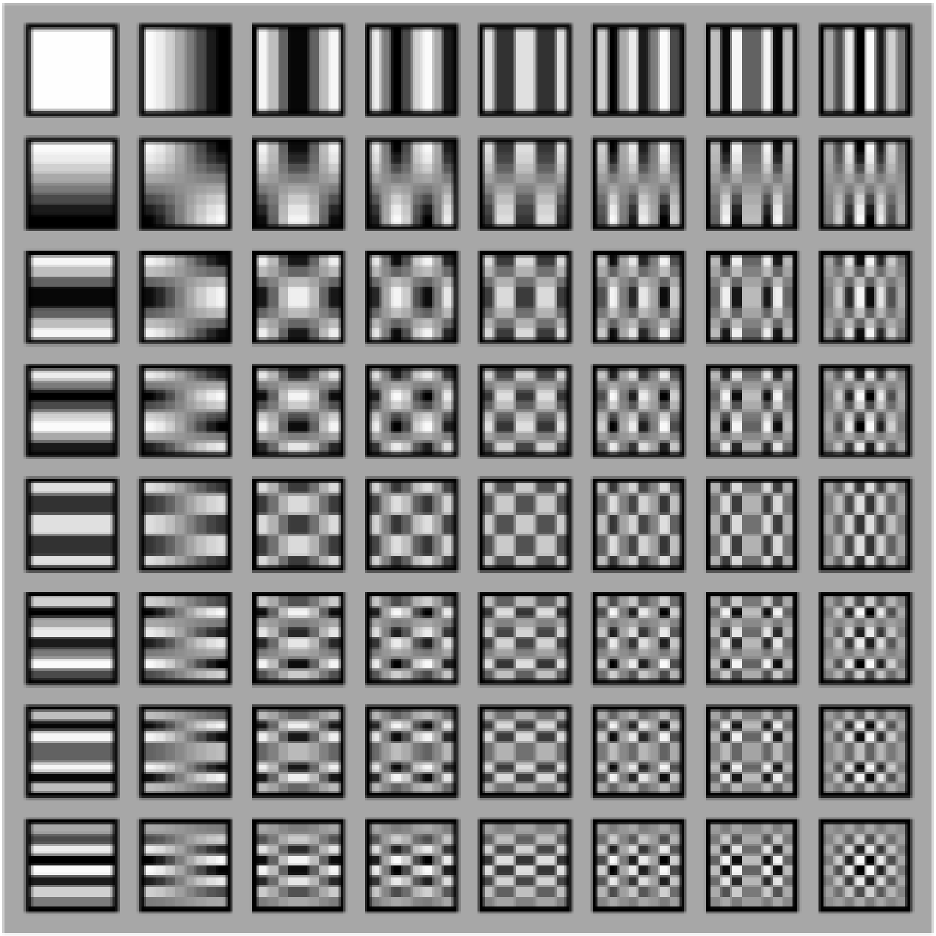
- It can be shown that cosine functions are orthonormal:

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

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

2D DCF basis functions

- In 1D DCT, we have 8 basis functions; in 2D DCT, we have 64 basis functions, each is a 8×8 spatial frequency image F(u,v).



2D Separable Basis

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

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

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

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

Karhunen-Loève Transform (KLT) *(optional)*

- The KLT 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 R_X of the input vector **X** defined as

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

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

KLT

- Our goal is to find a transform T such that the components of the output Y are uncorrelated, i.e. $E[Y_t Y_s] = 0$, if $t \neq s$. Thus, the autocorrelation matrix of 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 $u_1, u_2, .., u_k$, and *k* corresponding real and nonnegative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k \geq 0$.

KLT

- If we define the KLT as

  $T = [u_1, u_2, .., u_k]^T$

- Then, the autocorrelation matrix of Y becomes

$$\mathbf{R_Y} = E[\mathbf{Y}\mathbf{Y}^T] = E[\mathbf{T}\mathbf{X}\mathbf{X}^T\mathbf{T}] = \mathbf{T}\mathbf{R_X}\mathbf{T}^T$$

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

References

- Ze-Nian Li, M. S. Drew, *Fundamentals of Multimedia*, Prentice Hall Inc., 2004. Chapters 7 and 8.