Lossy Compression Algorithms

Lecturer：Jun Xiao
(肖俊）
College of Software and Technology

Outline

- Introduction
- Distortion Measures
- The Rate-Distortion Theory
- Quantization
- Transform Coding
- Wavelet-Based Coding

1. 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.

2. Distortion Measures

2.1 Concept of Distortion

- Distortion Measure
  - A mathematical quantity: specifies how close an approximation to its original
  - It's nature to think of the numerical difference
  - When it comes to image data, difference may not yield the intended result
  - Measures of perceptual distortion

2.2 Numerical Distortion Measures

- Many numerical distortion measures -- the most commonly used distortion measures are presented：MSE、SNR、PSNR
- Mean square error（MSE）： $$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$
  - Average pixel difference
- Signal-to-noise ratio（SNR)： $$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$
  - The size of the error relative to the signal
- Peak-signal-to-noise ratio (PSNR)：

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

  - The size of the error relative to the peak value of the signal

2.2 Numerical Distortion Measures

- Examples of PSNR and corresponding images



original image | polluted by noise | processed by noise filter

PSNR=18.24 | PSNR=39.5

3.The Rate-Distortion Theory

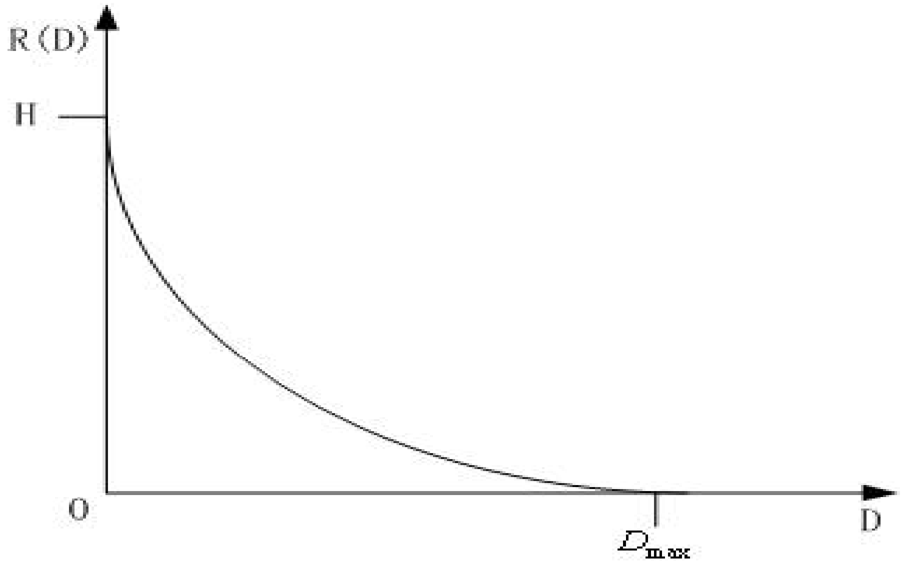
3.1 Concept

- Lossy compression always involves a tradeoff between rate and distortion
  - Rate -- the average number of bits required to represent each source symbol；
  - R(D) note rate-distortion function；

- What is R(D)?
  - R(D) specifies the lowest rate at which the source data can be encoded while keeping the distortion bounded above by D
  - At $D=0$, no loss, so is the entropy of the source data
  - Describe a fundamental limit for the performance of a coding algorithm
  - Can be used to evaluate the performance of different algorithm

3.2 A Typical R-D Function

- A figure of a typical rate-distortion function



- D = 0, the entropy of the source data
- R(D) = 0, nothing coded
- For a given source, it's difficult to find a closed-form analytic description of the rate-distortion function

4. Quantization

4.1 Functions of Quantization

- Quantization: the heart of any lossy scheme
  - Without quantization, almost no losing information
  - Reduce the number of distinct values via quantization
  - Main source of the “loss” in lossy compression
- Each quantizer has its unique partition of the input range and the set of output values.
  - Scalar quantizer
    - Uniform
    - Nonuniform
  - Vector quantizer

4.2 Uniform Scalar Quantization

- Uniform scalar quantizer
  - Partitions the input domain into equally spaced intervals
  - Decision boundaries: the end points of partition intervals
  - Output value: midpoint of the interval
  - Step size: the length of each interval

- Two types of uniform scalar quantizer
  - midrise： with an even number of output levels, one partition interval brackets zero；
  - midtread： odd number of output levels, zero is an output value.

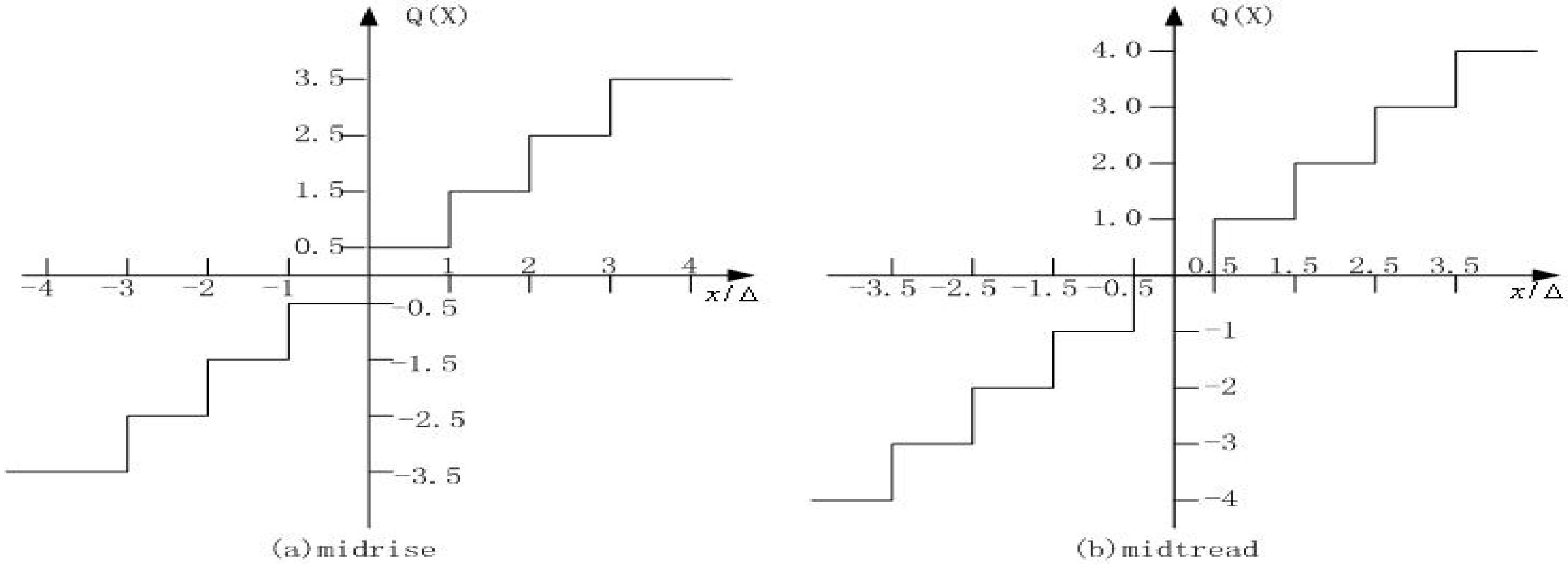
- The goal of a successful uniform quantizer
  - Minimize the distortion for a given source input with a desired number of output values

4.2 Uniform Scalar Quantization

- Given step size $\triangle = 1$,output values for the two type of Quantizers be computed as：

$$Q_{midrise}(x) = \lceil x \rceil - 0.5$$
$$Q_{midread}(x) = \lfloor x + 0.5 \rfloor$$

- Two types quantizers：



4.2 Uniform Scalar Quantization

- Performance of a M level quantizer：
  - Decision Boundaries： B＝ ｛b0,b1,…,bM｝
  - The set of output values： Y={y1,y2,…,ym}
  - The input is uniformly distributed： [-Xmax, Xmax]
  - The rate of quantizer: $R = \log_2^M$ is the number of bits required to code M things；
  - Step size is given by： $\Delta =$ 2Xmax/M
  - Granular distortion: error caused by the quantizer for bounded input
  - Overload distortion: error caused by quantizer for input values larger than Xmax or smaller than -Xmax

4.2 Uniform Scalar Quantization

- Granular distortion for a midrise quantizer
  - Decision boundaries bi:[(i-1)Δ,iΔ],i=1..M/2, covering positive data X (another for native X values)
  - Output values yi： iΔ-Δ/2， i=1..M/2
  - The total distortion: twice the sum over the positive data：

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

  - The error value at X is e(x)=x-Δ/2, variance of errors：

$$\sigma_d^2 = \frac{1}{\Delta}\int_0^{\Delta} (e(x) - \overline{e})^2 dx = \frac{1}{\Delta}\int_0^{\Delta} (x - \frac{\Delta}{2} - 0)^2 dx = \frac{\Delta^2}{12}$$

4.2 Uniform Scalar Quantization

- Signal variance $\sigma_x^2 = (2X_{\max})^2/12$; if the quantizer is n bits, $M = 2^n$
- SQNR can be calculated as：

$$
\begin{aligned}
SQNR &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_d^2}\right) \\
&= 10\log_{10}\left(\frac{(2X_{\max})^2}{12}\cdot\frac{12}{\Delta^2}\right) \\
&= 10\log_{10}\left(\frac{(2X_{\max})^2}{12}\cdot\frac{12}{\left(\frac{2X_{\max}}{M}\right)^2}\right) \\
&= 10\log_{10} M^2 = 20n\cdot\log_{10} 2 \\
&= 6.02n\,(dB)
\end{aligned}
$$

4.3 Nonuniform Scalar Quantization

- If the input source is not uniformly distributed, a uniform quantizer may be inefficient.
- Increasing the number of decision levels within the densely distributed region can lower granular distortion
- Enlarge the region where the source is sparsely distributed can keep the total number of decision levels
- So nonuniform quantizers have nonuniforumly defined decision boundaries.
- Two common approaches for nonuniform quantization：
  - The Lloyd-Max Quantizer
  - The companded quantizer

4.3 Nonuniform Scalar Quantization

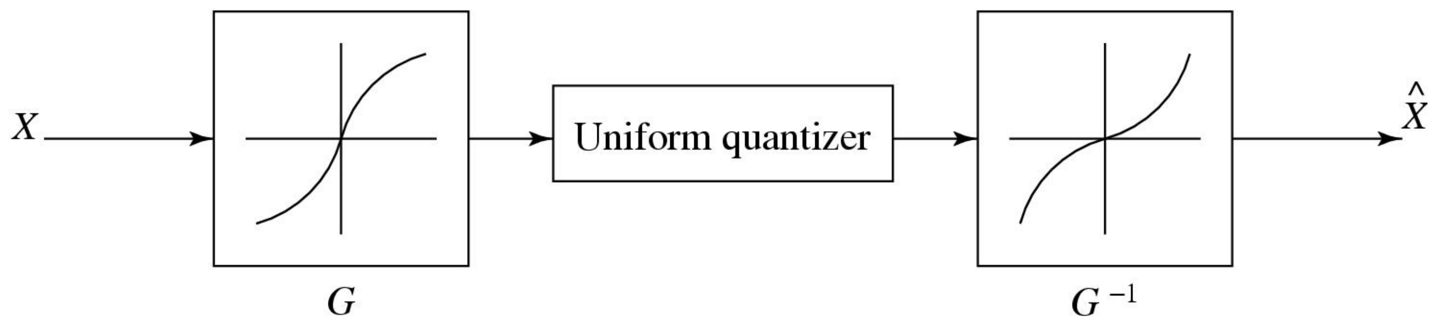


Fig. 8.4: Companded quantization.

- *Companded quantization* is **nonlinear**.
- As shown above, 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.

5. Transform Coding

5.1 Basic Idea

- According principles of information theory
  - Coding vectors is more efficient than coding scalars
  - Need to group consecutive samples from input into vectors
- Let X= $\{x_1,x_2,\ldots,x_k\}$ be vector of samples, there's an amount correlation among neighboring.
- If Y is the result of a linear transform T of the input vector and its components have much less correlation, then Y can be coded more efficiently than X.
  - The transform T itself does not compress any data.
  - The compression comes from the processing and quantization of the components of Y.
- DCT is a widely used transform, it can perform de-correlation of the input signal.

5.1 Basic Idea

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

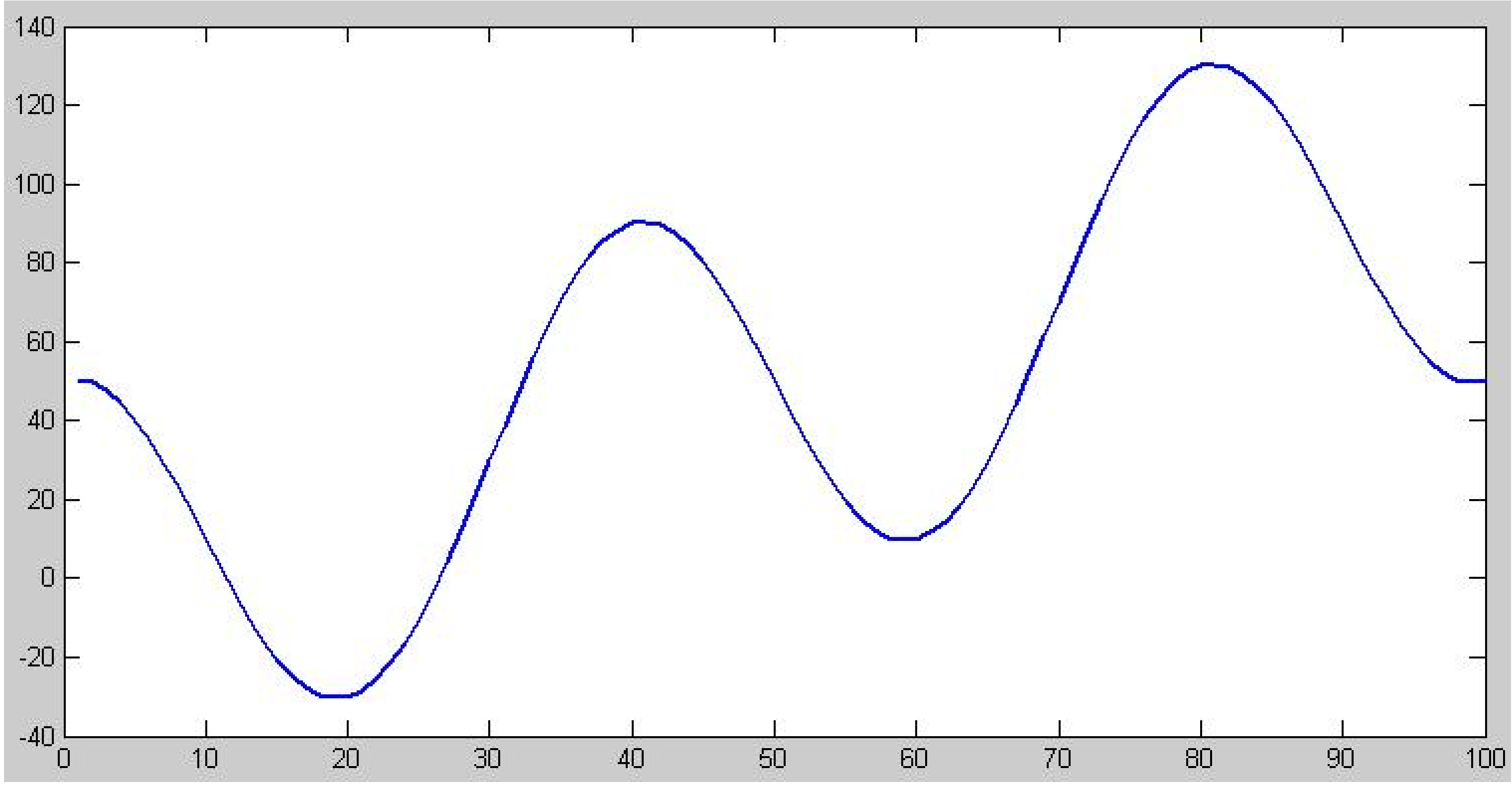
$$C(u)=\begin{cases}\frac{\sqrt{2}}{2},u=0\\ 1,otherewise\end{cases},u=0,1,\ldots,7,i=0,1,\ldots,7$$

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

$$C(u)=\begin{cases}\frac{1}{\sqrt{N}},u=0\\ \sqrt{\frac{2}{N}},otherewise\end{cases},u=0,1,\ldots,N,i=0,1,\ldots,N$$

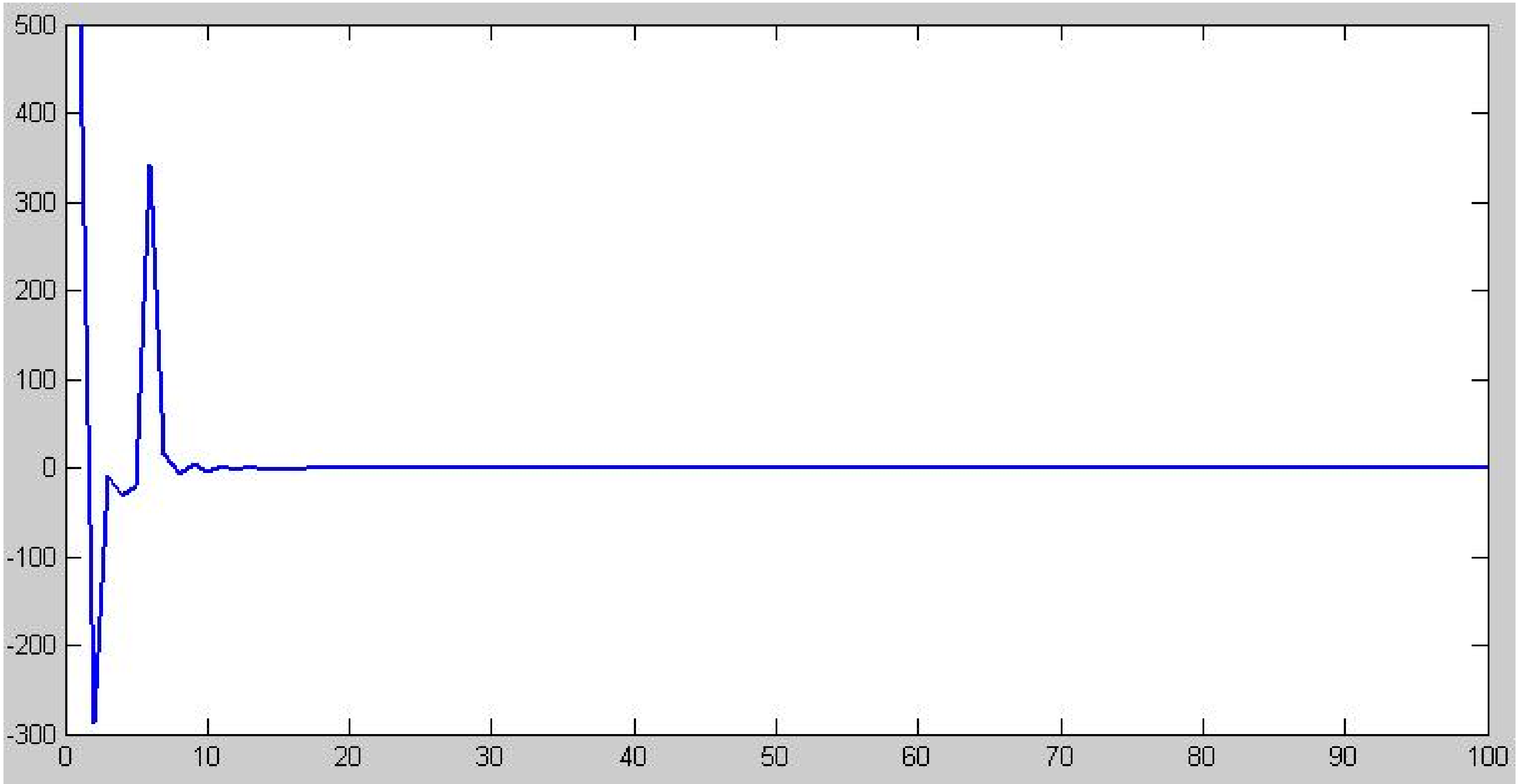
How to compress data by DCT?

- Matlab simulation
  - x = (1:100) + 50*cos((1:100)*2*pi/40);



How to compress data by DCT?

– x_dct = dct(x);



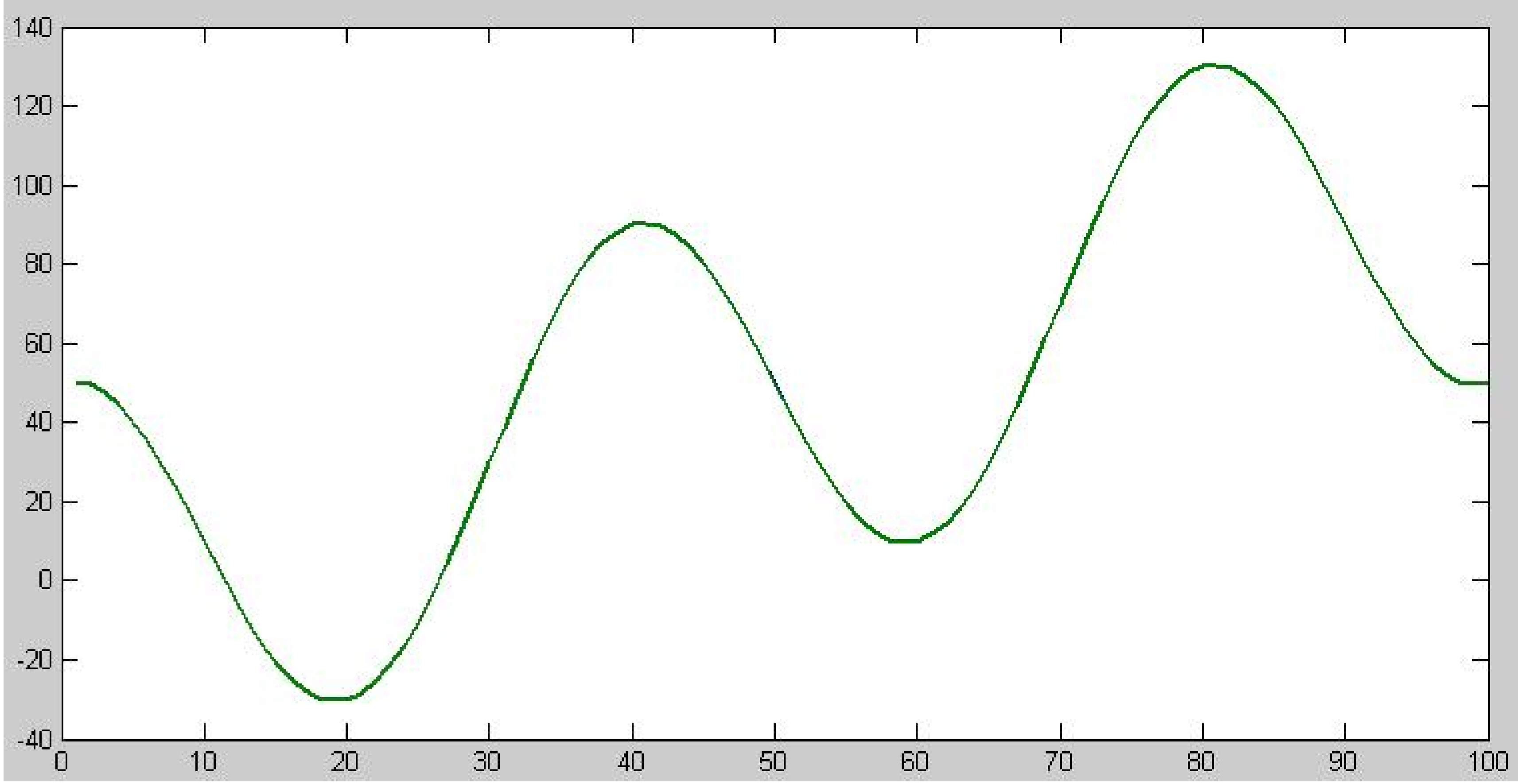
How to compress data by DCT?

- x_dct

```
Columns 1 through 12
500.0000 -286.5678   -8.4166  -31.8304  -19.6289  341.0121   16.0468   -5.8367    4.5208   -3.5262    2.3473   -2.3565
Columns 13 through 24
  1.4766   -1.6838    1.0253   -1.2617    0.7571   -0.9795    0.5831   -0.7817    0.4634   -0.6376    0.3772   -0.5294
Columns 25 through 36
  0.3129   -0.4461    0.2636   -0.3806    0.2249   -0.3281    0.1940   -0.2854    0.1688   -0.2502    0.1481   -0.2208
Columns 37 through 48
  0.1308   -0.1961    0.1162   -0.1750    0.1038   -0.1569    0.0931   -0.1412    0.0838   -0.1275    0.0757   -0.1155
Columns 49 through 60
  0.0686   -0.1049    0.0623   -0.0954    0.0567   -0.0870    0.0517   -0.0794    0.0472   -0.0726    0.0432   -0.0664
Columns 61 through 72
  0.0395   -0.0607    0.0361   -0.0556    0.0330   -0.0508    0.0302   -0.0464    0.0275   -0.0424    0.0251   -0.0386
Columns 73 through 84
  0.0228   -0.0350    0.0207   -0.0317    0.0187   -0.0286    0.0168   -0.0256    0.0150   -0.0228    0.0133   -0.0201
Columns 85 through 96
  0.0116   -0.0175    0.0100   -0.0150    0.0085   -0.0125    0.0070   -0.0102    0.0056   -0.0079    0.0042   -0.0056
Columns 97 through 100
  0.0028   -0.0033    0.0014   -0.0011
```

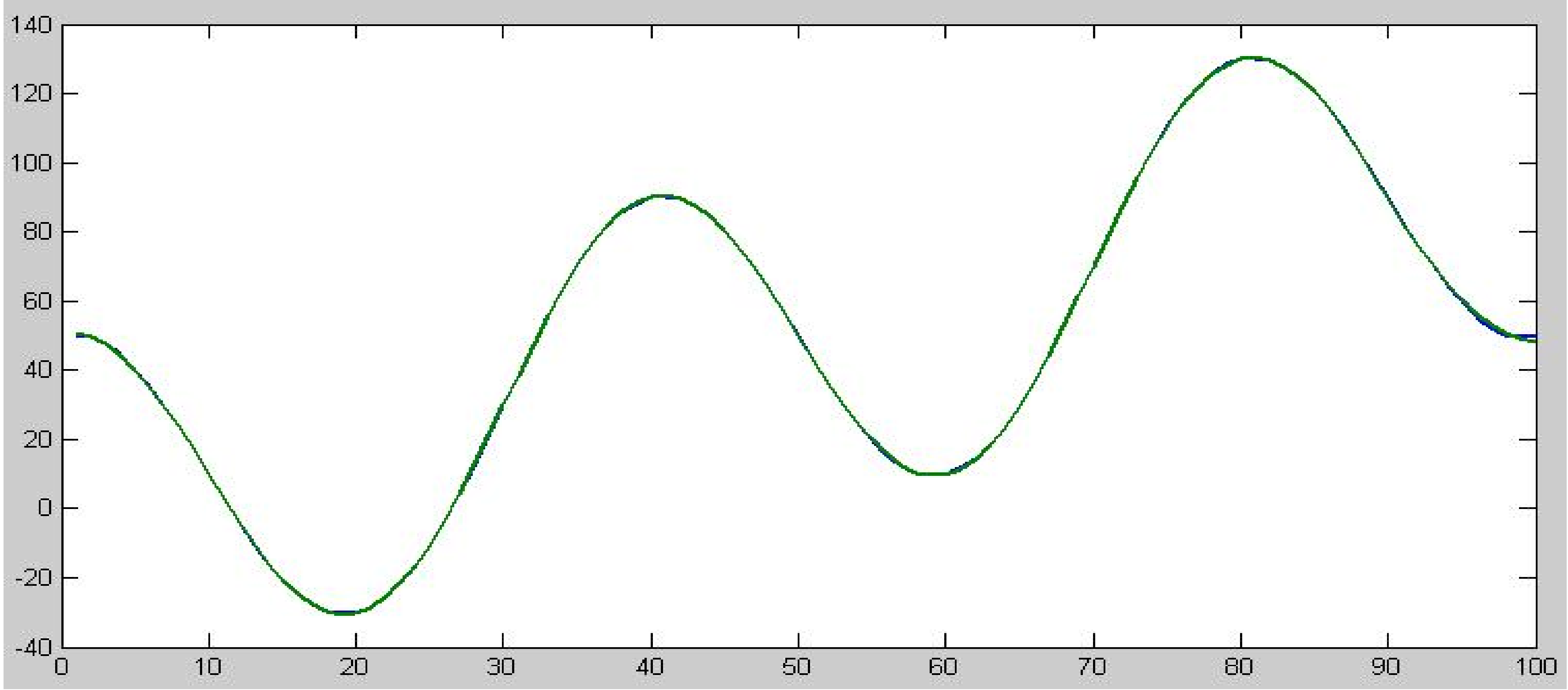
How to compress data by DCT?

- y = idct(x_dct);



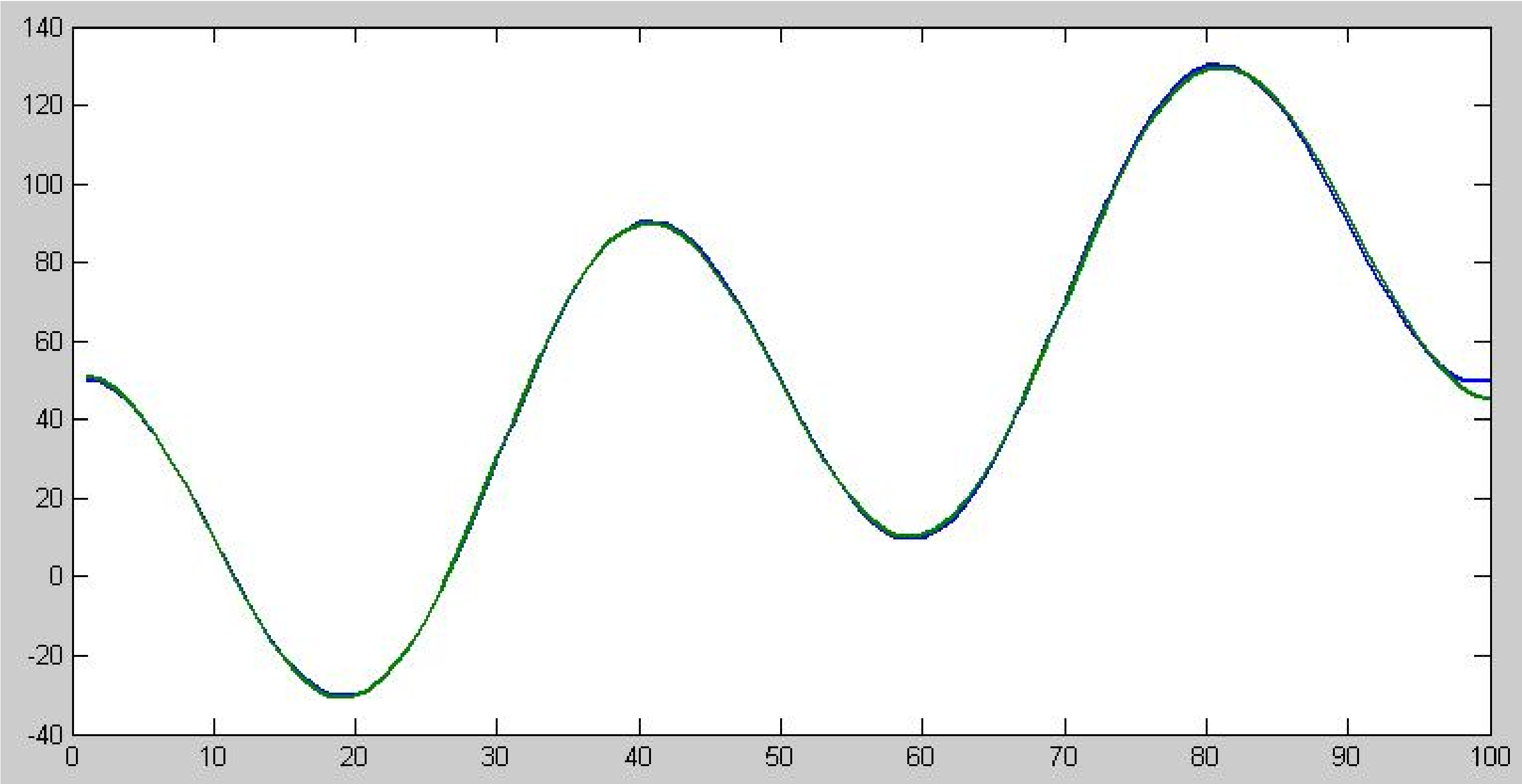
How to compress data by DCT?

– x_dct(16:100) = 0;
– z = idct(x_dct);



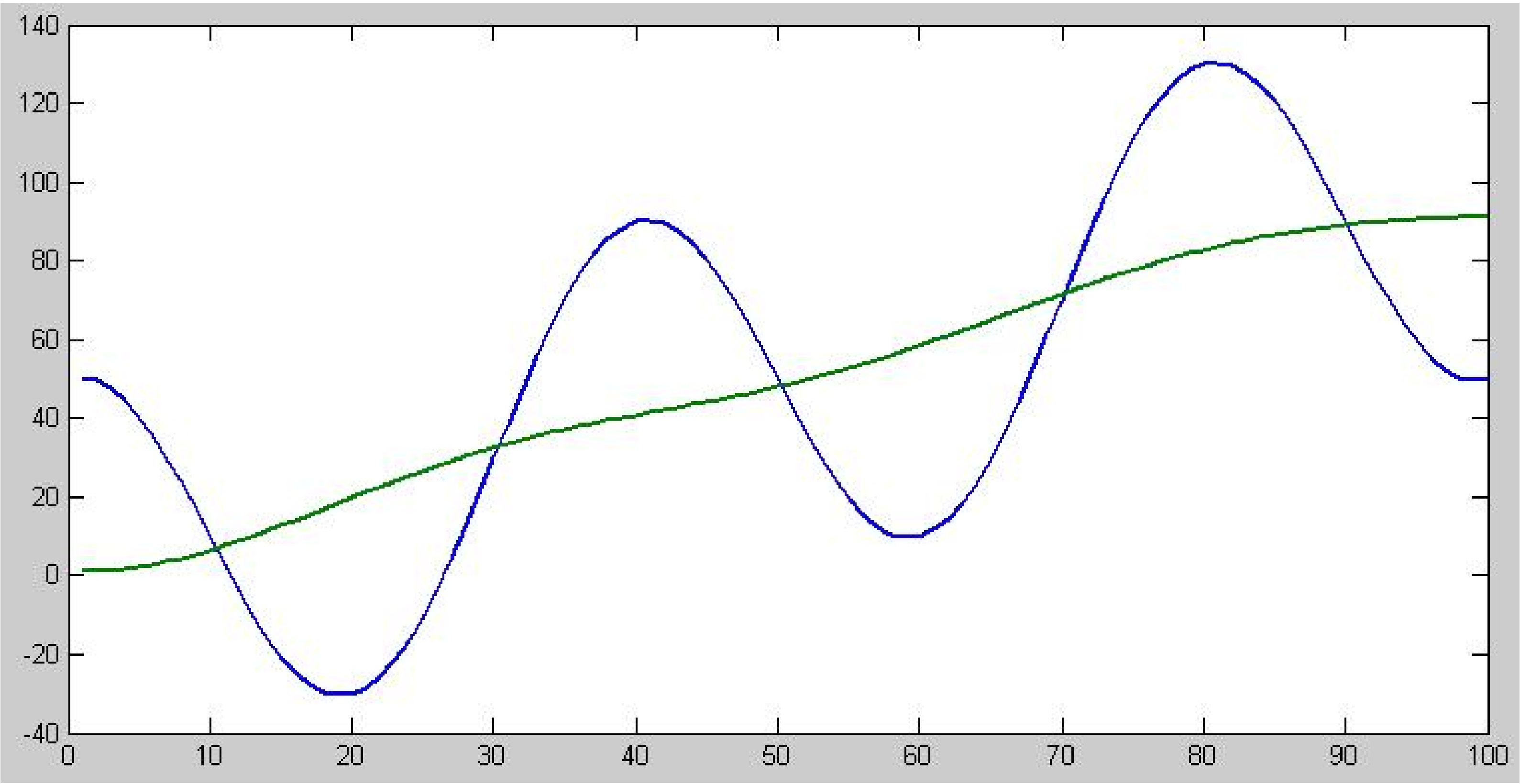
How to compress data by DCT?

– x_dct(8:100) = 0;
– z = idct(x_dct);



How to compress data by DCT?

- x_dct(6:100) = 0;
- z = idct(x_dct);



5.2 Discrete Cosine Transform (DCT)

- 1D Discrete Cosine Transform：

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

- 1D Inverse Discrete Cosine Transform:

$$\tilde{f}_i = \sum_{i=0}^{7} \frac{C(u)}{2} \cos \frac{(2i+1)u\pi}{16} F(u) \qquad C(u) = \begin{cases} \frac{\sqrt{2}}{2} & if\ u = 0 \\ 1 & else \end{cases}$$

- 2D transform can be used to process 2D signals such as digital images

5.2 Discrete Cosine Transform (DCT)

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

$$F(u) = \frac{C(u)}{2} [\cos(\frac{1u\pi}{16}), \cos(\frac{3u\pi}{16}), \cos(\frac{5u\pi}{16}), \cos(\frac{7u\pi}{16}), \cos(\frac{9u\pi}{16}), \cos(\frac{11u\pi}{16}), \cos(\frac{13u\pi}{16}), \cos(\frac{15u\pi}{16})] \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

5.2 Discrete Cosine Transform (DCT)

$$
\begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix}
=
\begin{bmatrix}
\frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) & \frac{1}{2\sqrt{2}}\cos(\frac{0\pi}{16}) \\
\frac{1}{2}\cos(\frac{1\pi}{16}) & \frac{1}{2}\cos(\frac{3\pi}{16}) & \frac{1}{2}\cos(\frac{5\pi}{16}) & \frac{1}{2}\cos(\frac{7\pi}{16}) & \frac{1}{2}\cos(\frac{9\pi}{16}) & \frac{1}{2}\cos(\frac{11\pi}{16}) & \frac{1}{2}\cos(\frac{13\pi}{16}) & \frac{1}{2}\cos(\frac{15\pi}{16}) \\
\cdots \\
\cdots \\
\frac{1}{2}\cos(\frac{7\pi}{16}) & \frac{1}{2}\cos(\frac{21\pi}{16}) & \frac{1}{2}\cos(\frac{35\pi}{16}) & \frac{1}{2}\cos(\frac{49\pi}{16}) & \frac{1}{2}\cos(\frac{63\pi}{16}) & \frac{1}{2}\cos(\frac{77u\pi}{16}) & \frac{1}{2}\cos(\frac{91u\pi}{16}) & \frac{1}{2}\cos(\frac{105\pi}{16})
\end{bmatrix}
\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}
$$

$$
\underbrace{\frac{C(u)}{2}\underbrace{[\cos(\frac{1u\pi}{16}),\cos(\frac{3u\pi}{16}),\cos(\frac{5u\pi}{16}),\cos(\frac{7u\pi}{16}),\cos(\frac{9u\pi}{16}),\cos(\frac{11u\pi}{16}),\cos(\frac{13u\pi}{16}),\cos(\frac{15u\pi}{16})]}_{\text{Basis Function: } B_i}}_{\text{Basis: } b_i}
$$

$$
b_i = \frac{C(i)}{2} B_i
$$

Example

- DCT in 3D Space

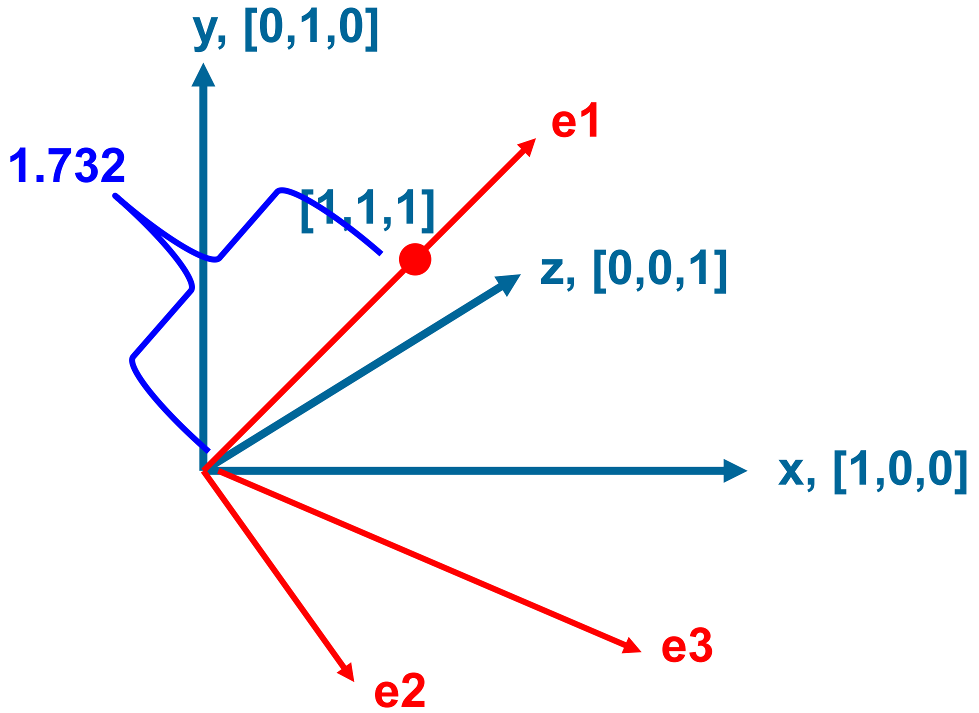
$$y(k) = w(k)\sum_{n=1}^{N} x(n)\cos\frac{\pi(2n-1)(k-1)}{2N}, \qquad k = 1, \ldots, N$$

P = [1,1,1]
=1*[1,0,0]+1*[0,1,0]+1*[0,0,1]

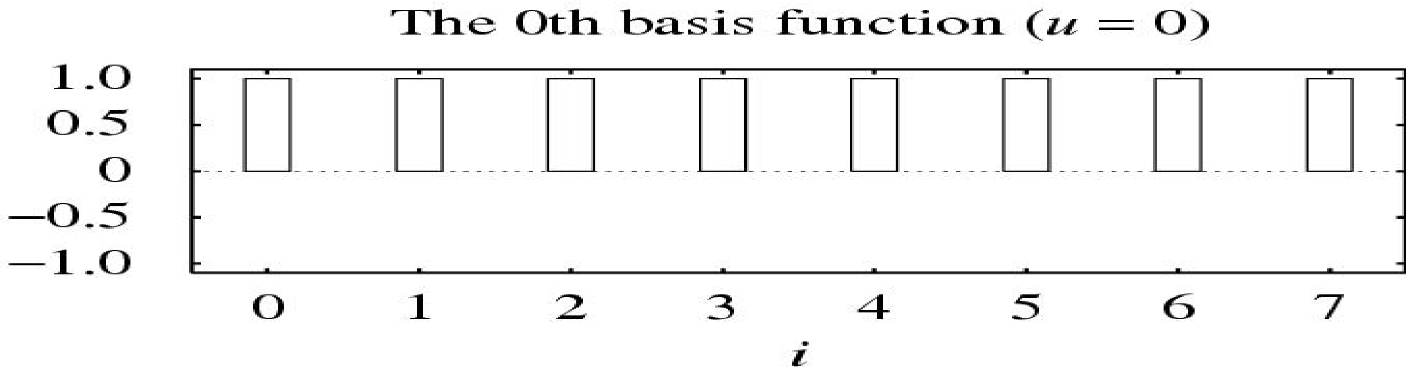
Dct(P)=[1.732,0,0]

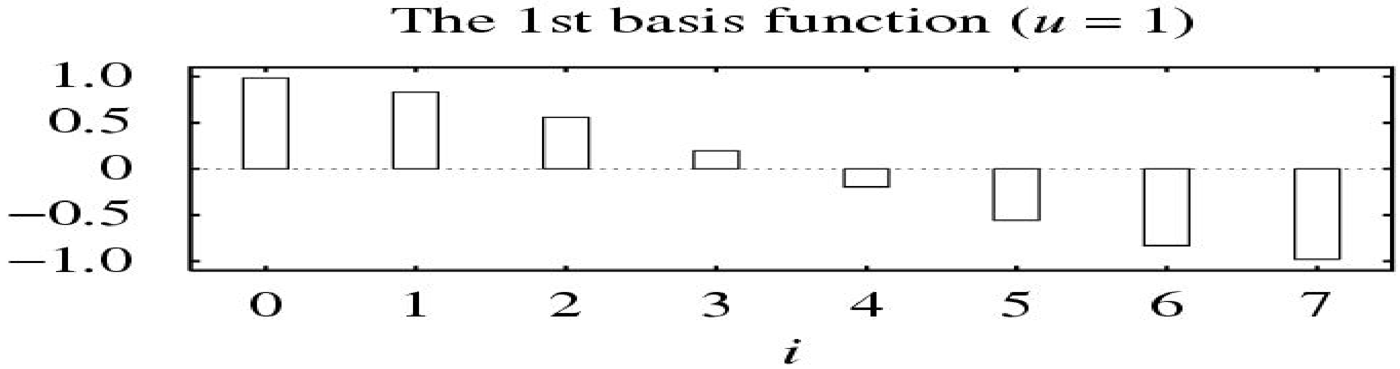
e1 = [0.5774 0.5774 0.5774]
e2 = [0.7071 0.0000 -0.7071]
e3 = [0.4082 -0.8165 0.4082]

P = 1.732*e1+0*e2+0*e3

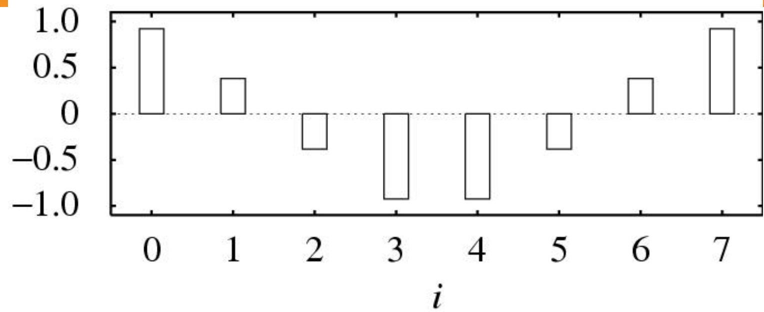


5.2 Discrete Cosine Transform (DCT)

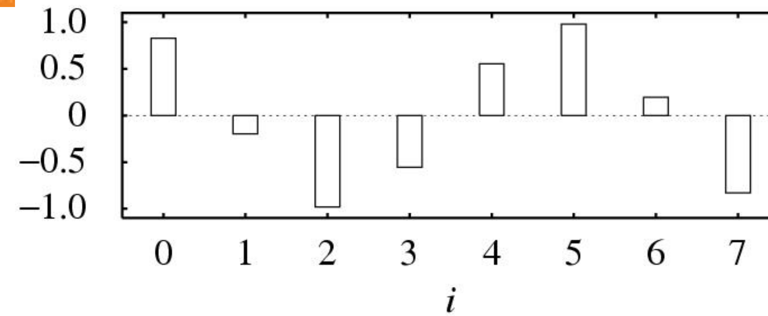




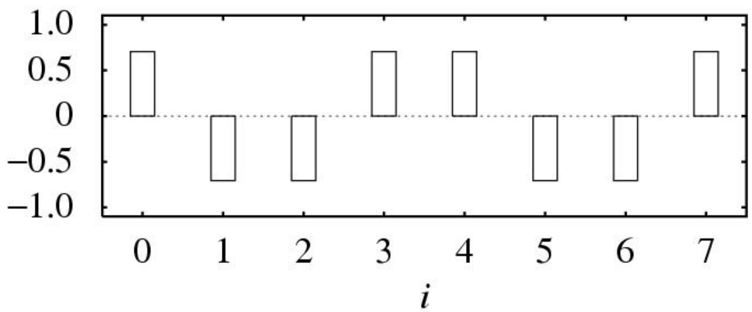
The 2nd basis function ($u = 2$)



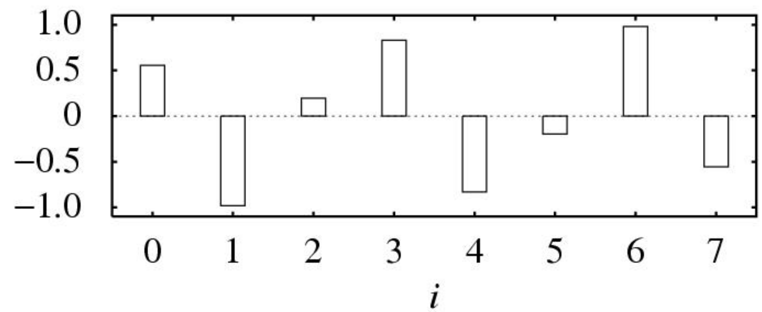
The 3rd basis function ($u = 3$)



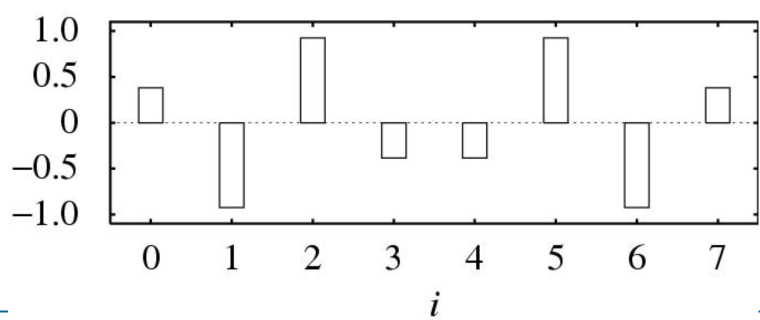
The 4th basis function ($u = 4$)



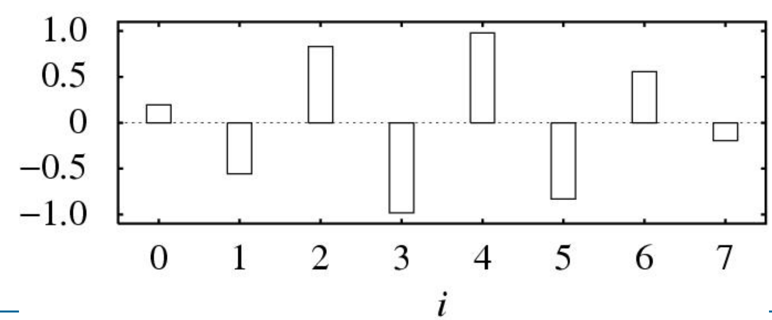
The 5th basis function ($u = 5$)



The 6th basis function ($u = 6$)



The 7th basis function ($u = 7$)



5.2 Discrete Cosine Transform (DCT)

- Cosine basis functions are orthogonal

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

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

Mathematics Meaning: Transform a vector from one linear space to another linear space

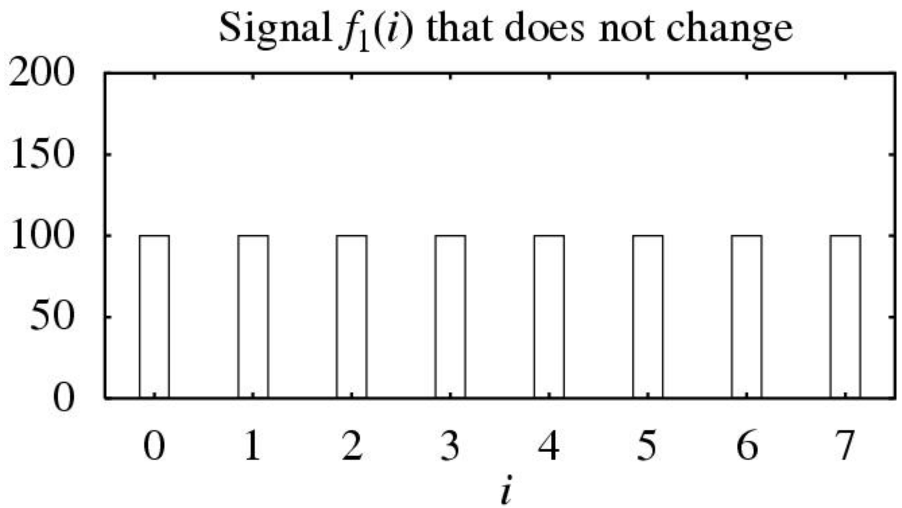
5.2 Discrete Cosine Transform (DCT)

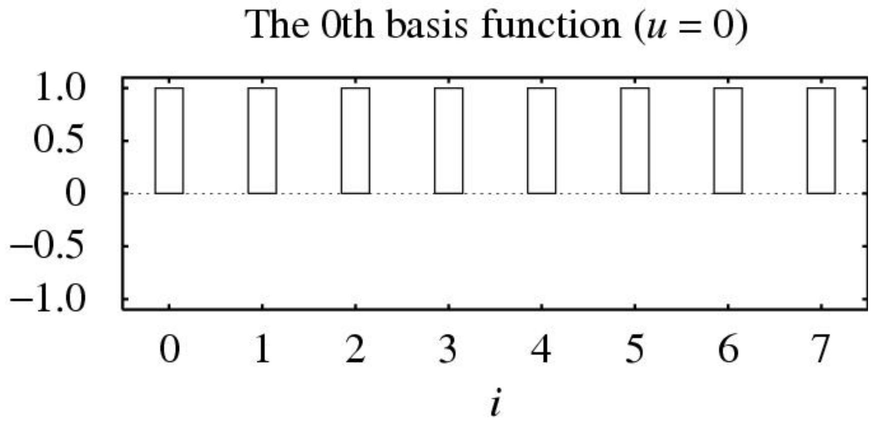
- Example (1)： $f_1(i)$=100, a signal with magnitude of 100

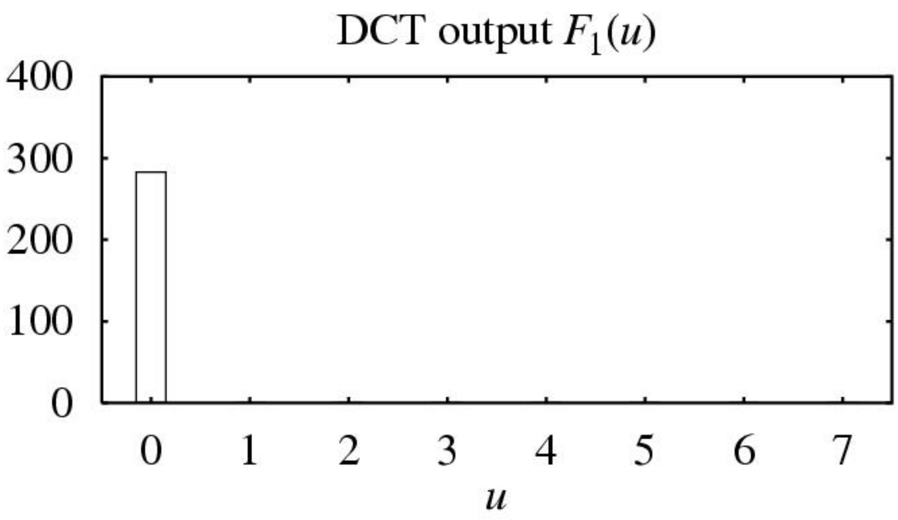
  { $$F(u)=\frac{C(u)}{2}\sum_{i=0}^{7}\cos\frac{(2i+1)u\pi}{16}f(i)$$ }

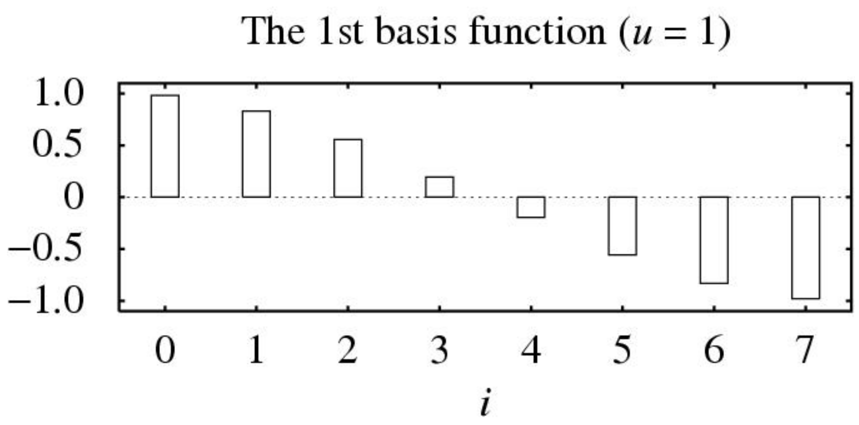
  - $F_1(0)=C(0)/2^*$ （$1\cdot100+ 1\cdot100 + 1\cdot100 + 1\cdot100 + 1\cdot100 + 1\cdot100 + 1\cdot100 + 1\cdot100$ ） noticed that $C(0)=\frac{\sqrt{2}}{2}$
  - $=C(0)\cdot400\approx283$
  - $F_1(1)=$ $$\frac{1}{2}(\cos\frac{\pi}{16}\cdot100+\cos\frac{3\pi}{16}\cdot100+\cos\frac{5\pi}{16}\cdot100+\cos\frac{7\pi}{16}\cdot100+\cos\frac{9\pi}{16}\cdot100+\cos\frac{11\pi}{16}\cdot100+\cos\frac{13\pi}{16}\cdot100+\cos\frac{15\pi}{16}\cdot100$$ $$=0$$
  - $F_1(2)= F_1(3)= F_1(4)= F_1(5)= F_1(6)= F_1(7)=0$

5.2 Discrete Cosine Transform (DCT)







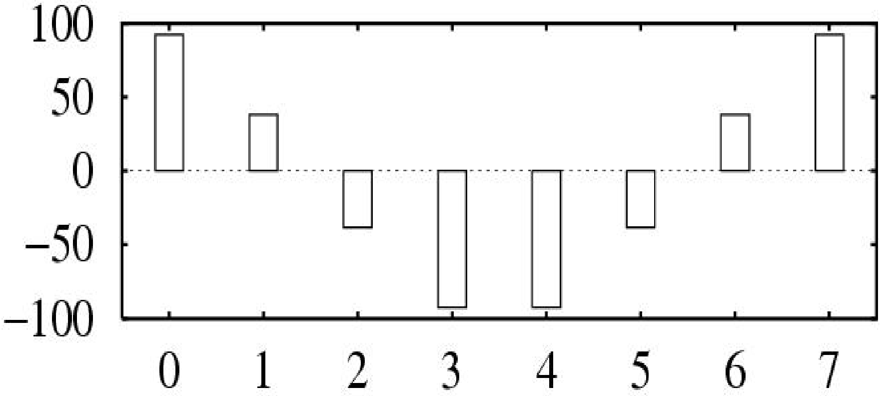


5.2 Discrete Cosine Transform (DCT)

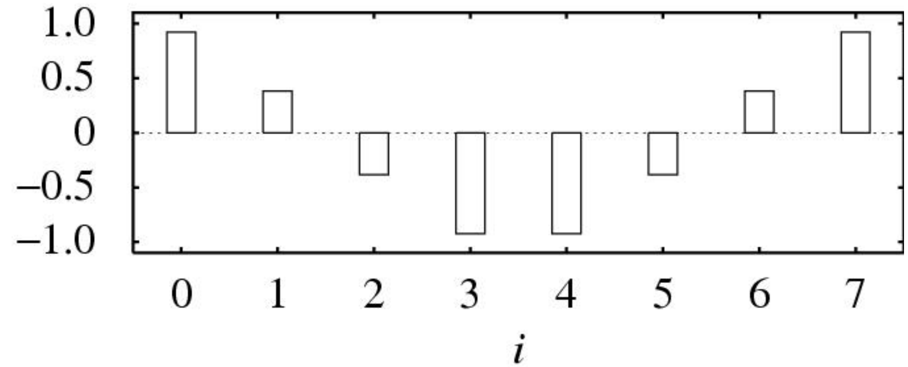
- Example 2: a signal $f_2(i)$, has the same frequency and phase as the second cosine basis function, amplitude is100
  - $F_2(0)=$ $$\frac{\sqrt{2}}{2\cdot 2}\cdot 1\cdot(100\cos\frac{\pi}{8}+100\cos\frac{3\pi}{8}+100\cos\frac{5\pi}{8}+100\cos\frac{7\pi}{8}$$ $$+100\cos\frac{9\pi}{8}+100\cos\frac{11\pi}{8}+100\cos\frac{13\pi}{8}+100\cos\frac{15\pi}{8})$$ $$=0$$
  - $F_2\ (2)=$ $$\frac{1}{2}\cdot(\cos\frac{\pi}{8}\cdot\cos\frac{\pi}{8}+\cos\frac{3\pi}{8}\cdot\cos\frac{3\pi}{8}+\cos\frac{5\pi}{8}\cdot\cos\frac{5\pi}{8}$$ $$+\cos\frac{7\pi}{8}\cdot\cos\frac{7\pi}{8}+\cos\frac{9\pi}{8}\cdot\cos\frac{9\pi}{8}+\cos\frac{11\pi}{8}\cdot\cos\frac{11\pi}{8}$$ $$+\cos\frac{13\pi}{8}\cdot\cos\frac{13\pi}{8}+\cos\frac{15\pi}{8}\cdot\cos\frac{15\pi}{8})\cdot 100$$ $$=200$$
- We can get other values by similar way
  - $F2(1)=F2(3)=F2(4)=\ldots=F2\ (7)=0$

5.2 Discrete Cosine Transform (DCT)

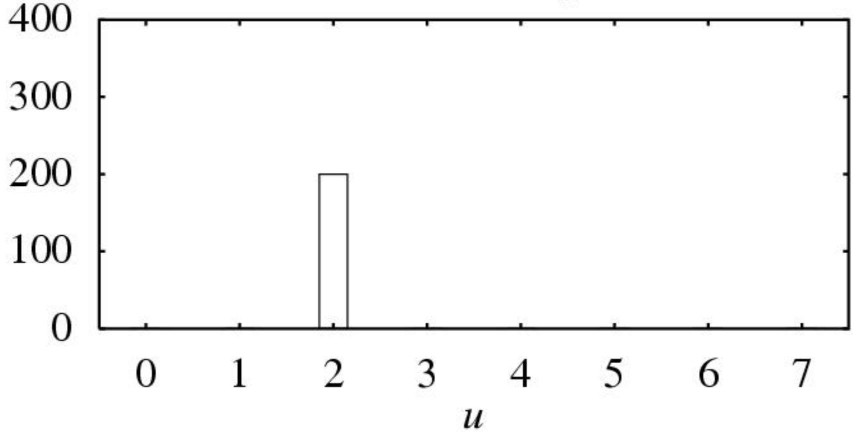
A changing signal $f_2(i)$
that has an AC component



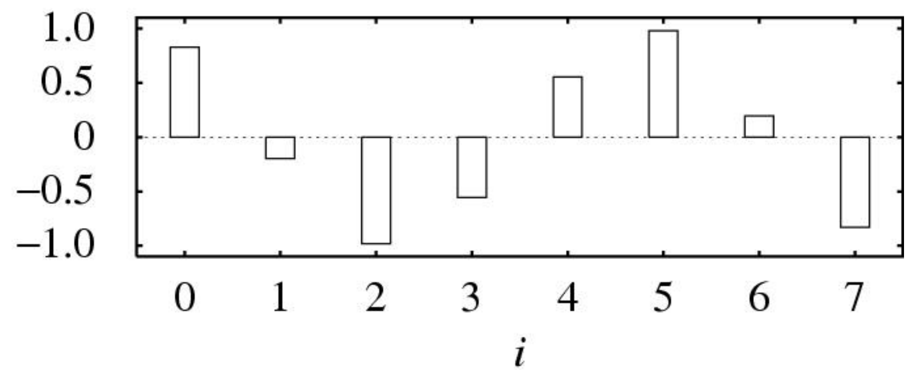
The 2nd basis function ($u = 2$)



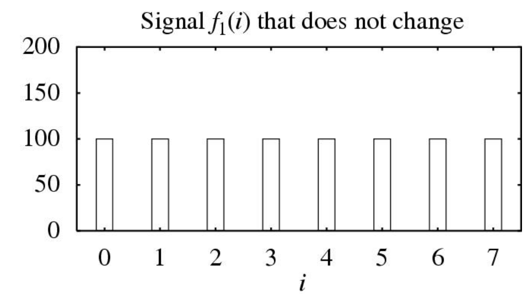
DCT output $F_2(u)$

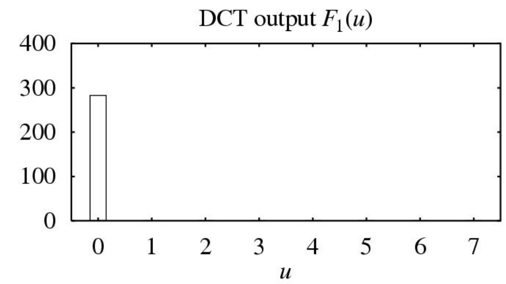


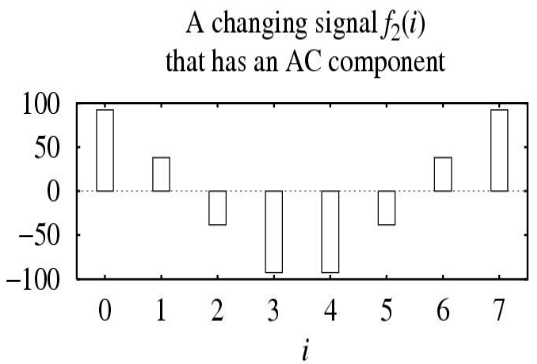
The 3rd basis function ($u = 3$)

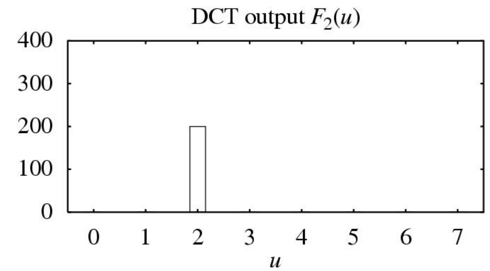


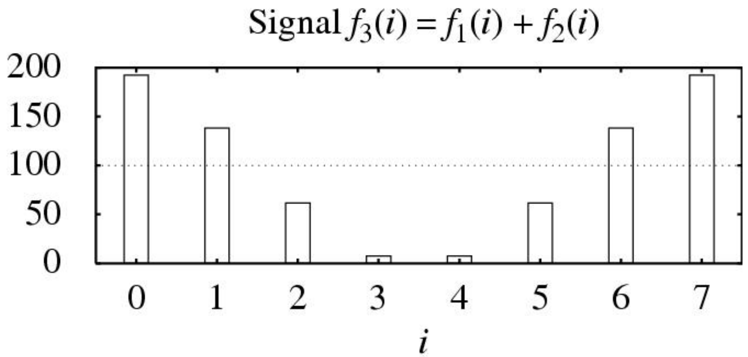
5.2 Discrete Cosine Transform (DCT)

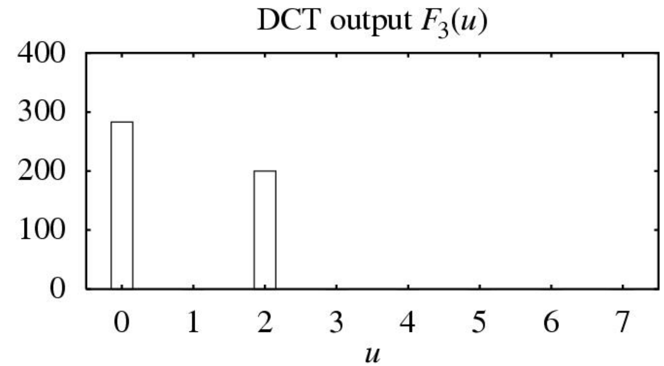












5.2 Discrete Cosine Transform (DCT)

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

$$T(\alpha p + \beta q) = \alpha T(p) + \beta T(q) \tag{8.21}$$

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.

5.2 Discrete Cosine Transform (DCT)

$$T([f(0), f(1), \ldots, f(n)]) == [F(0), F(1), \ldots, F(n)]$$

$$= [F(0), 0, \ldots, 0] + [0, F(1), \ldots, 0] + \ldots + [0, 0, \ldots, F(n)]$$

$$= T(\alpha_0 b_0) + T(\alpha_1 b_1) + \ldots + T(\alpha_n b_n)$$

$$= T(\alpha_0 b_0 + \alpha_1 b_1 + \ldots + \alpha_n b_n)$$

$$\Rightarrow [f(0), f(1), \ldots, f(n)] \equiv \alpha_0 b_0 + \alpha_1 b_1 + \ldots + \alpha_n b_n = \sum_{i=0}^{n} a_i b_i$$

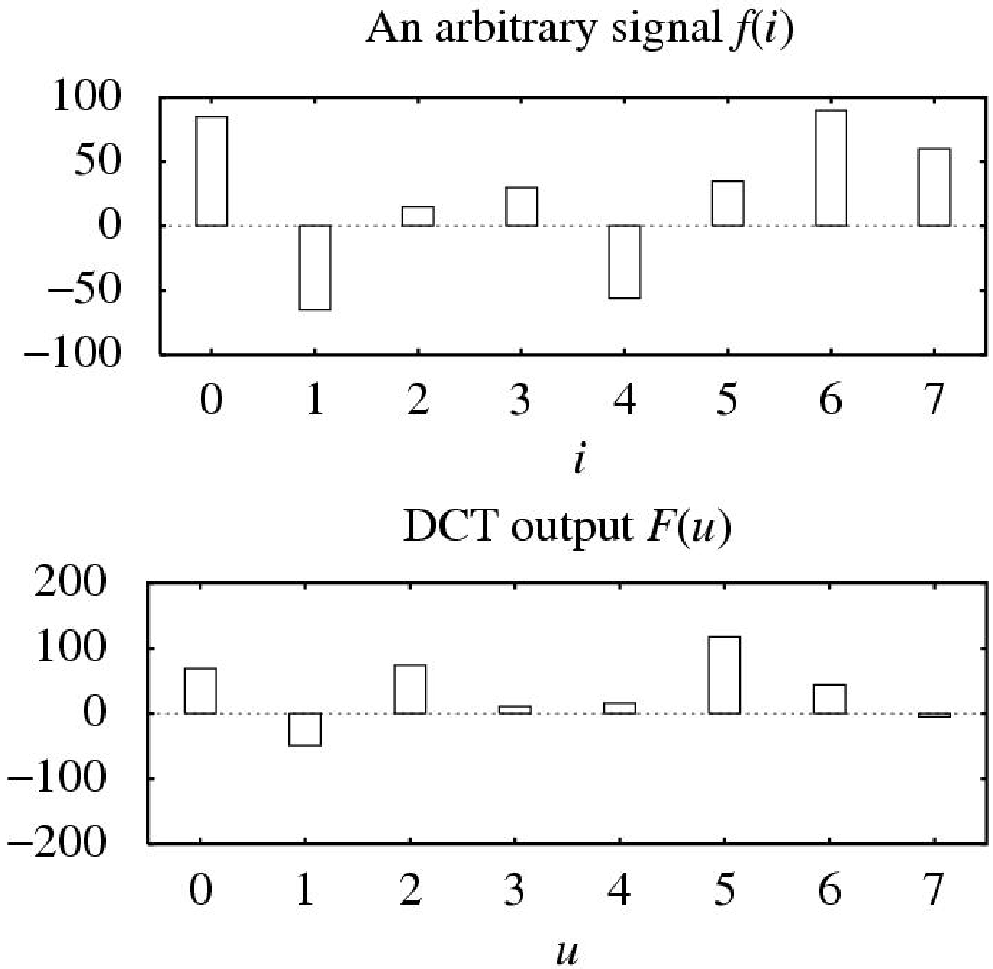
$$=> F(i) = a_i b_i b_i^T = a_i$$

$$=> [f(0), f(1), \ldots, f(n)] = \sum_{i=0}^{n} F(i) b_i$$

Physics Meaning: Approximate the original signal by a linear combination of the Basis Signal.

5.2 Discrete Cosine Transform (DCT)

85 -65 15 30 -56 35 90 60



- One-Dimensional IDCT
  - If F(u) contains (u=0…7):69 -49 74 11 16 117 44 -5
  - IDCT can be implemented by eight iterations：

$$Iteration\,0: \tilde{f}_i = \frac{C(0)}{2}\cos 0 \cdot F(0) \approx 24.3$$

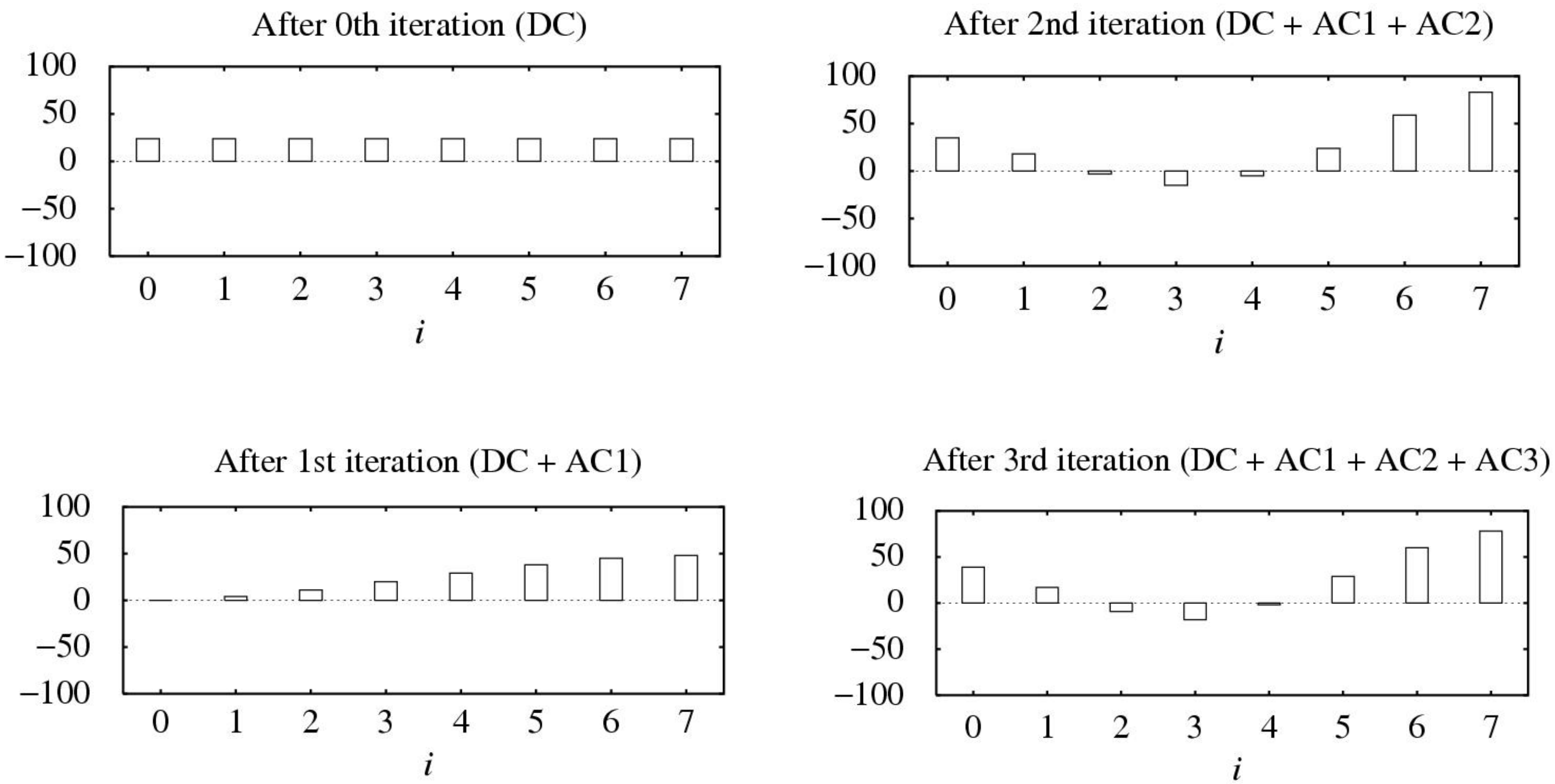
$$Iteration\,1: \tilde{f}_i = \frac{C(0)}{2}\cos 0 \cdot F(0) + \frac{C(1)}{2}\cos\frac{(2i+1)\pi}{16} \cdot F(1)$$

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

$$Iteration\,1: \tilde{f}_i = \frac{C(0)}{2}\cos 0 \cdot F(0) + \frac{C(1)}{2}\cos\frac{(2i+1)\pi}{16} \cdot F(1) + \frac{C(2)}{2}\cos\frac{(2i+1)\pi}{8} \cdot F(2)$$

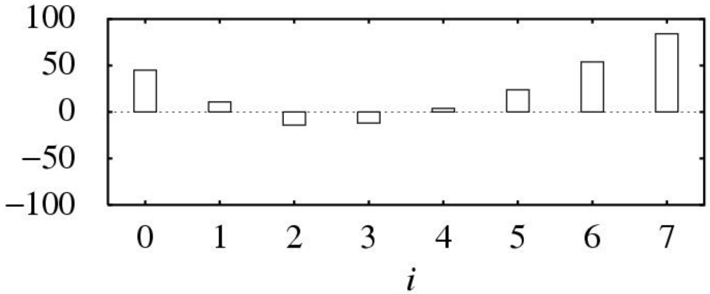
$$\approx 24.3 - 24.5 \cdot \cos\frac{(2i+1)\pi}{16} + 37 \cdot \cos\frac{(2i+1)\pi}{8}$$

5.2 Discrete Cosine Transform (DCT)

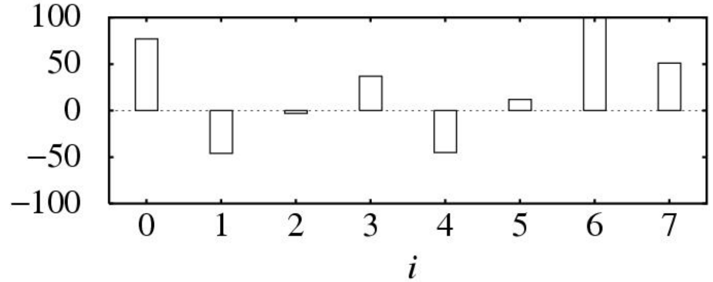


5.2 Discrete Cosine Transform (DCT)

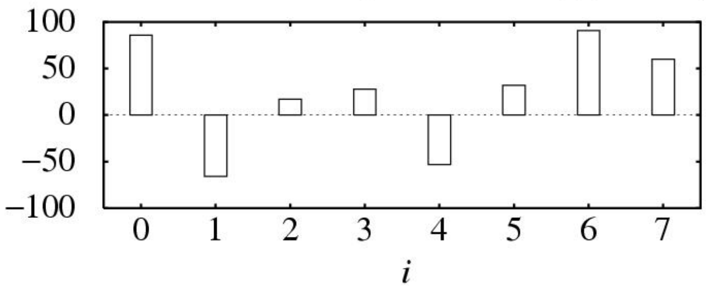
After 4th iteration (DC + AC1 + . . . + AC4)



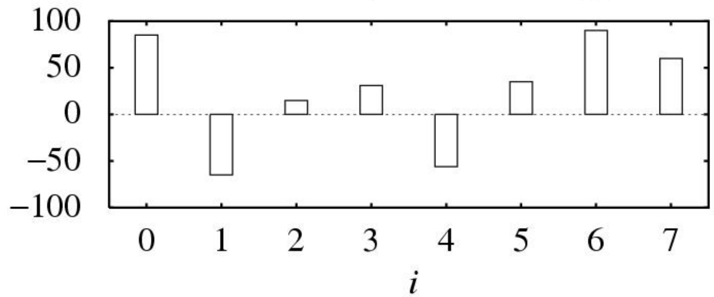
After 5th iteration (DC + AC1 + . . . + AC5)



After 6th iteration (DC + AC1 + . . . + AC6)



After 7th iteration (DC + AC1 + . . . + AC7)



5.2 Discrete Cosine Transform (DCT)

- DCT related concepts
  - Direct current (DC) and alternating current (AC)
    - Represent constant and variable magnitude respectively；
  - Cosine Transform
    - The process used to determine the amplitude of the AC and DC components of the signal.
  - Discrete Cosine Transform： integer indices
    - U＝0, we get the DC coefficient；
    - U=1,2, …,7， we get the first up to seventh AC coefficients.
  - Invert Discrete Cosine transform： using DC, AC and cosine functions to reconstruct the signal
  - DCT and IDCT adopt the same set of cosine functions which are know as basis functions

5.2 Discrete Cosine Transform (DCT)

- DCT enable to process or analyze the signal in frequency domain
- Suppose f(i) represents a signal changes with time i
  - 1D DCT transforms f(i) in time domain to F(U) in frequency domain.
  - F(u) are known as frequency response, form the frequency spectrum of f(i)

5.2 Discrete Cosine Transform (DCT)

Properties of DCT transform

- DCT produces the frequency spectrum $F(u)$ of signal $f(i)$
  - The 0th DCT coefficient $F(0)$ is the DC component of the signal $f(i)$;
  - The other seven DCT coefficients reflect the various changing components of signal $f(i)$ at different frequencies;
  - If DC is a negative value, this means that the average of $f(i)$ is less than zero ;
  - if AC is a negative value, this means that $f(i)$ and some basis function have the same frequency but one of them happens to be half a cycle behind.

5.2 Discrete Cosine Transform (DCT)

DCT（2D）Definition：

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

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

5.2 Discrete Cosine Transform (DCT)

- In the JPEG image compression standard
  - An image block is defined to have dimension M=N=8；
  - The definition of 2D DCT and its inverse IDCT are as follows：
- 2D Discrete Cosine Transform(2D DCT)：

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

- 2D Inverse Discrete Cosine Transform(2D IDCT)：

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

5.2 Discrete Cosine Transform (DCT)

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

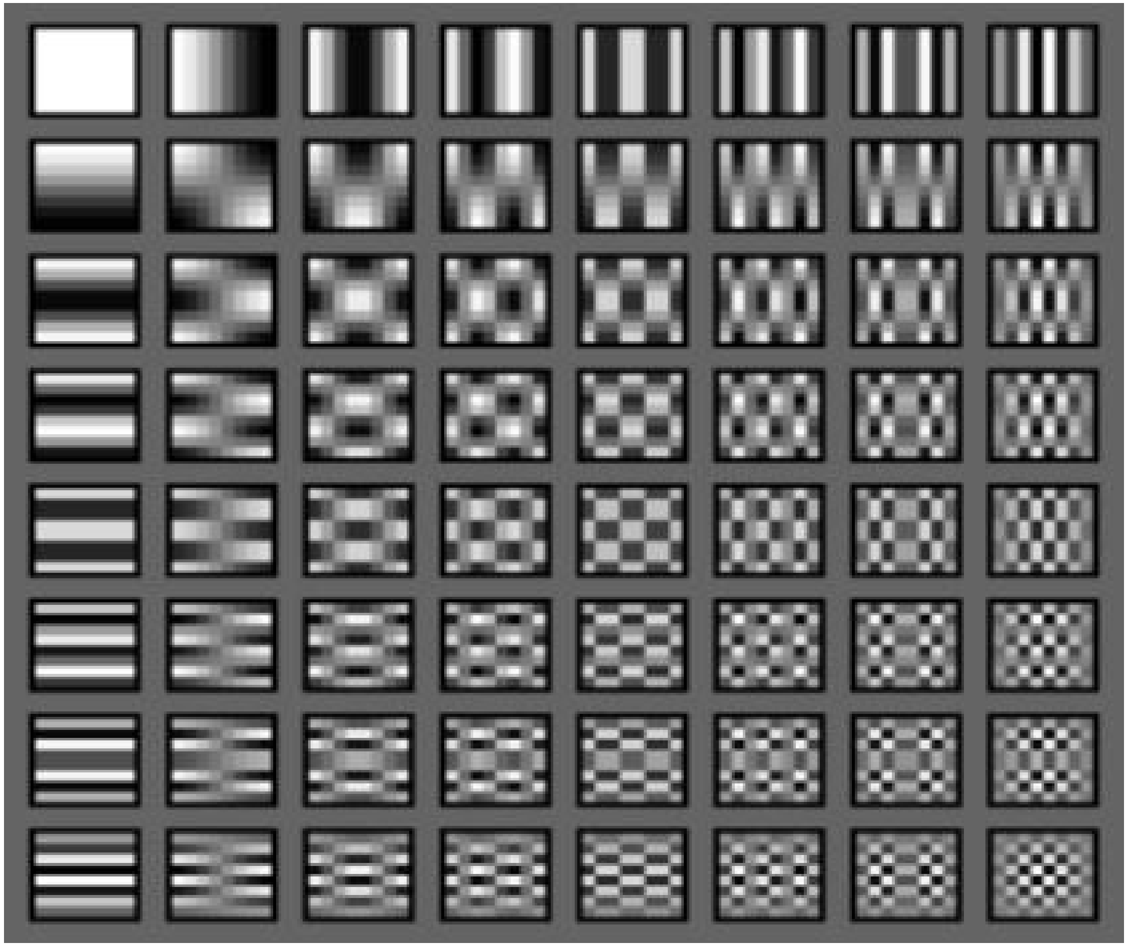
$$G(i,v) = \frac{1}{2} C(v) \sum_{j=0}^{7} \cos \frac{(2j+1)v\pi}{16} f(i,j)$$

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

- It is straightforward to see that this simple change savesmany arithmetic steps. The number of iterations required is reduced from 8 $\times$ 8 to 8+8.

5.2 Discrete Cosine Transform (DCT)

2D Basis Functions



5.3 Comparison of DCT and DFT

- 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:

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

  - 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 {f0, f1, . . . , f7} as:

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

5.3 Comparison of DCT and DFT

- Writing the sine and cosine terms explicitly, we have

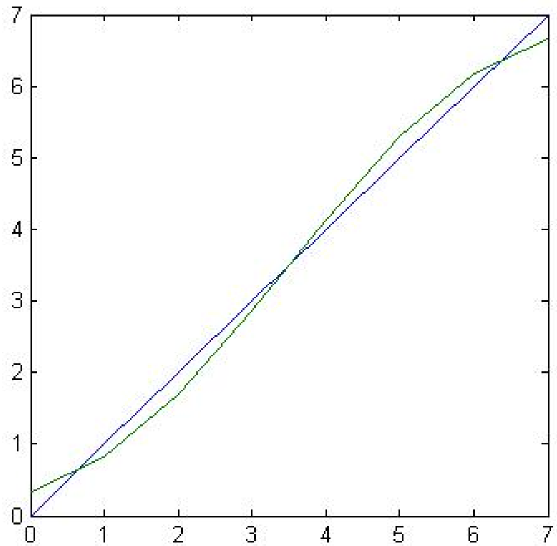
$$F_\omega = \sum_{x=0}^{7} f_x \cos\left(\frac{2\pi\omega x}{8}\right) - i\sum_{x=0}^{7} f_x \sin\left(\frac{2\pi\omega x}{8}\right)$$

- 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.

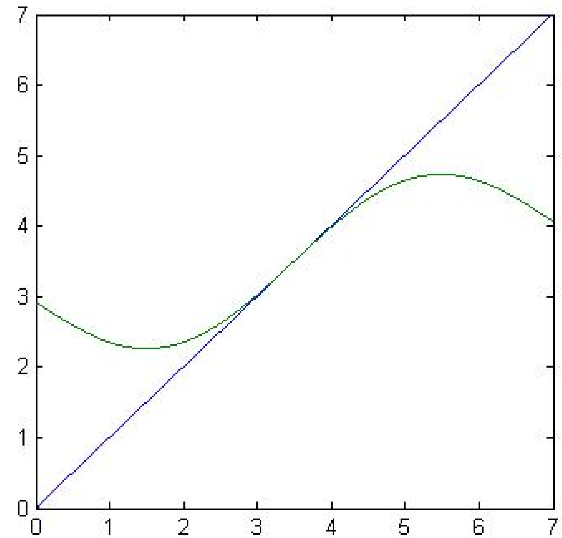
- DCT of 8 input samples corresponds to DFT of the 16 samples made up of original 8 input samples and a symmetric copy of these, as shown in Fig. 8.10.

5.3 Comparison of DCT and DFT

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



(a) three-term DCT approximation

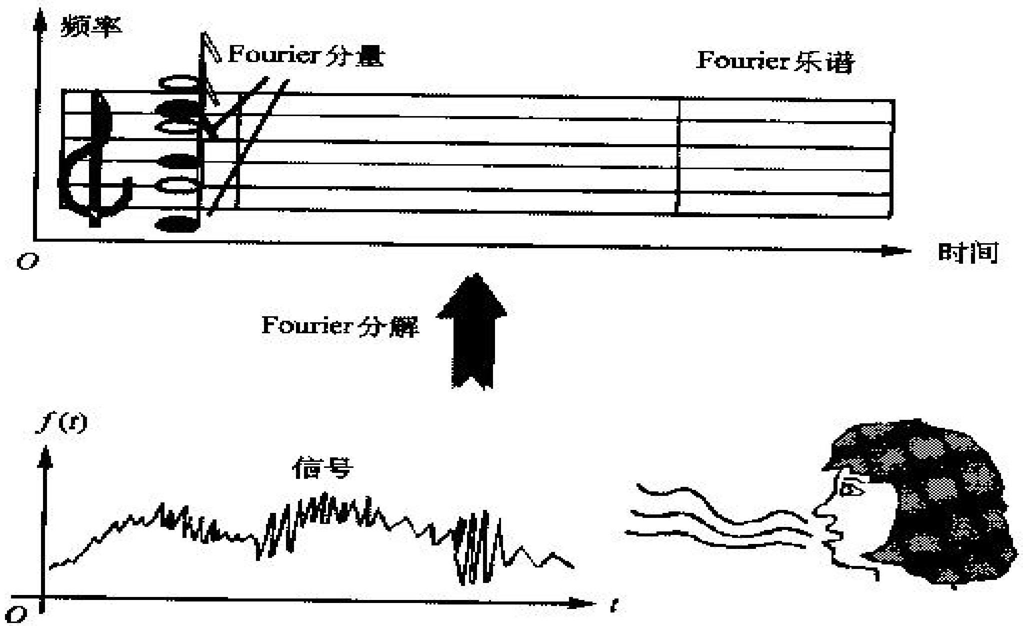


(b) three-term DFT approximation

6. Wavelet-Based Coding

6.1 Introduction

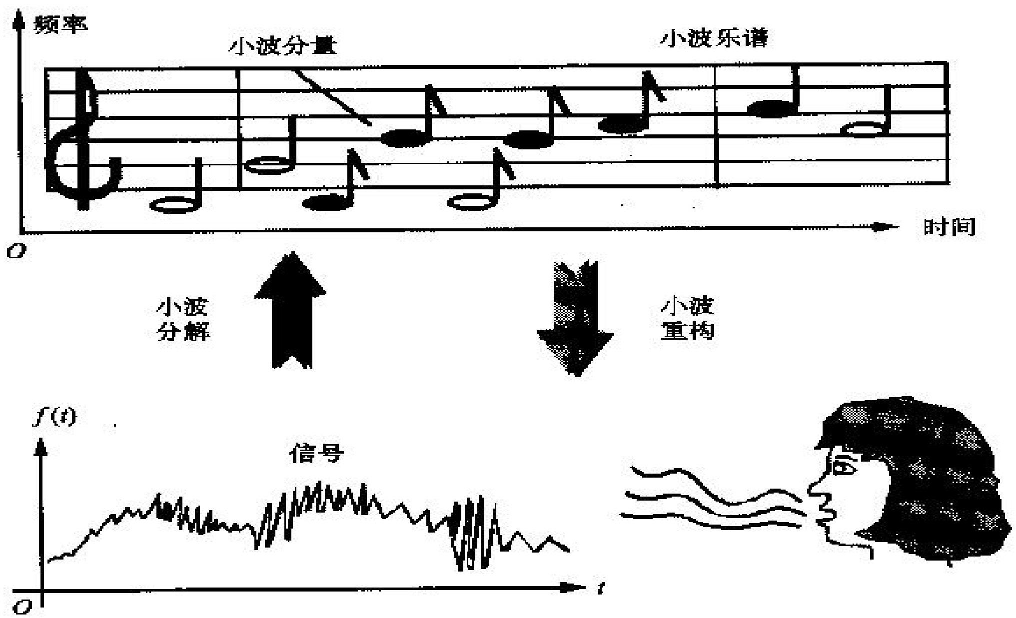
- DFT and DCT can give very fine resolution in the frequency domain, but no temporal resolution.



Fourier Decomposition of A Song Signal

5.1 Introduction

- Wavelet transform seeks to represents a signal with good resolution in both time and frequency.



Wavelet Decomposition of A Song Signal

5.2 Wavelet Transform Example

- Suppose we are given the following input sequence.

  $\{x_{n,i}\} = \{10, 13, 25, 26, 29, 21, 7, 15\}$

- Consider the transform that replaces the original sequence with its pairwise average $x_{n-1,i}$ and difference $d_{n-1,i}$ defined as follows:

$$x_{n-1,i} = \frac{x_{n,2i} + x_{n,2i+1}}{2}$$

$$d_{n-1,i} = \frac{x_{n,2i} - x_{n,2i+1}}{2}$$

- The averages and differences are applied only on consecutive pairs of input sequences whose first element has an even index. Therefore, the number of elements in each set $\{x_{n-1,i}\}$ and $\{d_{n-1,i}\}$ is exactly half of the number of elements in the original sequence.

5.2 Wavelet Transform Example

- Form a new sequence having length equal to that of the original sequence by concatenating the two sequences $\{x_{n-1,i}\}$ and $\{d_{n-1,i}\}$. The resulting sequence is

  $\{x_{n-1,i}, d_{n-1,i}\} = \{11.5, 25.5, 25, 11, -1.5, -0.5, 4, -4\}$

  $\{x_{n,i}\} = \{10, 13, 25, 26, 29, 21, 7, 15\}$

- This sequence has exactly the same number of elements as the input sequence — the transform did not increase the amount of data.

- Since the first half of the above sequence contain averages from the original sequence, we can view it as a coarser approximation to the original signal. The second half of this sequence can be viewed as the details or approximation errors of the first half.

5.3 1D Haar Transform

It is easily verified that the original sequence can be reconstructed from the transformed sequence using the relations

$$x_{n,\,2i} = x_{n-1,\,i} + d_{n-1,\,i}$$
$$x_{n,\,2i+1} = x_{n-1,\,i} - d_{n-1,\,i}$$

This transform is the discrete Haar wavelet transform.

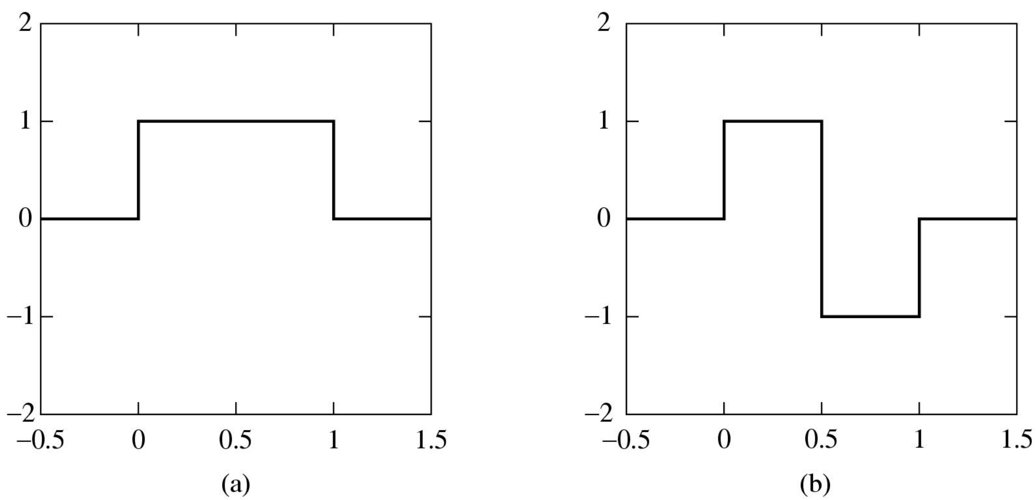
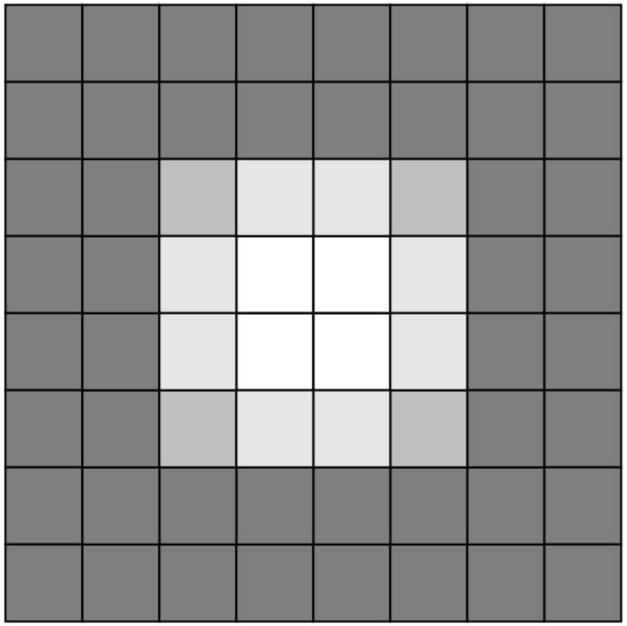


Fig. 8.12: Haar Transform: (a) scaling function, (b) wavelet function.

5.4 2D Haar Wavelet Transform

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	63	127	127	63	0	0
0	0	127	255	255	127	0	0
0	0	127	255	255	127	0	0
0	0	63	127	127	63	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)



(b)

Fig. 8.13: Input image for the 2D Haar Wavelet Transform. (a) The pixel values. (b) Shown as an 8 $\times$ 8 image.

5.4 2D Haar Wavelet Transform

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	95	95	0	0	–32	32	0
0	191	191	0	0	–64	64	0
0	191	191	0	0	–64	64	0
0	95	95	0	0	–32	32	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 8.14: Intermediate output of the 2D Haar Wavelet Transform.

5.4 2D Haar Wavelet Transform

0	0	0	0	0	0	0	0
0	143	143	0	0	–48	48	0
0	143	143	0	0	–48	48	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	–48	–48	0	0	16	–16	0
0	48	48	0	0	–16	16	0
0	0	0	0	0	0	0	0

Fig. 8.15: Output of the first level of the 2D Haar Wavelet Transform.

5.4 2D Haar Wavelet Transform

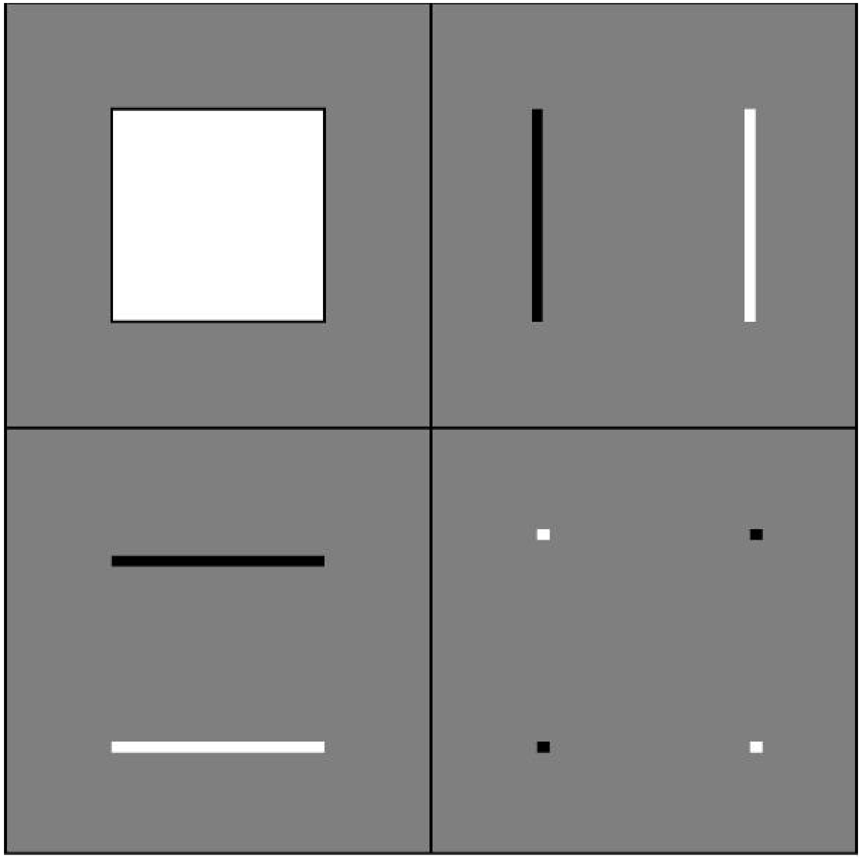


Fig. 8.16: A simple graphical illustration of Wavelet Transform.

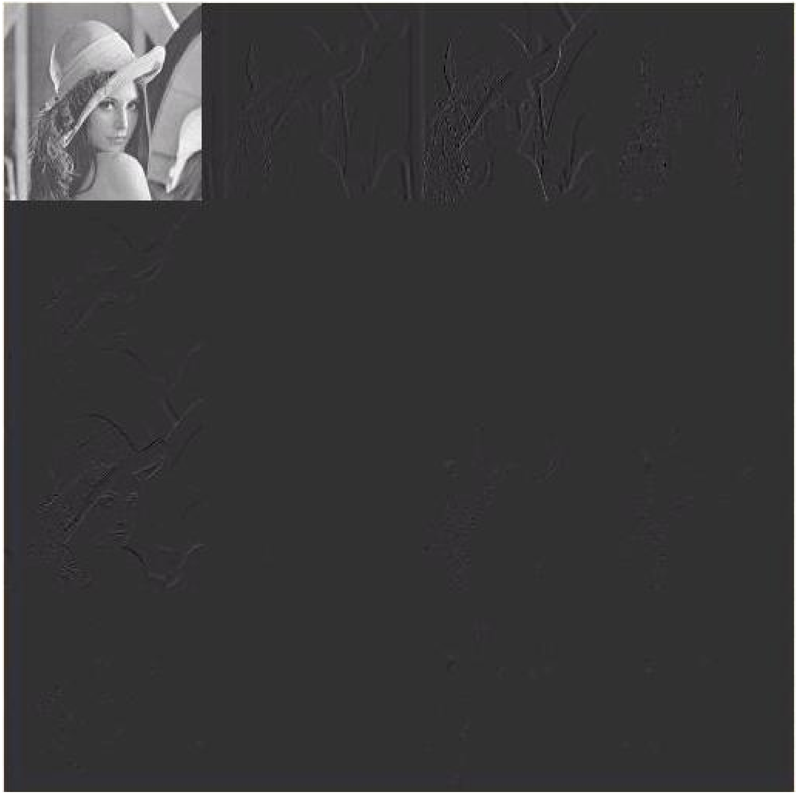
5.4 2D Haar Wavelet Transform

original image

wavelet horizontally transform

5.4 2D Haar Wavelet Transform

wavelet horizontally and vertically transform (one level)



wavelet transformation (2 levels)

The End

Thanks！

Email: junx@cs.zju.edu.cn