

4.1 傅立叶变换基础

傅里叶级数

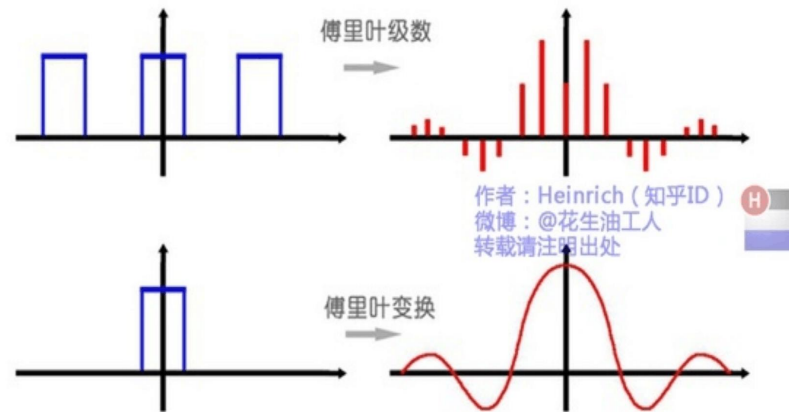
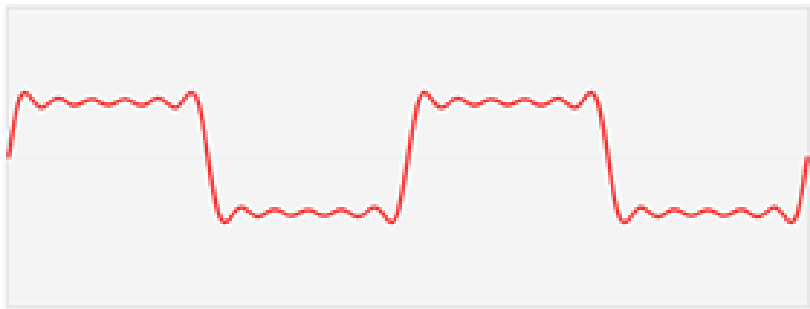
$$\begin{cases} \tilde{x}(t) = \sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t} \\ a_k = \frac{1}{T} \int_T \tilde{x}(t) e^{-jk\omega_0 t} dt \end{cases}$$

Fourier transform:

$$x(t) \xleftrightarrow{F} X(j\omega)$$

$$\begin{cases} X(j\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \\ x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega) e^{j\omega t} d\omega \end{cases}$$

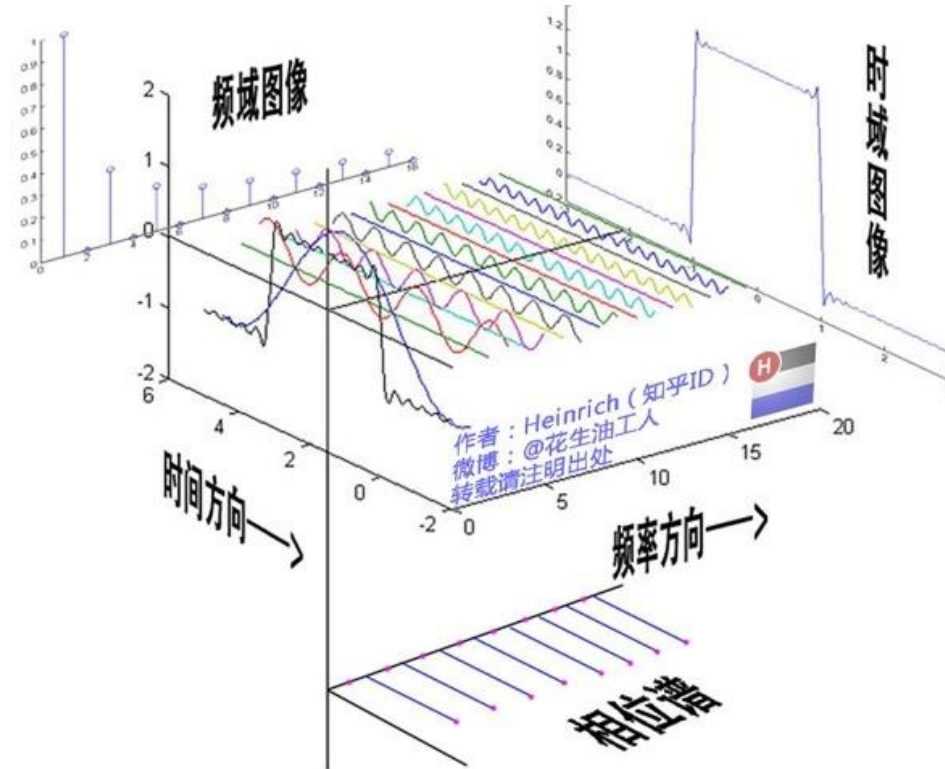
S(x) 是由多个正弦信号叠加构成的波型函数



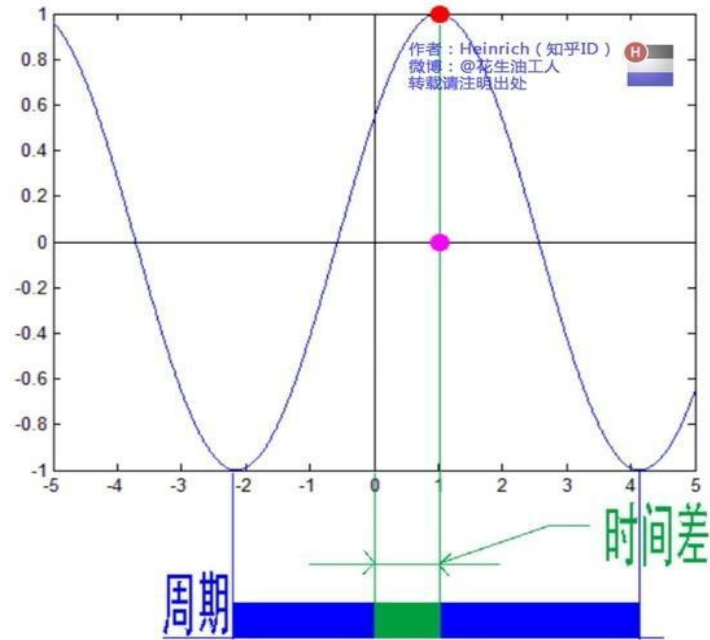
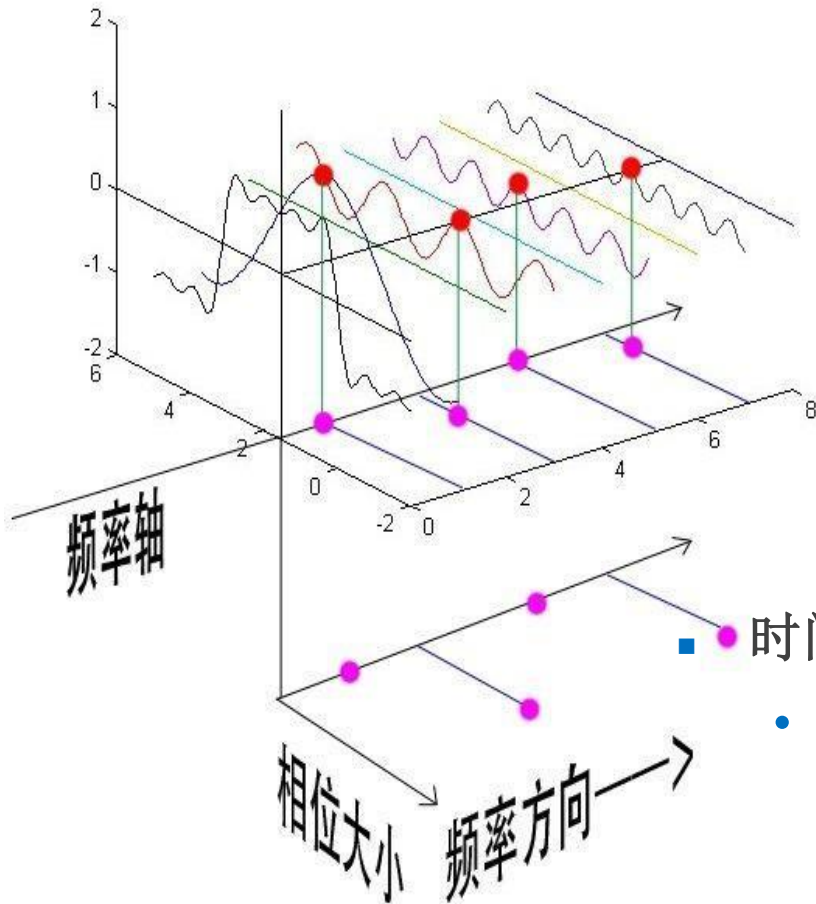
4.1 傅立叶变换基础

• 矩形波对应的频域表示

- 通过时域到频域的变换，我们得到了一个从侧面看的频谱
- 频谱并没有包含时域中全部的信息
- 基础的正弦波 $A \cdot \sin(\omega t + \theta)$ 中，振幅，频率，相位缺一不可，不同相位决定了波的位置，所以对于频域分析



4.1 傅立叶变换基础



■ 时间差并不是相位差

- 如果将全部周期看作 2π 或者 360° 的话，相位差则是时间差在一个周期中所占的比例。我们将时间差除周期再乘 2π ，就得到了相位差。

4.1 傅立叶变换基础

一维傅立叶变换及其反变换

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$F(x) = \int_{-\infty}^{\infty} f(u) e^{j2\pi ux} du$$

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

$$F(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) e^{j2\pi(ux+vy)} du dv$$

离散形式:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad x = 0, 1, 2, \dots, M-1$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad u = 0, 1, 2, \dots, M-1$$

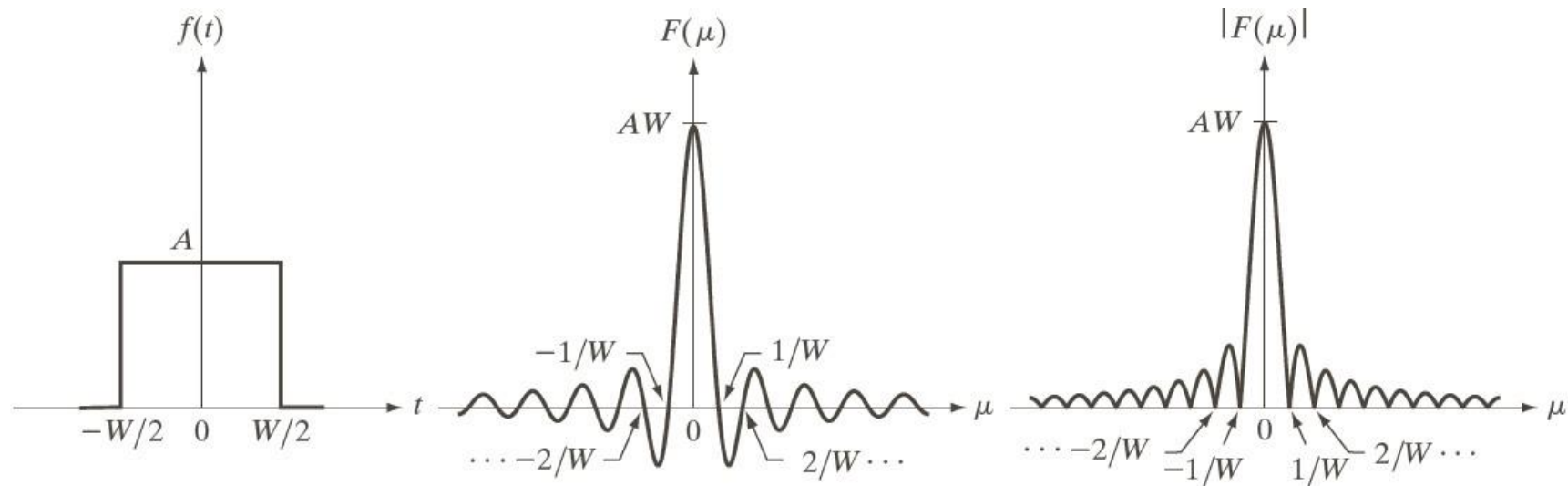
频域→不同的频域成份, 可以表示成极坐标形式:

$$F(u) = |F(u)| e^{-j\phi(u)}$$

$$|F(u)| = [R^2(u) + I^2(u)]^{\frac{1}{2}}$$

$$\phi(u) = \arctan \left[\frac{I(u)}{R(u)} \right]$$

4.1 傅立叶变换基础



a b c

FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.

4.1 傅立叶变换基础

- 傅里叶变换 (FT)、离散时间傅里叶变换 (DTFT) 和离散傅里叶变换 (DFT) 之间的关系

时间函数 \longleftrightarrow 频率函数

连续周期时间、离散非周期频率 — 傅里叶级数 (FS)

连续非周期时间、连续非周期频率 — 傅里叶变换 (FT)

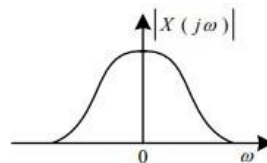
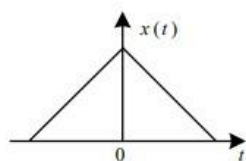
离散非周期时间、连续周期频率 — 序列的傅里叶变换 (DTFT)

离散周期时间、离散周期频率 — 离散傅里叶变换 (DFT)

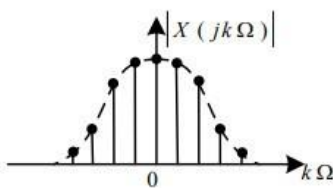
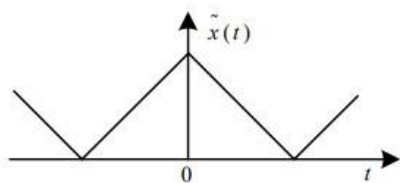
离散周期时间、离散周期频率 — 离散傅里叶级数 (DFS)

- DFS、DTFT与FS、FT的差别在于，前两者都是先在时域上采样，然后进行FS和FT变换，便于计算机进行数字运算和存储。

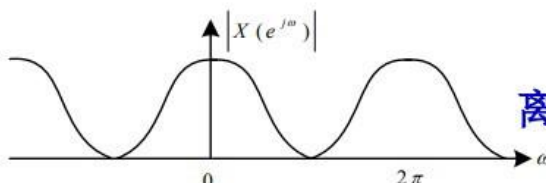
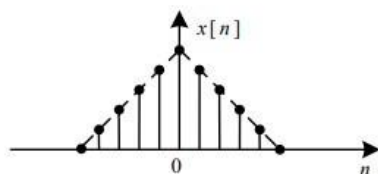
□ 时域的离散造成频域的延拓（周期性）。因而频域的离散也会造成时域的延拓（周期性）。



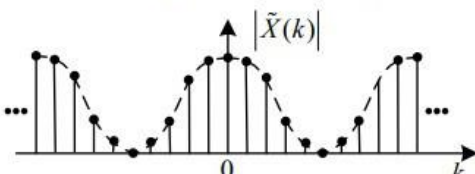
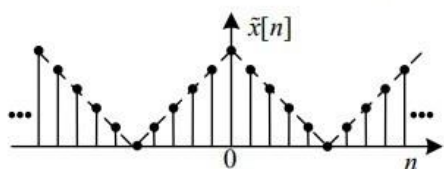
傅里叶变换 (FT)



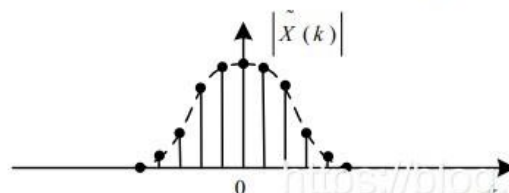
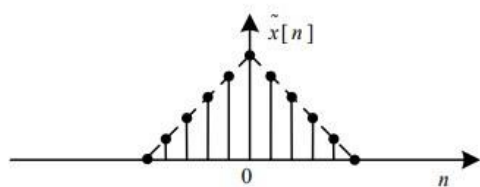
傅里叶级数 (FS)



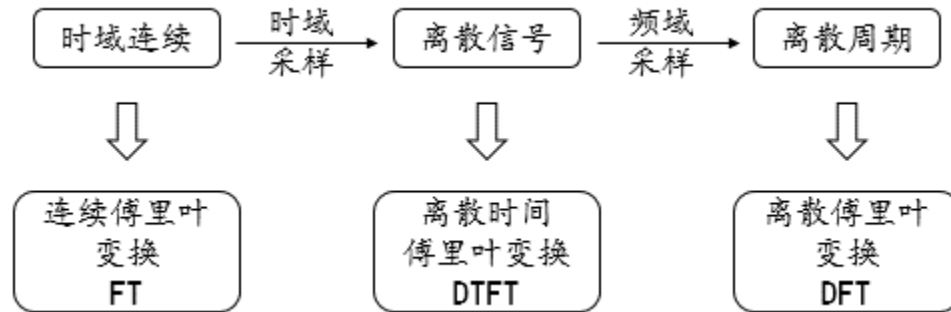
离散时间傅里叶变换 (DTFT)



离散傅里叶级数 (DFS)

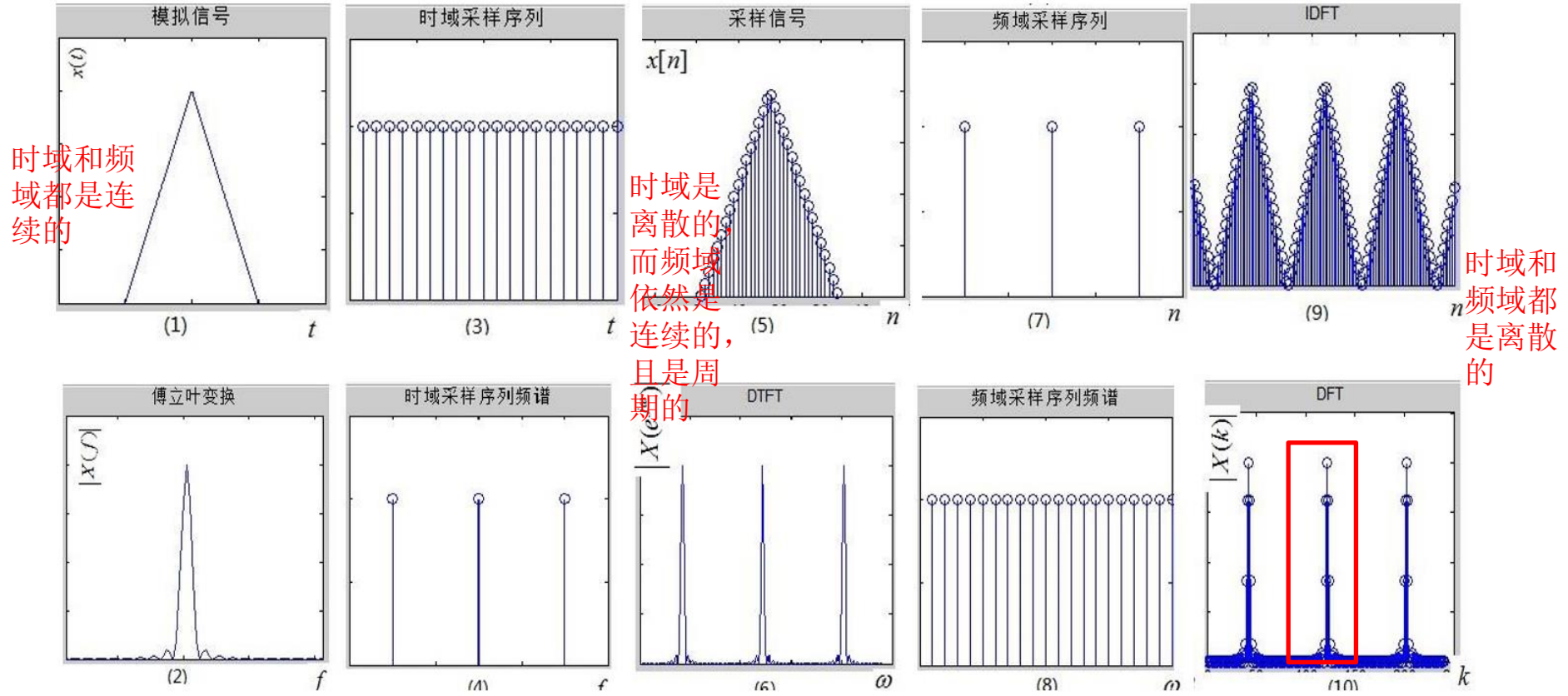


离散傅里叶变换 (DFT)



- DFS是discrete fourier series, 对离散周期信号进行级数展开。**DFT是将DFS取主值, DFS是DFT的周期延拓。**计算上DFS是在时域上先采样后做FS变换, DFT是先在时域上采样, 再FT变换, 即DTFT变换, **然后再将DTFT变换后的连续频域上采样并截取主值。**
- DTFT是对Discrete time fourier transformation, 是对序列的FT, 是先在时域中采样, 后傅里叶变换, 得到**连续的周期谱**, 而DFT, FFT得到是有限长的非周期离散谱。
- DFT只是为了计算机处理方便, 在频率域对DTFT进行的**采样并截取主值**而已。

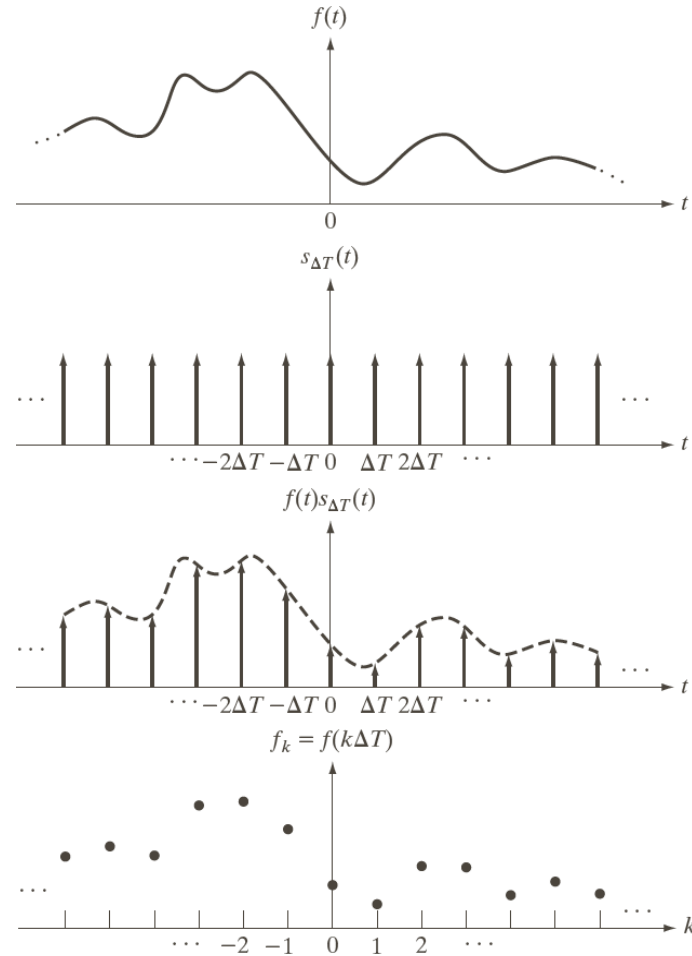
- 频域相乘等于时域卷积，时域相乘相当于频域卷积
- 时域进行采样，则频域周期延拓，同理在频域进行采样，则时域也会周期延拓



时域相乘采样信号（连续变为离散）
 频域卷积采样信号的频谱（变周期连续）

频域相乘采样信号（连续变离散）
 时域卷积采样信号的频谱（非周期变周期）

取样：连续函数转换为离散序列

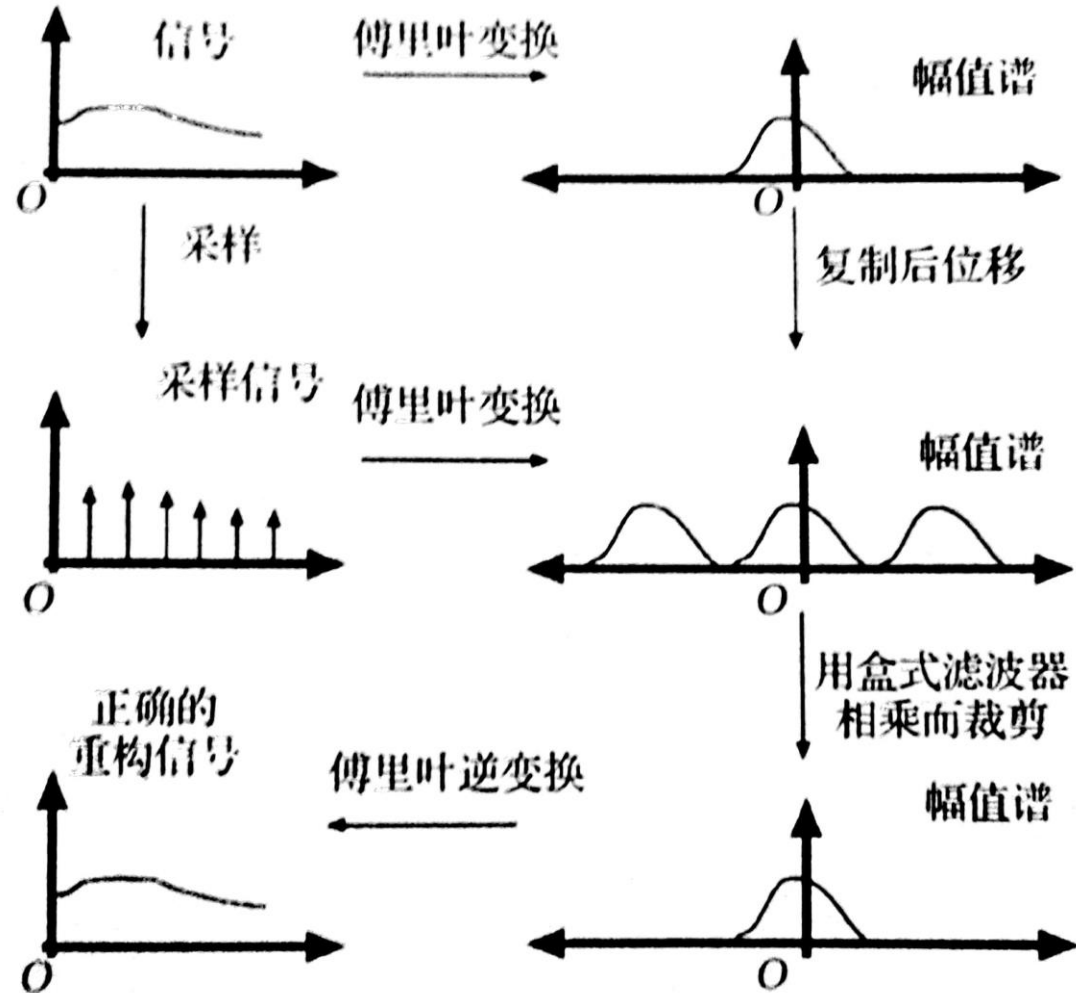


a
b
c
d

FIGURE 4.5

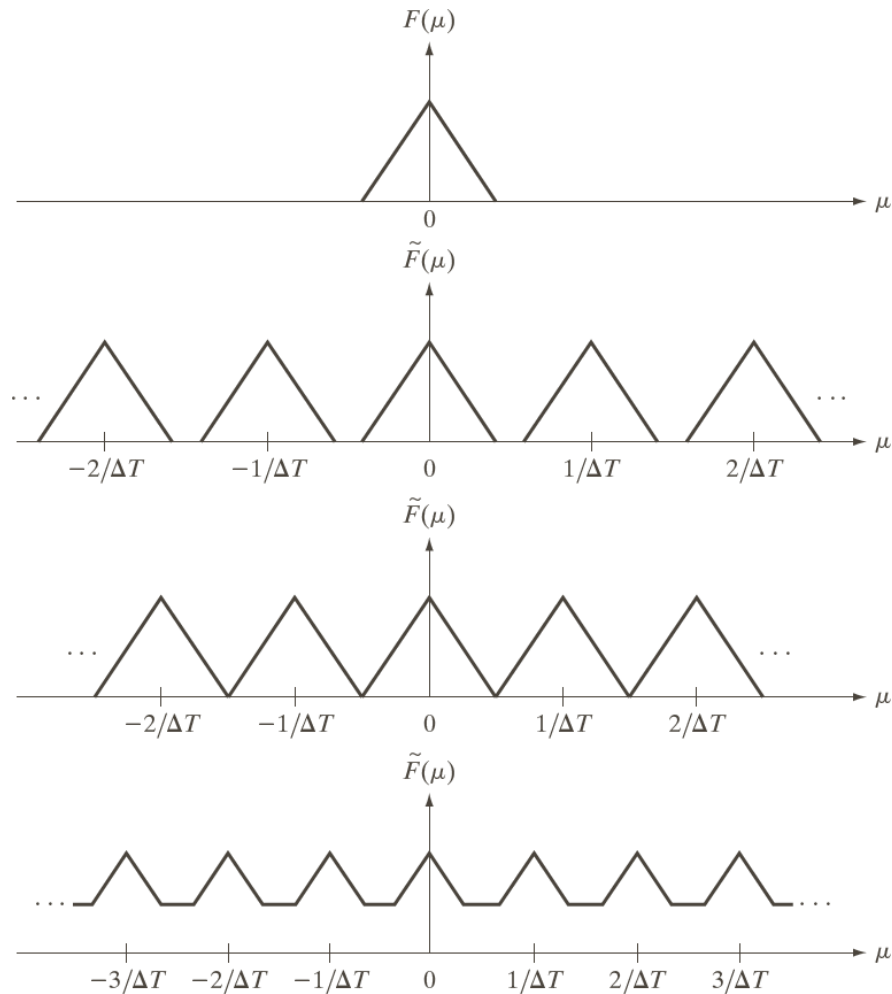
(a) A continuous function. (b) Train of impulses used to model the sampling process. (c) Sampled function formed as the product of (a) and (b). (d) Sample values obtained by integration and using the sifting property of the impulse. (The dashed line in (c) is shown for reference. It is not part of the data.)

• 采样信号重构



取样函数的傅里叶变换

奈奎斯特采样率：
完全等于信号最高频率的两倍的
取样率



a
b
c
d

FIGURE 4.6
(a) Fourier transform of a band-limited function.
(b)–(d) Transforms of the corresponding sampled function under the conditions of over-sampling, critically-sampling, and under-sampling, respectively.

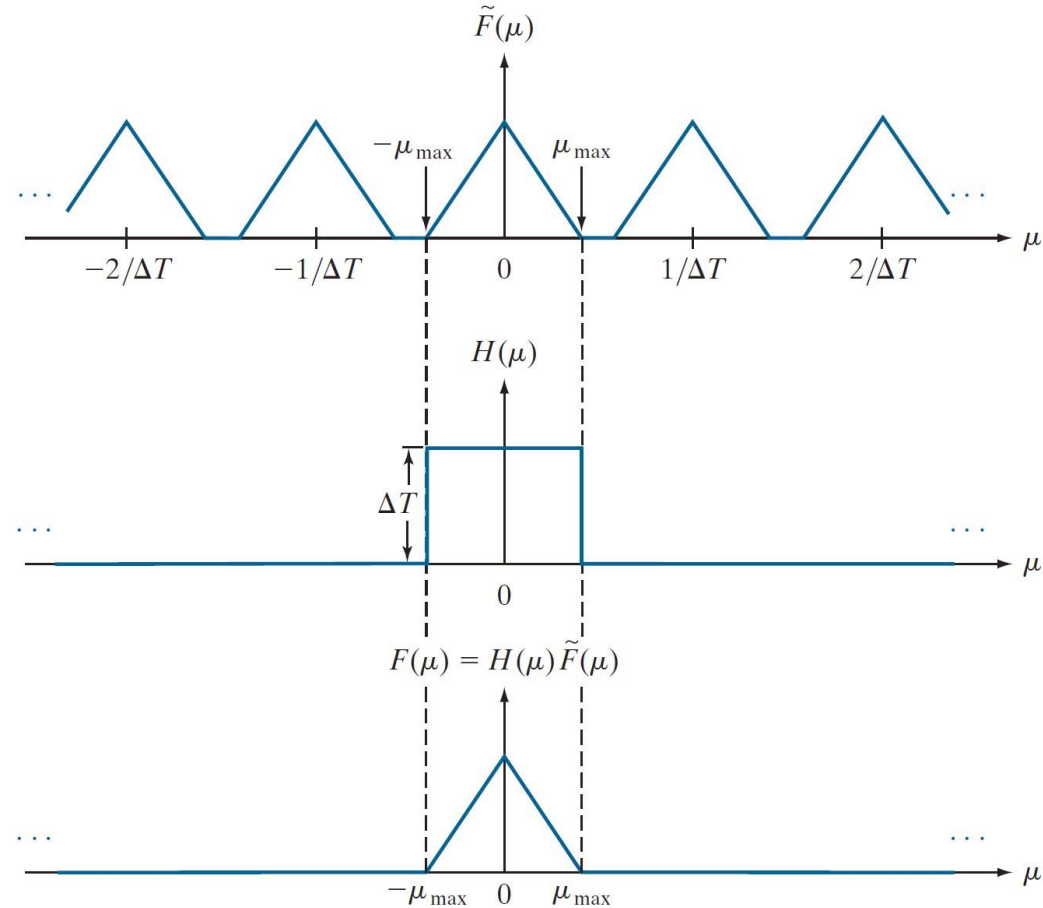
低通滤波提取带限信号

□ 理想低通滤波器的幅度快速过渡特性用物理器件无法实现

a
b
c

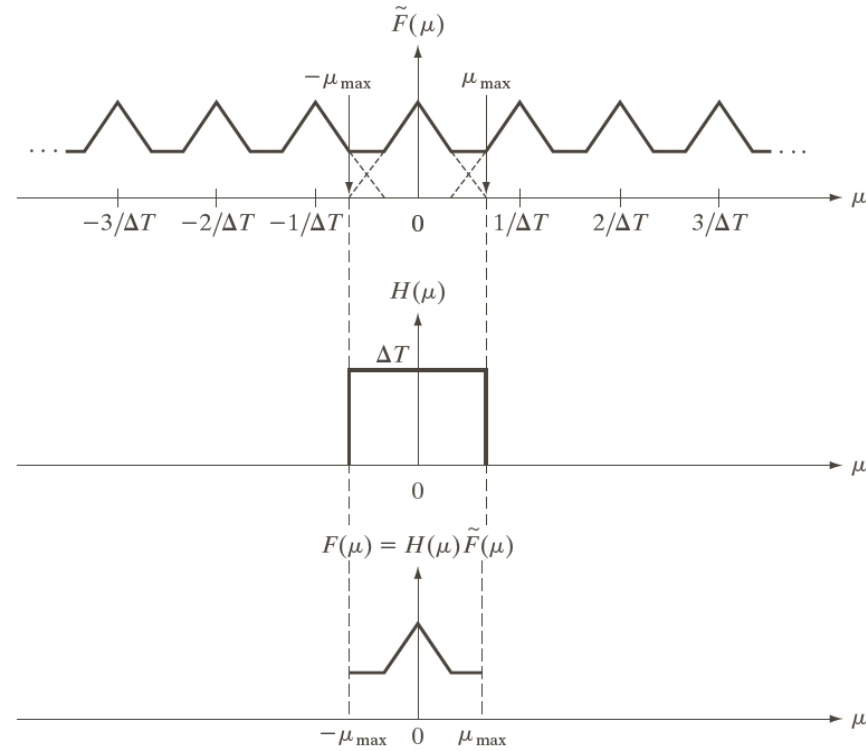
FIGURE 4.8

(a) Fourier transform of a sampled, band-limited function.
 (b) Ideal lowpass filter transfer function.
 (c) The product of (b) and (a), used to extract one period of the infinitely periodic sequence in (a).



混淆

来自邻近周期的频率分量的干扰导致了混淆，妨碍 $F(\mu)$ 的完美复原



a
b
c

FIGURE 4.9 (a) Fourier transform of an under-sampled, band-limited function. (Interference from adjacent periods is shown dashed in this figure). (b) The same ideal lowpass filter used in Fig. 4.8(b). (c) The product of (a) and (b). The interference from adjacent periods results in aliasing that prevents perfect recovery of $F(\mu)$ and, therefore, of the original, band-limited continuous function. Compare with Fig. 4.8.

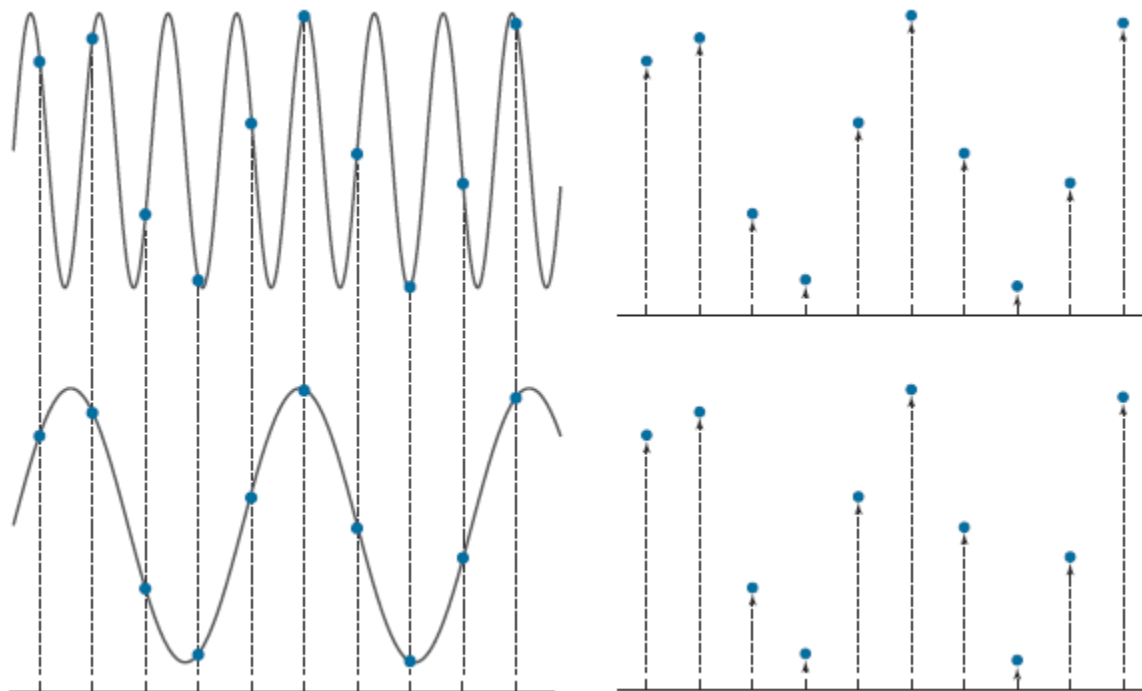
混淆



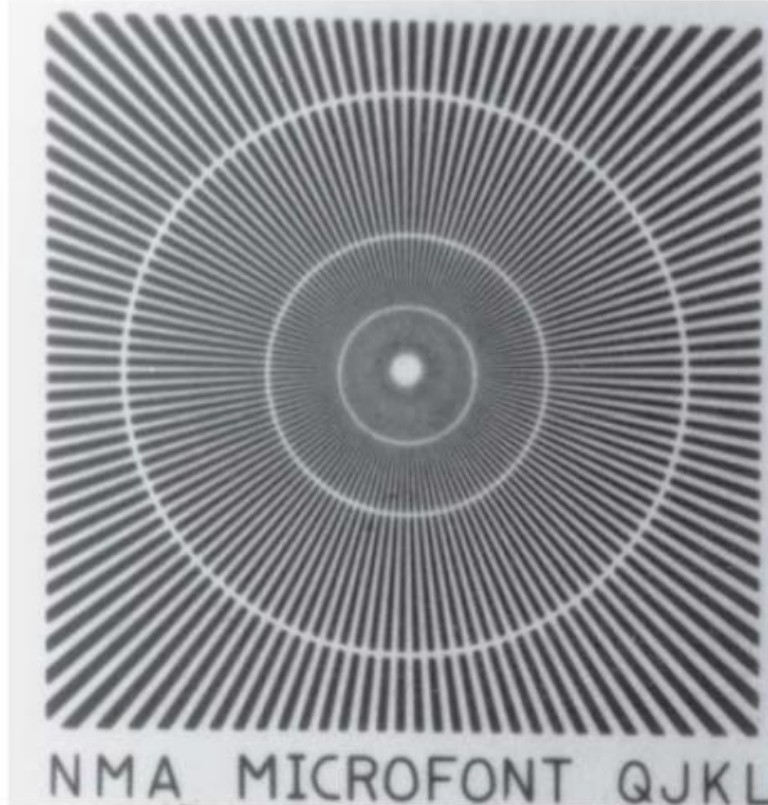
a b
c d

FIGURE 4.9

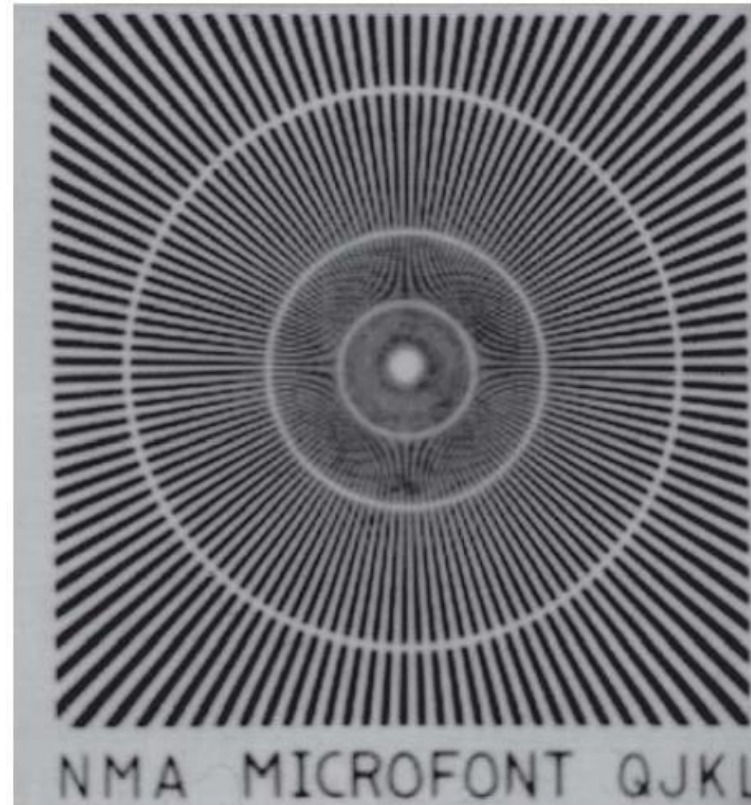
The functions in (a) and (c) are totally different, but their digitized versions in (b) and (d) are identical. Aliasing occurs when the samples of two or more functions coincide, but the functions are different elsewhere.



数字图像中的混淆



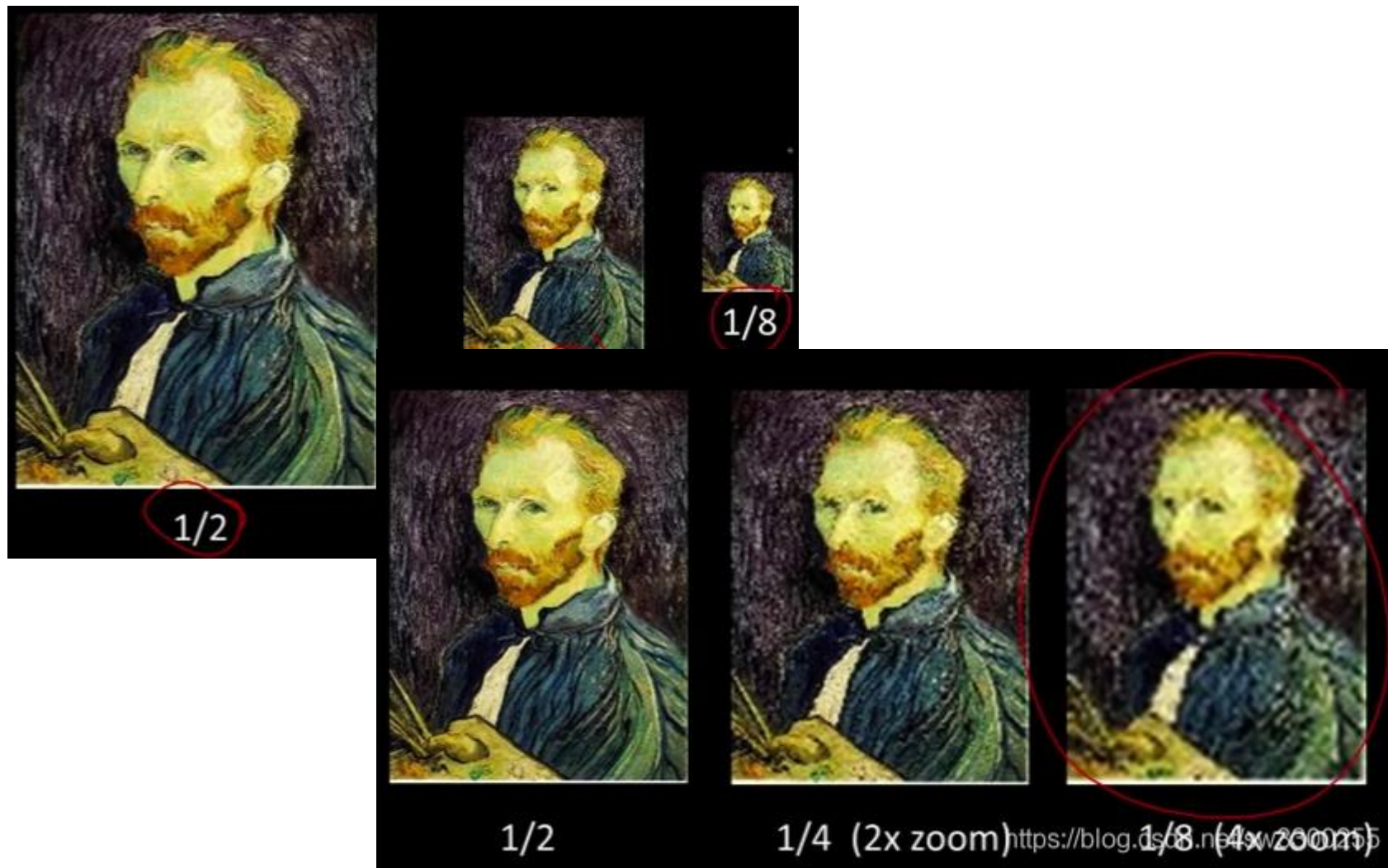
(a) Original image



(b) Sampled image

- 有两种方法可以消除混叠现象。一是直接提高采样频率，以获得更高的尼奎斯特频率，但是采样频率不能无限提高；二是在采样频率固定的情况下，可通过低通滤波器消除大于尼奎斯特频率的高频信号，从而消除混叠现象

数字图像中的混淆



数字图像中的混淆

□ 图像内插和重采样

- 完美重建需要用sinc函数在空间做二维卷积，涉及无限求和
- 实践中采用近似方式
 - ✓ 放大可视为过采样，缩写可视为欠采样

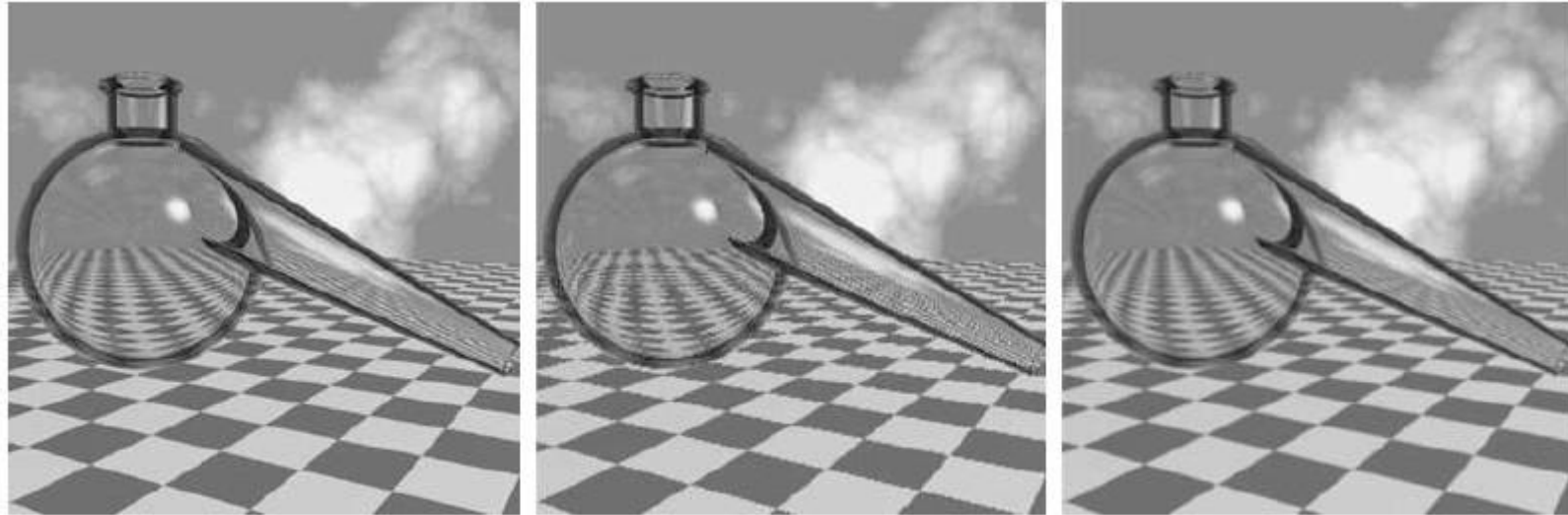


a b c

FIGURE 4.19 Illustration of aliasing on resampled natural images. (a) A digital image of size 772×548 pixels with visually negligible aliasing. (b) Result of resizing the image to 33% of its original size by pixel deletion and then restoring it to its original size by pixel replication. Aliasing is clearly visible. (c) Result of blurring the image in (a) with an averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

数字图像中的混淆

- 图像内插和重采样
 - 图像缩小产生的锯齿现象



a b c

FIGURE 4.18 Illustration of jaggies. (a) A 1024×1024 digital image of a computer-generated scene with negligible visible aliasing. (b) Result of reducing (a) to 25% of its original size using bilinear interpolation. (c) Result of blurring the image in (a) with a 5×5 averaging filter prior to resizing it to 25% using bilinear interpolation. (Original image courtesy of D. P. Mitchell, Mental Landscape, LLC.)



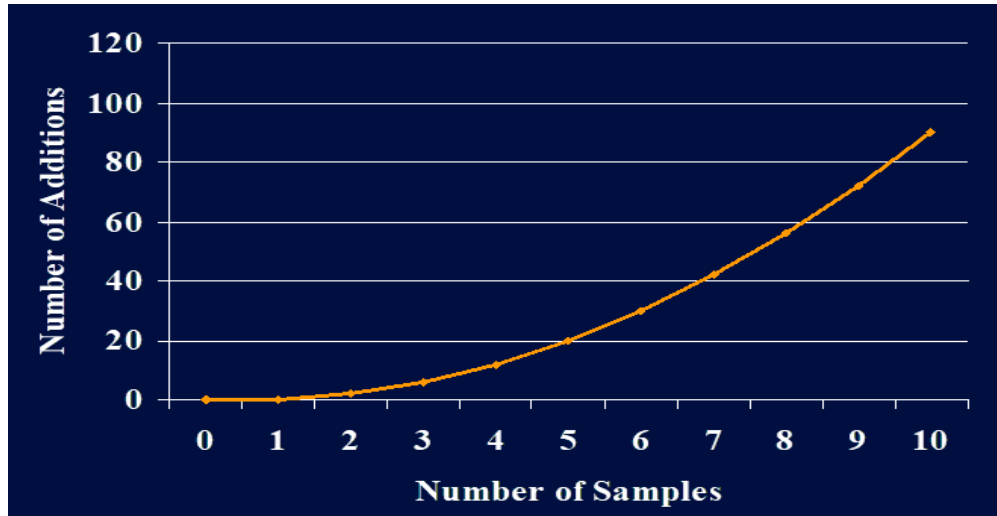
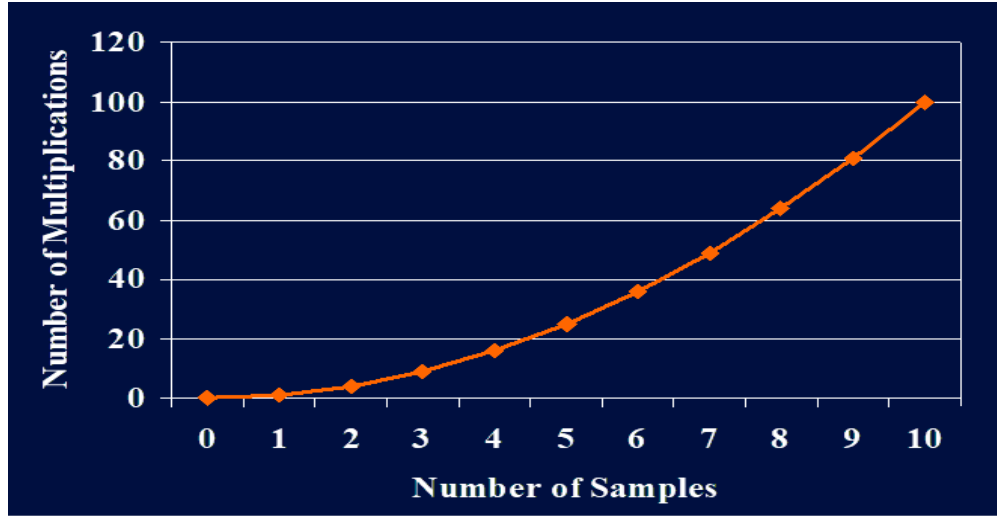
FFT

- 直接应用卷积和相关运算在时域中处理，计算量大，费时，很难达到实时处理的要求。一般可采用DFT方法将输入的数字信号首先进行DFT变换，在频域中进行各种有效的处理，然后进行DFT反变换，恢复为时域信号。
- 离散傅立叶变换(Discrete Fourier Transform--简称DFT)在数字信号处理和数字图像处理中应用十分广泛，它建立了离散时域和离散频域之间的联系。计算机对变换后的信号进行频域处理，比在时域中直接处理更加方便，计算量也大大减少，提高了处理速度。
- DFT还有一个明显的优点是快速算法，即FFT(Fast Fourier Transform)算法。

- 直接计算DFT的计算量与变换区间长度N的平方成正比，当N较大时，计算量太大

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, k = 0, 1, \dots, N-1, W_N = e^{-j\frac{2\pi}{N}}$$

	复数乘法	复数加法
一个 $X(k)$	N	$N-1$
N 个 $X(k)$ (N 点DFT)	N^2	$N(N-1)$



当N较大时，计算量太大，直接用DFT算法进行谱分析和信号的实时处理是不切实际的





1965年发现了DFT的一种快速算法，使DFT的运算效率提高1—2个数量级，为数字信号处理技术应用用于各种信号的实时处理创造了条件，推动了数字处理技术的发展

An Algorithm for the Machine Calculation of Complex Fourier Series

https://www.jstor.org/stable/2003354 翻译此页

作者: JW Cooley - 1965 - 被引用次数: 14440 - 相关文章

An Algorithm for the Machine Calculation of Complex Fourier Series. By James W. Cooley and John W. Tukey. An efficient method for the calculation of the ...

An Algorithm for the Machine Calculation of Complex Fourier Series

By James W. Cooley and John W. Tukey

An efficient method for the calculation of the intersections of a 2^m factorial experiment was introduced by Yates and is widely known by his name. The generalization to 2^n was given by Box et al. [1]. Good [2] generalized these methods and gave elegant algorithms for which one class of applications is the calculation of Fourier series. In their full generality, Good's methods are applicable to certain problems in which one must multiply an N-vector by an N x N matrix which can be factored into m sparse matrices, where m is proportional to log N. This results in a procedure requiring a number of operations proportional to N log N rather than N^2. These methods are applied here to the calculation of complex Fourier series. They are useful in situations where the number of data points N, or can be chosen to be, a highly composite number. The algorithm is here derived and presented in a rather different form. Attention is given to the choice of N. It is also shown how special advantage can be obtained in the use of a binary computer with N = 2^n and how the entire calculation can be performed within the array of N data storage locations used for the given Fourier coefficients.

Consider the problem of calculating the complex Fourier series

(1) X(j) = sum_{k=0}^{N-1} A(k) * W^{jk}, j = 0, 1, ..., N-1,

where the given Fourier coefficients A(k) are complex and W is the principal Nth root of unity, W = e^{i2*pi/N}.

A straightforward calculation using (1) would require N^2 operations where "operation" means as it will throughout this note, a complex multiplication followed by a complex addition.

The algorithm described here iterates on the array of given complex Fourier amplitudes and yields the result in less than 2N log N operations without requiring more data storage than is required for the given array A. To derive the algorithm, suppose N is a composite, i.e., N = r_1 r_2. Then let the indices in (1) be expressed

(2) j = j_1 + j_2, j_1 = 0, 1, ..., r_1 - 1, j_2 = 0, 1, ..., r_2 - 1,

(3) k = k_1 + k_2, k_1 = 0, 1, ..., r_1 - 1, k_2 = 0, 1, ..., r_2 - 1.

Then, one can write

(4) X(j_1, j_2) = sum_{k_1=0}^{r_1-1} sum_{k_2=0}^{r_2-1} A(k_1, k_2) * W^{j_1 k_1 + j_2 k_2}

Received August 17, 1964. Research in part at Princeton University under the sponsorship of the Army Research Office (Durham). The authors wish to thank Richard Garwin for his crucial role in communication and encouragement.

MACHINE CALCULATION OF COMPLEX FOURIER SERIES

298

Since

(5) W^{r_1 k_1} = W^{i k_1 r_1}

the inner sum, over k_1, depends only on j_1 and k_2 and can be defined as a new array,

(6) A_1(j_1, k_2) = sum_{k_1=0}^{r_1-1} A(k_1, k_2) * W^{i k_1 r_1}

The result can then be written

(7) X(j_1, j_2) = sum_{k_2=0}^{r_2-1} A_1(j_1, k_2) * W^{j_2 k_2 + i k_2 r_1 j_1}

There are N elements in the array A_1, each requiring r_1 operations, giving a total of N r_1 operations to obtain A_1. Similarly, it takes N r_2 operations to calculate X from A_1. Therefore, this two-step algorithm, given by (6) and (7), requires a total of

(8) T = N(r_1 + r_2)

operations. It is easy to see how successive applications of the above procedure, starting with its application to (6), give an n-step algorithm requiring

(9) T = N(r_1 + r_2 + ... + r_n)

operations, where

(10) N = r_1 r_2 ... r_n.

If r_j = s_j t_j with s_j, t_j > 1, then s_j + t_j < r_j unless s_j = t_j = 2, when s_j + t_j = r_j. In general, then, using as many factors as possible provides a minimum to (9), but factors of 2 can be combined in pairs without loss. If we are able to choose N to be highly composite, we may make very real gains. If all r_j are equal to r, then, from (10) we have

(11) m = log N

and the total number of operations is

(12) T(r) = r^m log N.

If N = r^m t^p, then we find that

(13) T = m r + p r + r^p

log N = m log r + p log s + p log t + ...

so that

(14) T/N = m/r + p/s + p/t + ...

is a weighted mean of the quantities

(15) r/log r, s/log s, t/log t, ...

whose values run as follows

Table with 2 columns: r, r/log r. Values: 2, 1.00; 3, 1.10; 4, 1.18; 5, 1.25; 6, 1.31; 7, 1.38; 8, 1.43; 9, 1.47; 10, 1.50.

The use of r_j = 3 is formally most efficient, but the gain is only about 6% over the use of 2 or 4, which have other advantages. If necessary, the use of r_j up to 10 can increase the number of computations by no more than 50%. Accordingly, we can find "highly composite" values of N within a few percent of any given large number.

Whenever possible, the use of N = r^m with r = 2 or 4 offers important advantages for computers with binary arithmetic, both in addressing and in multiplication economy.

The algorithm with r = 2 is derived by expressing the indices in the form

(14) j = j_{-1} 2^{m-1} + ... + j_1 2 + j_0,

k = k_{-1} 2^{m-1} + ... + k_1 2 + k_0,

where j_i and k_i are equal to 0 or 1 and are the contents of the respective bit positions in the binary representation of j and k. All arrays will now be written as functions of the bits of their indices. With this convention (1) is written

(15) X(j_{-1}, ..., j_0) = sum_{k_{-1}=0}^{2^{m-1}-1} ... sum_{k_0=0}^{2^0-1} A(k_{-1}, ..., k_0) * W^{j_{-1} k_{-1} + ... + j_0 k_0},

where the sums are over k_i = 0, 1. Since

(16) W^{2^{m-1} k_{-1}} = W^{2^{m-1} k_{-1}}

the innermost sum of (15), over k_{-1}, depends only on j_{-1}, k_{-2}, ..., k_0 and can be written

(17) A_1(j_{-1}, k_{-2}, ..., k_0) = sum_{k_{-1}=0}^{2^{m-1}-1} A(k_{-1}, ..., k_0) * W^{j_{-1} k_{-1} + ... + j_0 k_0}

Proceeding to the next innermost sum, over k_{-2}, and so on, and using

(18) W^{(j_{-1} 2^{m-1} + ... + j_1 2 + j_0) k_{-2}} = W^{j_{-1} k_{-2} + ... + j_0 k_{-2}}

one obtains successive arrays,

(19) A_1(j_{-1}, j_{-2}, ..., j_{-m+1}, k_{-m+1}, ..., k_0) * W^{j_{-1} k_{-1} + ... + j_{-m+1} k_{-m+1}}

for l = 1, 2, ..., m.

MACHINE CALCULATION OF COMPLEX FOURIER SERIES

300

Writing out the sum this appears as

(20) A_1(j_{-1}, ..., j_{-m+1}, k_{-m+1}, ..., k_0) = sum_{k_{-1}=0}^{2^{m-1}-1} ... sum_{k_0=0}^{2^0-1} A(k_{-1}, ..., k_0) * W^{j_{-1} k_{-1} + ... + j_{-m+1} k_{-m+1}}

According to the indexing convention, this is stored in a location whose index is

(21) j_{-1} 2^{m-1} + ... + j_{-m+1} 2^{m-m+1} + k_{-m+1} 2^{m-m+1} + ... + k_0

It can be seen in (20) that only the two storage locations with indices having 0 and 1 in the 2^{m-1} bit position are involved in the computation. Parallel computation is permitted since the operation described by (20) can be carried out with all values of j_{-1}, ..., j_{-m+1}, and k_{-m+1}, ..., k_0 simultaneously. In some applications it is convenient to use (20) to express A_1 in terms of A_{l-1}, giving what is equivalent to an algorithm with r = 4.

The last array calculated gives the desired Fourier sums,

(22) X(j_{-1}, ..., j_0) = A_1(j_{-1}, ..., j_0)

in such an order that the index of an X must have its binary bits put in reverse order to yield its index in the array A_1.

In some applications, where Fourier sums are to be evaluated twice, the above procedure could be programmed so that no bit-inversion is necessary. For example, consider the solution of the difference equation,

(23) aX(j+1) + bX(j) + cX(j-1) = F(j).

The present method could be first applied to calculate the Fourier amplitudes of F(j) from the formula

(24) B(k) = 1/N sum_{j=0}^{N-1} F(j) W^{jk}

The Fourier amplitudes of the solution are, then,

(25) A(k) = B(k) / (aW^{k+1} + b + cW^{-k})

The B(k) and A(k) arrays are in bit-inverted order, but with an obvious modification of (20), A(k) can be used to yield the solution with correct indexing.

A computer program for the IBM 7094 has been written which calculates three-dimensional Fourier sums by the above method. The computing time taken for computing three-dimensional 2^m x 2^m x 2^m arrays of data points was as follows:

* A multiple processing circuit using this algorithm was designed by R. E. Miller and S. Winograd of the IBM Watson Research Center. In this case r = 4 was found to be most practical.

MACHINE CALCULATION OF COMPLEX FOURIER SERIES

301

Table with 6 columns: a, b, c, No. Pts., Time (minutes). Values: 4, 4, 2, 2^10, .02; 11, 0, 0, 2^10, .02; 4, 4, 4, 2^10, .04; 12, 0, 0, 2^10, .07; 5, 4, 4, 2^10, .10; 5, 5, 2, 2^10, .12; 13, 0, 0, 2^10, .13

IBM Watson Research Center, Yorktown Heights, New York

Bull. Telephone Laboratories, Murray Hill, New Jersey

Princeton University, Princeton, New Jersey

J. G. R. F. Box, L. R. Cooley, W. R. Cooley, O. L. Davies (Ed.), F. R. Hinesworth & G. P. Sinton, The Design and Analysis of Industrial Experiments, Oliver & Boyd, Edinburgh, 1961.

J. J. Good, The interaction algorithm and practical Fourier series, J. Roy. Statist. Soc. Ser. B, v. 23, 1961, p. 361-372; Mathematics, v. 23, 1960, p. 372-375. MR 24 #1674, MR 23 #3481.

减少运算量的思路和方法

- N点DFT的复乘次数等于 N^2 。把N点DFT分解为几个较短的DFT，可使乘法次数大大减少
- 同时充分利用旋转因子 W_N^m 的周期性和对称性

周期性:
$$W_N^{m+lN} = e^{-j\frac{2\pi}{N}(m+lN)} = e^{-j\frac{2\pi}{N}m} = W_N^m$$

对称性:
$$W_N^{-m} = W_N^{N-m} \quad [W_N^{N-m}]^* = W_N^m$$

$$W_N^{m+\frac{N}{2}} = -W_N^m$$

(1) 按n的奇偶把x(n)分解为两个N/2点的子序列

$$X(k) = \sum_{n=\text{偶数}} x(n)W_N^{kn} + \sum_{n=\text{奇数}} x(n)W_N^{kn}$$

$$= \sum_{r=0}^{N/2-1} x(2r)W_N^{2kr} + \sum_{r=0}^{N/2-1} x(2r+1)W_N^{k(2r+1)} \quad \left\{ \begin{array}{l} x_1(r) = x(2r), \\ x_2(r) = x(2r+1) \end{array} \right.$$

$$= \sum_{r=0}^{N/2-1} x_1(r)W_N^{2kr} + W_N^k \sum_{r=0}^{N/2-1} x_2(r)W_N^{2kr} \quad r = 0, 1, \dots, \frac{N}{2} - 1$$

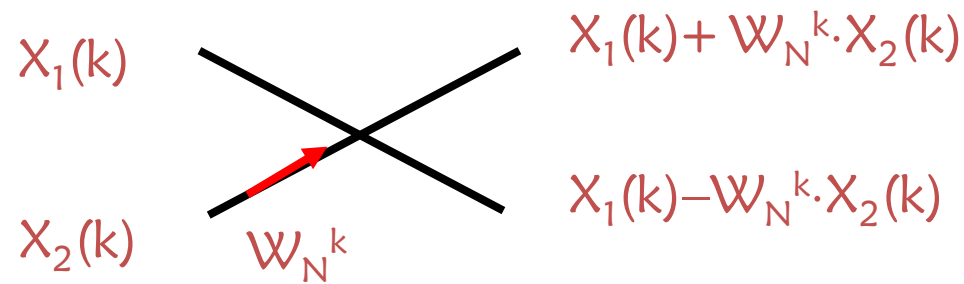
$$= \sum_{r=0}^{N/2-1} x_1(r)W_{N/2}^{kr} + W_N^k \sum_{r=0}^{N/2-1} x_2(r)W_{N/2}^{kr} = X_1(k) + W_N^k X_2(k)$$

注意： 这里的k的取值范围为0, 1, ..., N-1

由于 $X_1(k)$ 和 $X_2(k)$ 均以 $N/2$ 为周期，且 $W_N^{k+\frac{N}{2}} = -W_N^k$
 $X(k)$ 又可表示为：

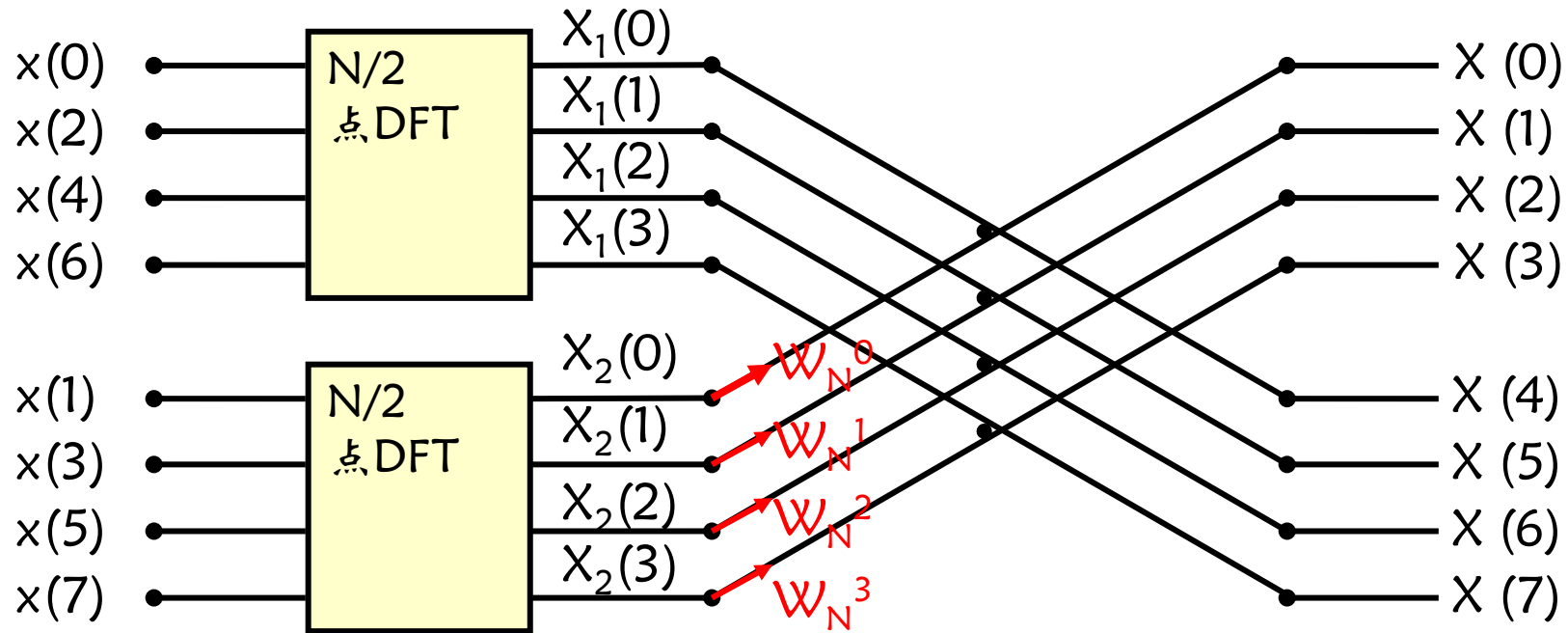
$$\begin{cases} X(k) = X_1(k) + W_N^k X_2(k) \\ X(k + \frac{N}{2}) = X_1(k) - W_N^k X_2(k) \end{cases} \quad k = 0, 1, \dots, \frac{N}{2} - 1$$

这样将 N 点DFT分解为两个 $N/2$ 点的DFT



图：蝶形运算符号

完成一个蝶形运算需要一次复数乘和两次复数加法运算，经过一次分解后，共需要复数乘和复数加的次数为 $2(N/2)^2 + N/2$ 和 $N^2/2$



$$\left\{ \begin{array}{l} X(k) = X_1(k) + W_N^k X_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 \\ X(k + \frac{N}{2}) = X_1(k) - W_N^k X_2(k) \quad k = 0, 1, \dots, \frac{N}{2} - 1 \end{array} \right.$$

分解后的运算量:

	复数乘法	复数加法
一个 $N/2$ 点DFT	$(N/2)^2$	$N/2(N/2-1)$
两个 $N/2$ 点DFT	$N^2/2$	$N(N/2-1)$
一个蝶形	1	2
$N/2$ 个蝶形	$N/2$	N
总计	$N^2/2 + N,$ $\approx N^2/2$	$N(N/2 - 1) + N$ $\approx N^2/2$

运算量减少了近一半

(3) 第二次分解:

- 将 $x_1(r)$ 按 r 取奇、偶可分解成2个长度为 $N/4$ 的子序列

$$x_3(l) = x_1(2l), \quad l=0,1,\dots, N/4-1;$$

$$x_4(l) = x_1(2l+1),$$

根据上面推导可得: $X_1(k) = X_3(k) + W_{N/2}^k \cdot X_4(k), \quad k=0,1,\dots,N/2-1$

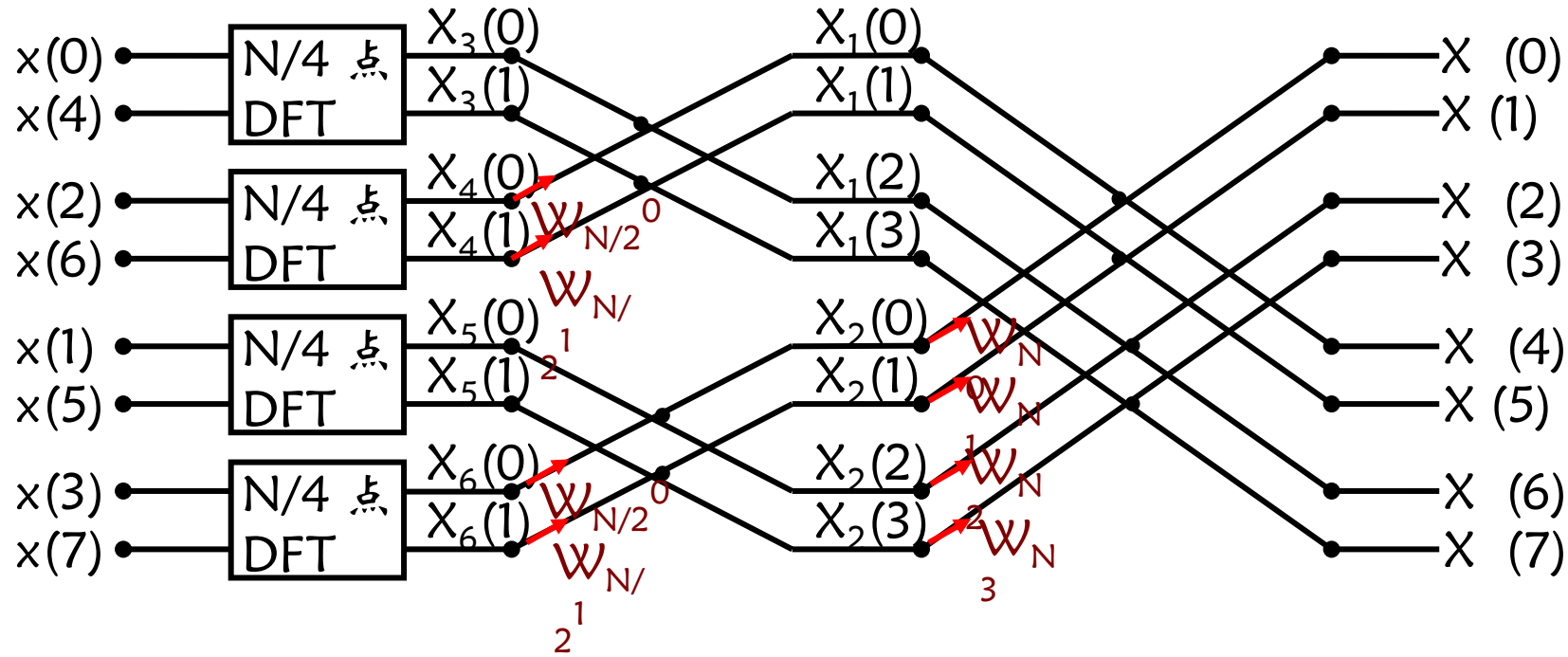
$$\left\{ \begin{array}{l} X_1(k) = X_3(k) + W_{N/2}^k \cdot X_4(k), \quad k=0,1,\dots,N/4-1; \\ X_1(k+N/2) = X_3(k) - W_{N/2}^k \cdot X_4(k), \quad k=0,1,\dots,N/4-1; \end{array} \right.$$

- 将 $x_2(r)$ 按 r 取奇、偶可分解成2个长 $N/4$ 的子序列

$$x_5(l) = x_2(2l), \quad l=0,1,\dots, N/4$$

$$x_6(l) = x_2(2l+1); \quad \text{同理得}$$

$$\left\{ \begin{array}{l} X_2(k) = X_5(k) + W_{N/2}^k \cdot X_6(k), \quad k=0,1,\dots,N/4-1; \\ X_2(k+N/2) = X_5(k) - W_{N/2}^k \cdot X_6(k), \quad k=0,1,\dots,N/4-1; \end{array} \right.$$

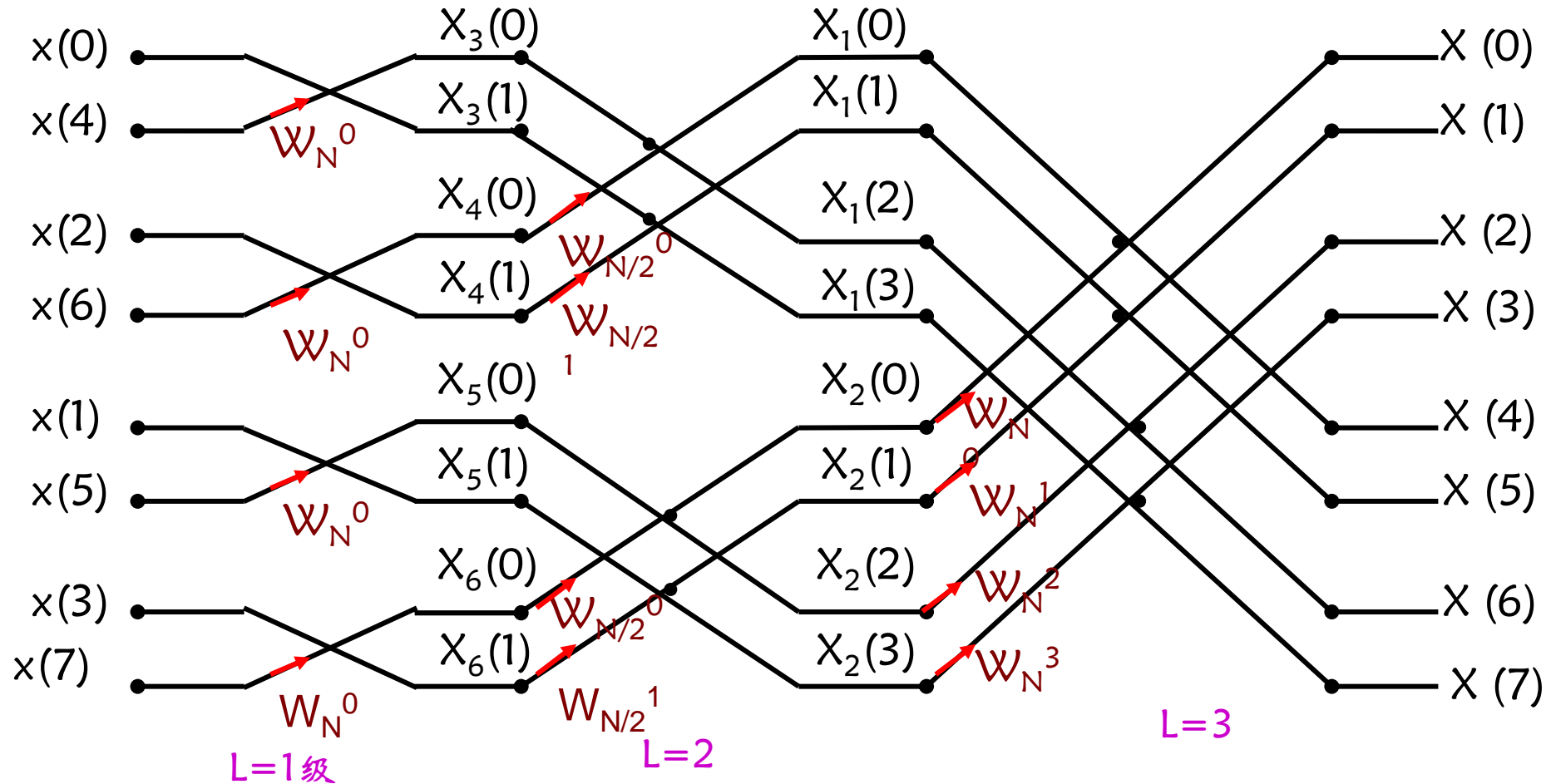


$$\left\{ \begin{array}{l} X_1(k) = X_3(k) + W_{N/2}^k \cdot X_4(k) \\ X_1(k+N/2) = X_3(k) - W_{N/2}^k \cdot X_4(k) \end{array} \right.$$

$$\left\{ \begin{array}{l} X_2(k) = X_5(k) + W_{N/2}^k \cdot X_6(k) \\ X_2(k+N/2) = X_5(k) - W_{N/2}^k \cdot X_6(k) \end{array} \right.$$

$$k=0,1,\dots,N/4-1$$

再次分解，对N=8点，可分解三次



运算量比较

1、直接DFT运算N点运算：

复数乘次数： $N \times N$

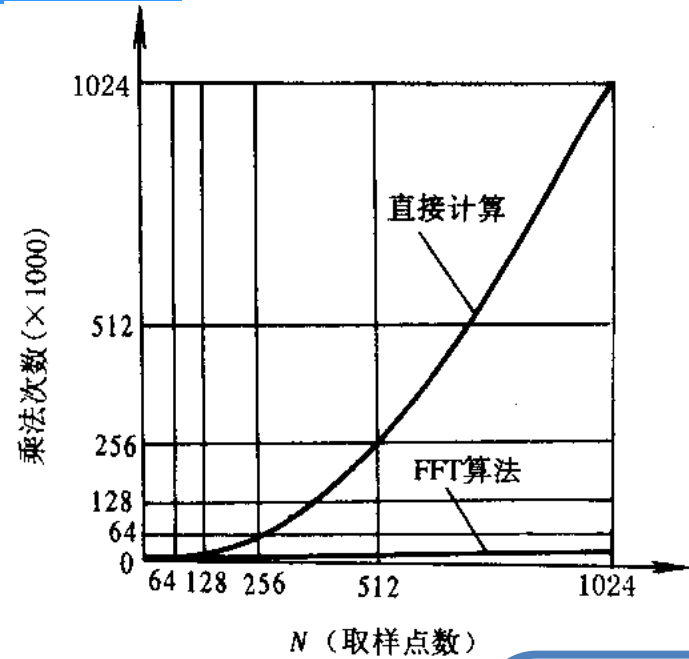
复数加次数： $N \times (N-1)$

2、用DIT-FFT作N点运算：

复数乘次数： $M \times N/2 = N/2 \times \log_2 N$

复加次数： $2 \times N/2 \times M = N \times \log_2 N$

FFT大大减少了运算次数，提高了运算速度



1000采样点，使用FFT算法只需要计算3000次，而常规的DFT算法需要计算1000000次

4.1 离散傅立叶变换(DFT)

二维DFT及其反变换

一个图像关于的函数的离散

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$$

谱: $|F(u, v)| = \left[R^2(x, y) + I^2(x, y) \right]^{\frac{1}{2}}$

相角: $\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$

$$u = 0.1.2 \dots M-1$$

$$v = 0.1.2 \dots N-1$$

$$x = 0.1.2 \dots M-1$$

$$y = 0.1.2 \dots N-1$$

信号处理

图像处理

时间域

空间域

一维

二维

频率域

空间频率域

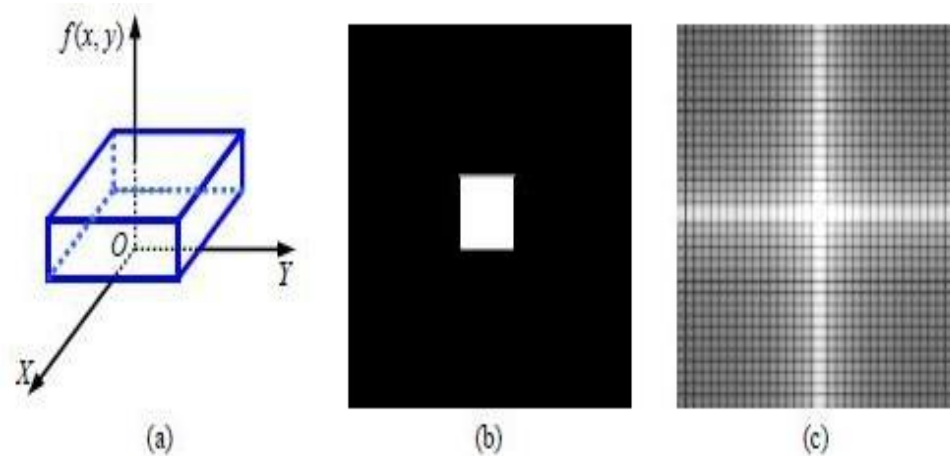


图 4.1.1 2-D 图像函数和傅里叶频谱的显示



2D DFT的性质：可分离性

$$\begin{aligned} F(u, v) &= \sum_y \sum_x \left\{ f(x, y) e^{-j2\pi ux/N} \right\} \cdot e^{-j2\pi vy/N} \\ &= \sum_{y=0}^{N-1} f(u, y) e^{-j2\pi vy/N} \end{aligned}$$

一个二维正（反）傅立叶变换可分解为二个一维正（反）傅立叶变换

- ① Y不变，X方向每一行一维变换FT
- ② X不变，Y方向每一列一维变换FT

2D DFT的性质：空间位移

↪ 空间位移

$x \Rightarrow x_0, y \Rightarrow y_0$

$$f(x, y) \Leftrightarrow F(u, v)$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-j2\pi(ux_0 + vy_0)/N}$$

$$F(u, v) e^{-j2\pi(ux_0 + vy_0)/N}$$

$$= |F(u, v)| e^{j\theta(u, v)} |e^{-j2\pi(ux_0 + vy_0)/N}|$$

$$= |F(u, v)| e^{j[\theta(u, v) - 2\pi(ux_0 + vy_0)/N]}$$

幅值

相位

幅值不变 $\Rightarrow |F(u, v)|$

相角变化 $\Rightarrow -2\pi(ux_0 + vy_0)/N$

任何一点的FT可以用某点的FT乘以相角的变化得到



2D DFT的性质：频率位移，周期性

↳ 频率位移

$u \Rightarrow u_0, v \Rightarrow v_0$

$$f(x, y) \Leftrightarrow F(u, v)$$

$$f(x, y)e^{j2\pi(u_0x+v_0y)/N} \Leftrightarrow F(u-u_0, v-v_0)$$

↳ 周期性

$$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N) = F(u + k_1M, v + k_2N)$$

$$f(x, y) = f(x + k_1M, y) = f(x, y + k_2N) = f(x + k_1M, y + k_2N)$$



二维DFT的性质

- 线性 $f_1(x, y) + f_2(x, y)$ $F_1(u, v) + F_2(u, v)$
- 比例 $f(ax, by)$ $\frac{1}{ab} F\left(\frac{u}{a}, \frac{v}{b}\right)$
- 平移 $f(x - a, y - b)$ $e^{-j2\pi(au+bv)} F(u, v)$
 $e^{j2\pi(cx+dy)} f(x, y)$ $F(u - c, v - d)$
- 卷积 $f_1(x, y) * f_2(x, y)$ $F_1(u, v)F_2(u, v)$
 $f_1(x, y)f_2(x, y)$ $F_1(u, v) * F_2(u, v)$
- 旋转 $f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$
 $F(u \cos \theta + v \sin \theta, -u \sin \theta + v \cos \theta)$

2D DFT及其反变换的一些性质

Spatial Domain [†]		Frequency Domain [†]	
1)	$f(x, y)$ real	\Leftrightarrow	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	\Leftrightarrow	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	\Leftrightarrow	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	\Leftrightarrow	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	\Leftrightarrow	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	\Leftrightarrow	$F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	\Leftrightarrow	$F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	\Leftrightarrow	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	\Leftrightarrow	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	\Leftrightarrow	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	\Leftrightarrow	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	\Leftrightarrow	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	\Leftrightarrow	$F(u, v)$ complex and odd

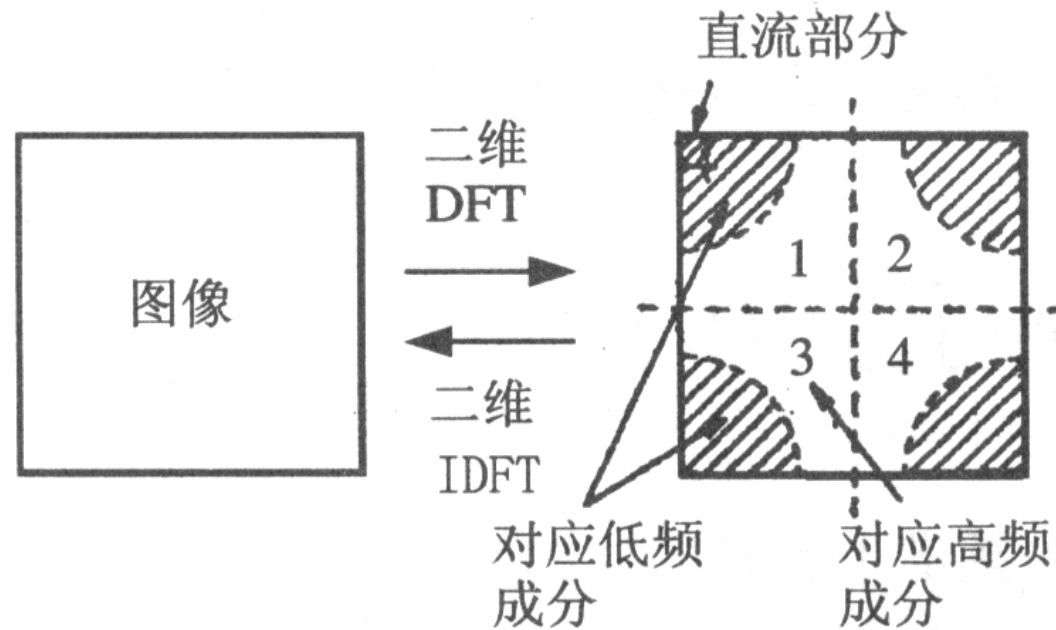
TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

[†]Recall that $x, y, u,$ and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and $y,$ and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

频谱图像中心化

➤ 二维离散傅立叶变换的显示

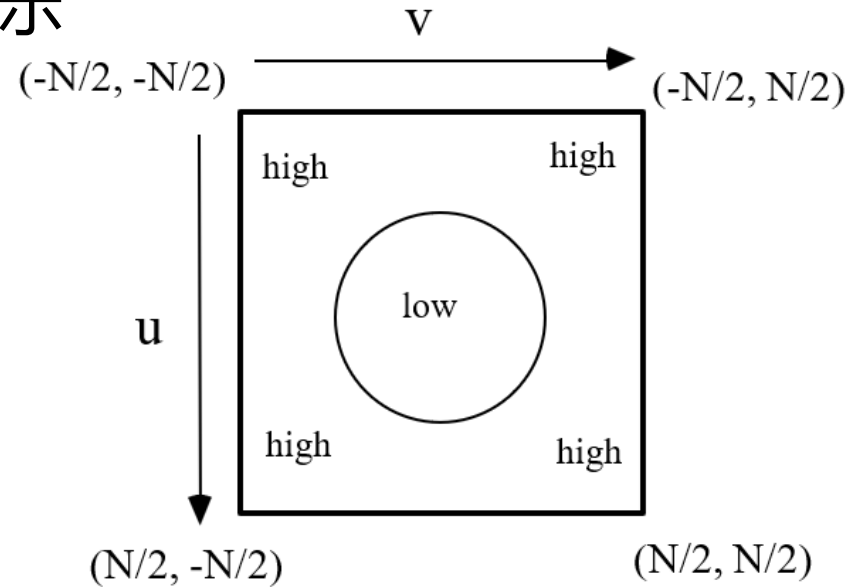
- ✓ 按照标准的傅里叶变换公式，其幅度谱的强度分布具有下列特性：



频谱图像中心化

➤ 二维离散傅立叶变换的显示

- ✓ 在光学傅立叶变换中，人们已习惯于变化领域中的低谱部分位于中央
- ✓ 使频域的频谱分布中间低、周围高，有利于对频谱的解释和进行各种计算与分析



➤ 变化最慢的频率成分 ($u=v=0$) 对应一幅图像的平均灰度级

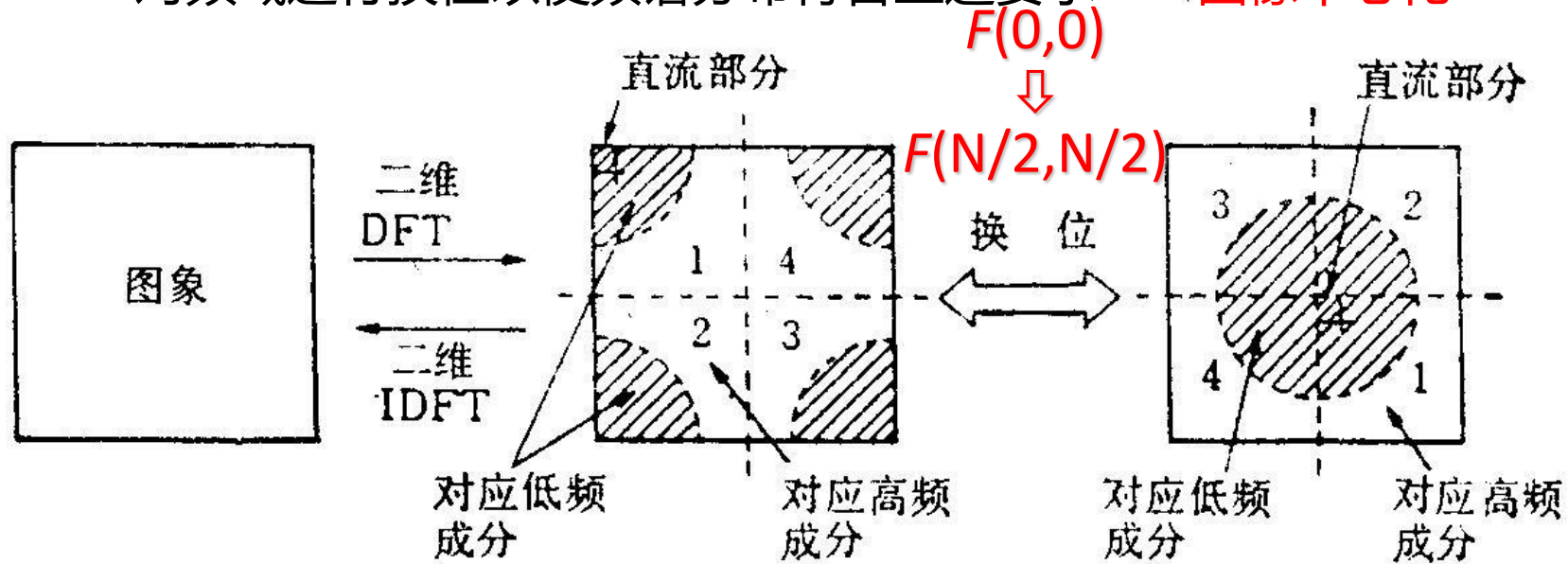
$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

● $F(0,0)$ 有时也称为频谱的直流 (dc) 分量

频谱图像中心化

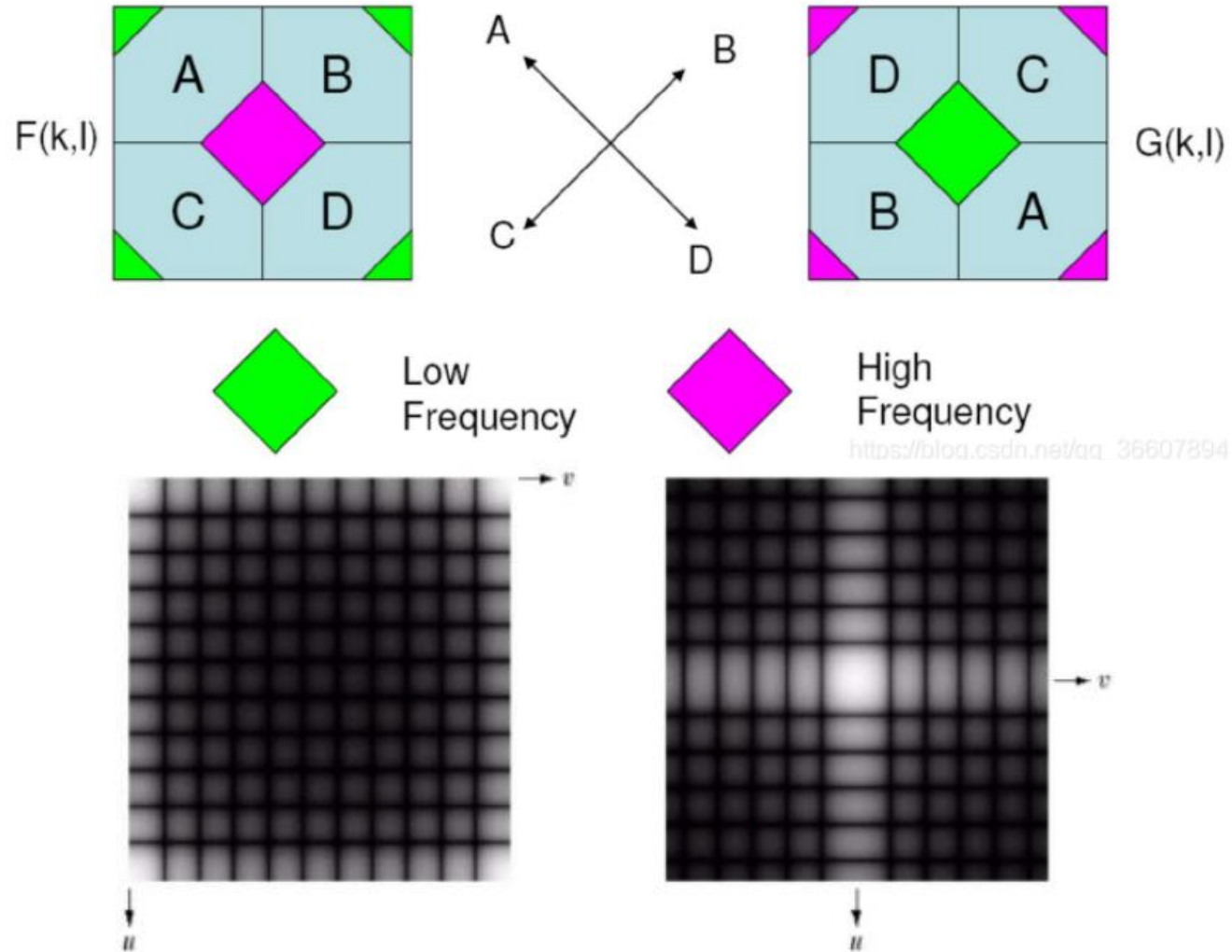
➤ 二维离散傅立叶变换的显示

- ✓ 为此，借助于傅里叶变换的周期性与频率位移性质，对此，通常对频域进行换位以使频谱分布符合上述要求——**图像中心化**



二维离散傅立叶变换结果中频率成分分布示意图

频谱图像中心化





频谱图像中心化

➤ 二维离散傅立叶变换的显示

✓ 使频域的中心位移 $u_0=v_0=N/2$:

$$f(x, y)(-1)^{x+y} \Leftrightarrow F\left(u - \frac{N}{2}, v - \frac{N}{2}\right)$$

✓ 相当于对原始图像 $f(x, y)$ 乘以 $(-1)^{m+n}$, 再进行傅里叶变换

$$F'(u, v) = F\{f(x, y)(-1)^{x+y}\}$$

✓ 对应于 $F'(u, v)$ 的反变换不等于 $f(x, y)$:

$$f(x, y) = F^{-1}\{F'(u, v)\} \times (-1)^{x+y}$$

证明: 可以通过对 $f(x, y)(-1)^{x+y}$ 作FT来实现, 即

$$f(x, y) \Leftrightarrow f(x, y)(-1)^{x+y}$$



频谱图像中心化

例：借助于傅里叶变换的周期性与频率位移性质，对频域进行换位以使频谱分布符合上述要求——图像中心化

证明

$$f(x, y)e^{j2\pi(u_0x+v_0y)/N} \Leftrightarrow F(u-u_0, v-v_0)$$

$$\text{令 } u_0 = v_0 = \frac{N}{2}$$

$$f(x, y)e^{j2\pi(\frac{N}{2}x+\frac{N}{2}y)/N} \Leftrightarrow F(u-\frac{N}{2}, v-\frac{N}{2})$$

$$f(x, y)e^{j\pi(x+y)} \Leftrightarrow F(u-\frac{N}{2}, v-\frac{N}{2})$$

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u-\frac{N}{2}, v-\frac{N}{2})$$

频谱图像中心化

$$f(x)e^{j2\pi(u_0x/M)} \Leftrightarrow F(u - u_0)$$

$$\mathcal{D}\left[f(x, y)(-1)^{x+y}\right] = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right)$$

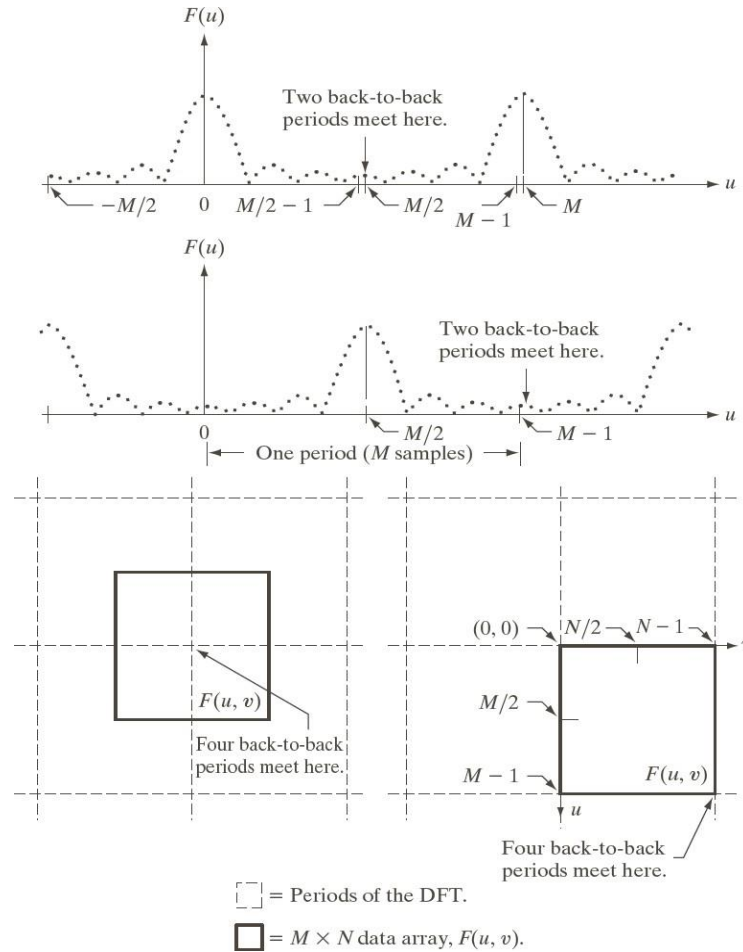
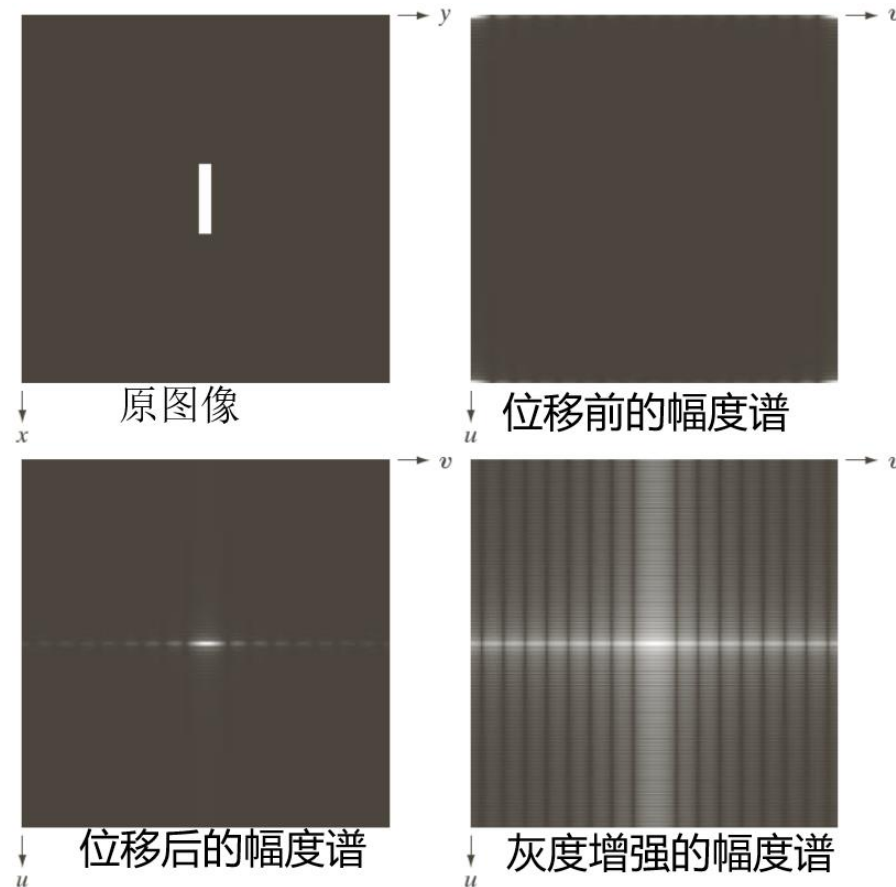


FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

数字图像的DFT变换

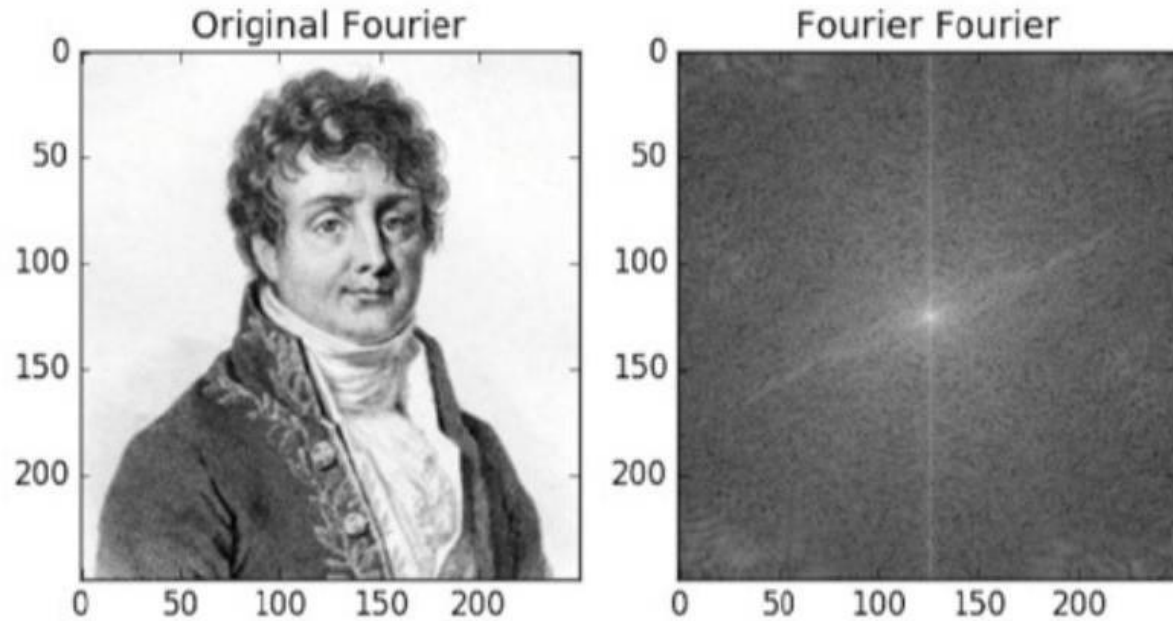
□ 傅里叶谱图的中心化平移和对数增强



a b
c d

FIGURE 4.24

(a) Image.
(b) Spectrum showing bright spots in the four corners.
(c) Centered spectrum.
(d) Result showing increased detail after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The coordinate convention used throughout the book places the origin of the spatial and frequency domains at the top left.

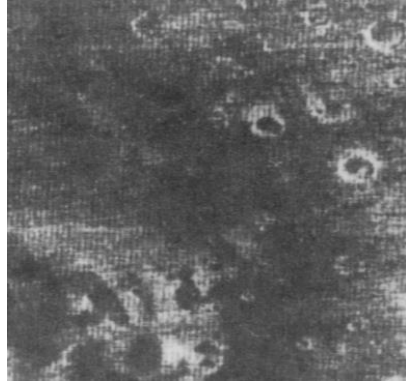


右边图就是频率分布图谱，其中

- 越靠近中心的位置频率越低，
- 越亮（灰度值越高）的位置代表该频率的信号振幅越大。

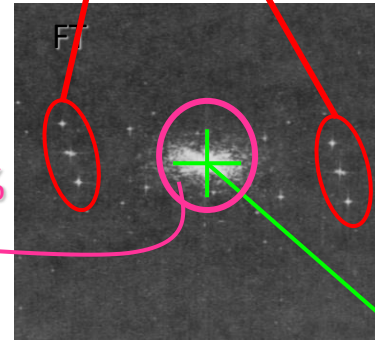
FFT的结果是复数形式，保留了图像的全部信息，但去绝对值得到的频谱图只表现了振幅而没有体现相位。

原图



突变：边缘、轮廓、噪音

原图

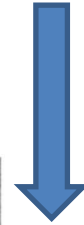
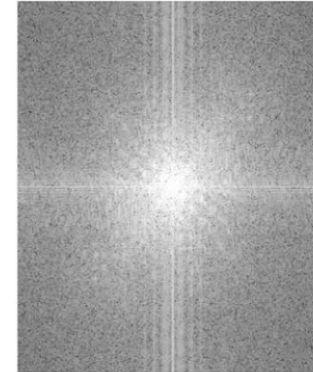


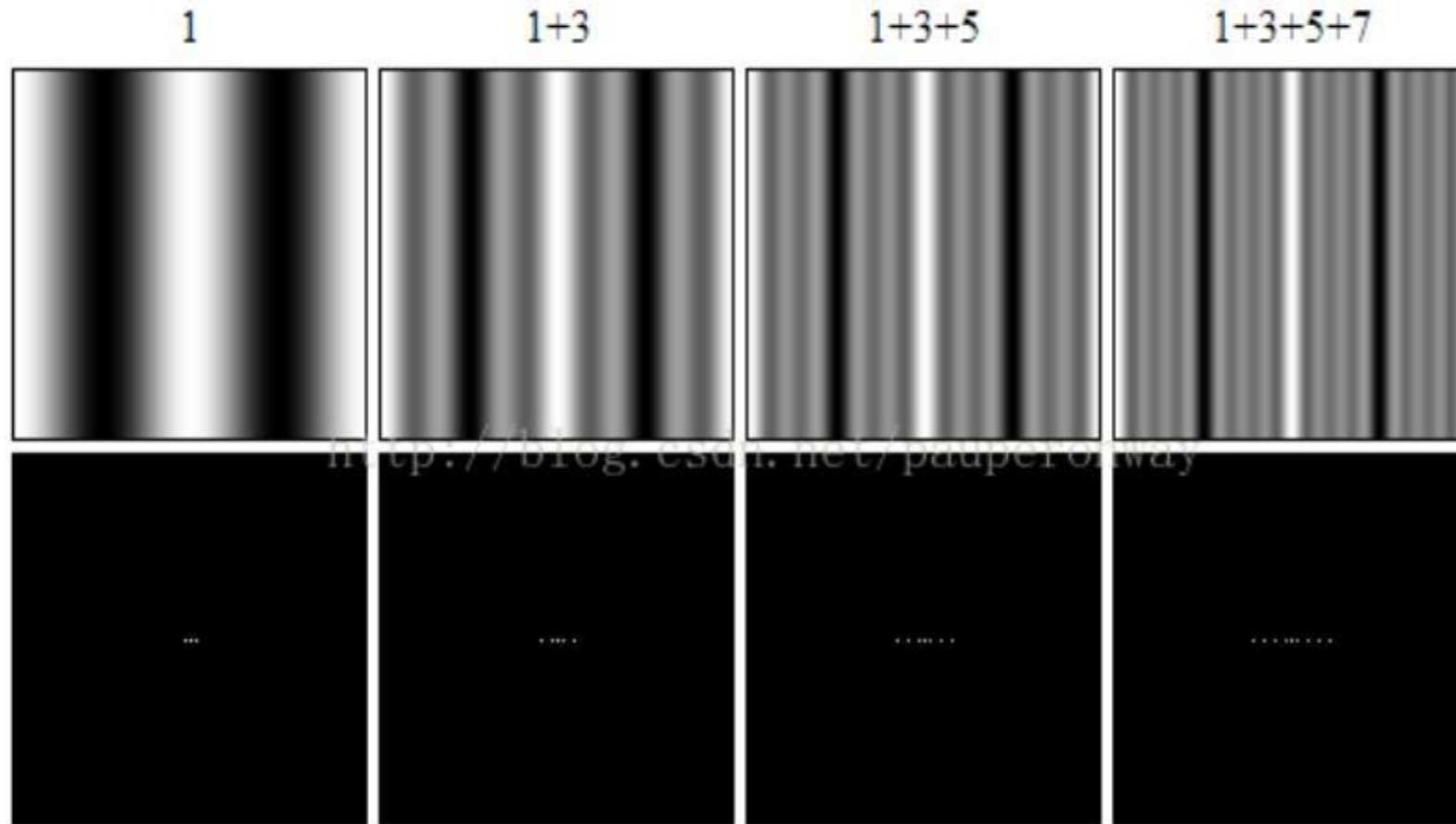
能量分布集中在低频区 85%

$F(0,0)$

图像平均亮度

灰度变化平缓 \Leftrightarrow 低频分量
 灰度变化突变 \Leftrightarrow 高频分量





数字图像的DFT变换

二维DFT及其反变换

下图实例：

中心化，矩形宽高化为 2 : 1

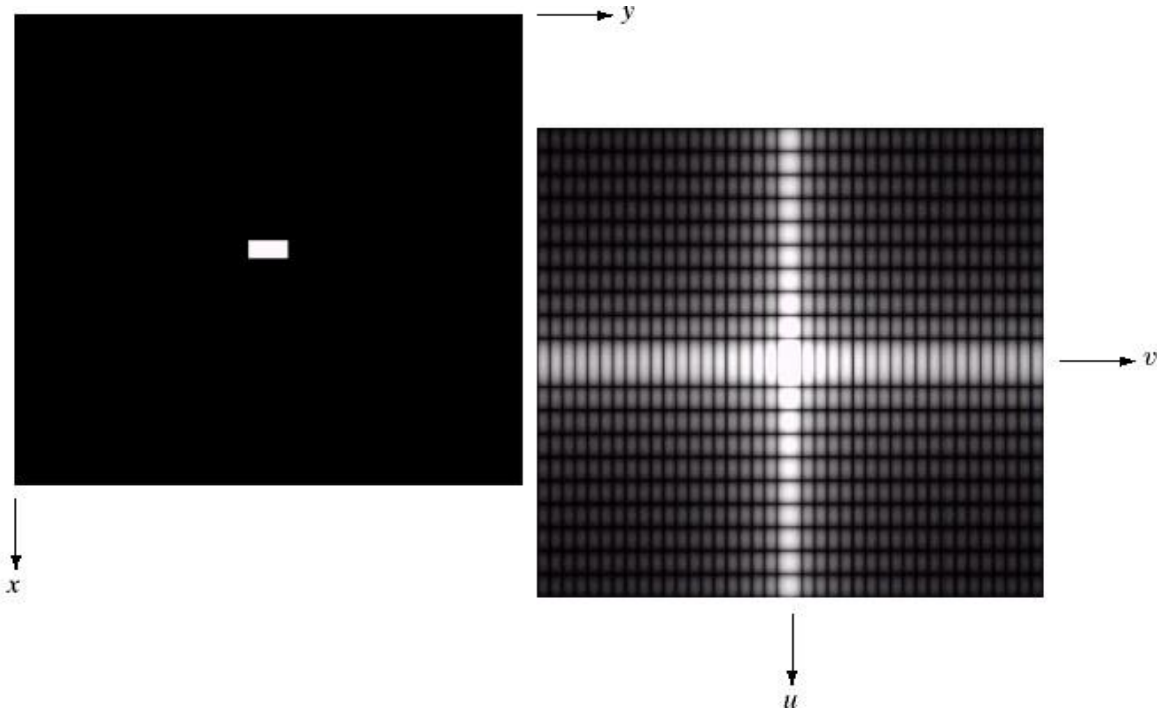
反映到频域轴亮点间距恰好相反

$$f(ax, by) \iff \frac{1}{ab} F\left(\frac{u}{a}, \frac{v}{b}\right)$$

a b

FIGURE 4.3

(a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels.
 (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

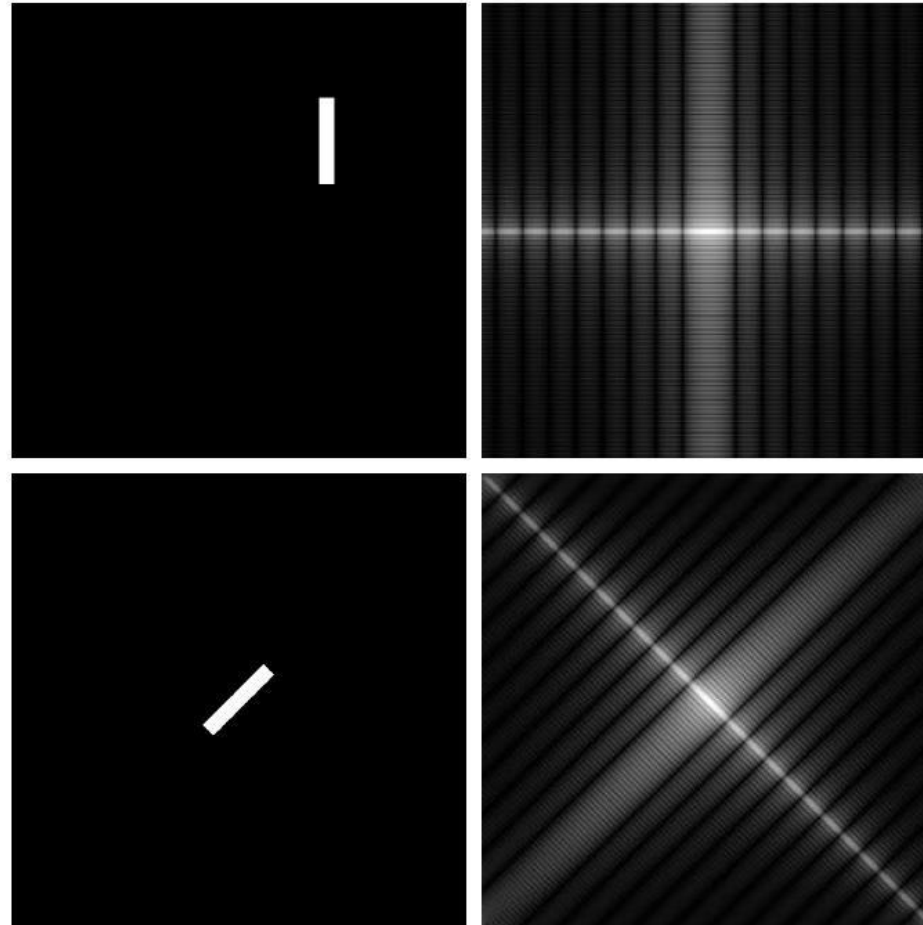


数字图像的DFT变换

↪ 旋转特性

a b
c d

FIGURE 4.25
(a) The rectangle in Fig. 4.24(a) translated, and (b) the corresponding spectrum. (c) Rotated rectangle, and (d) the corresponding spectrum. The spectrum corresponding to the translated rectangle is identical to the spectrum corresponding to the original image in Fig. 4.24(a).



空域图像旋转角度
对应于频域DFT函数
旋转相同角度。

$$f(x - a, y - b)$$



$$e^{-j2\pi(au+bv)} F(u, v)$$

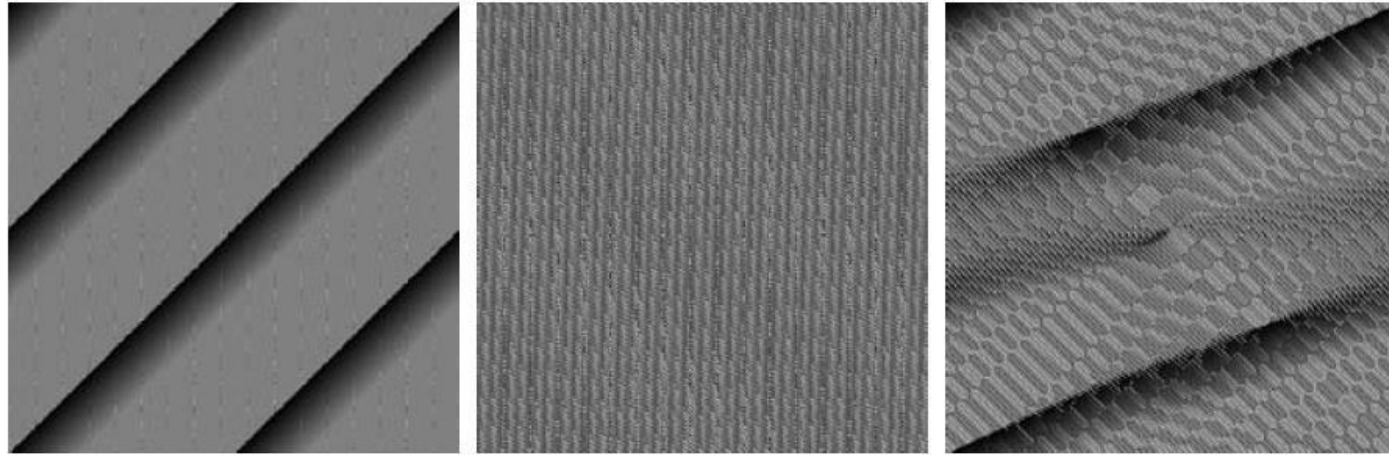
$$f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$$



$$F(u \cos \theta + v \sin \theta, -u \sin \theta + v \cos \theta)$$

数字图像的DFT变换

- 图像发生平移或旋转变换后，对应相角发生变化
 - 缺少直观的类比性
 - 相角携带了反映图像中物体空间位置的信息

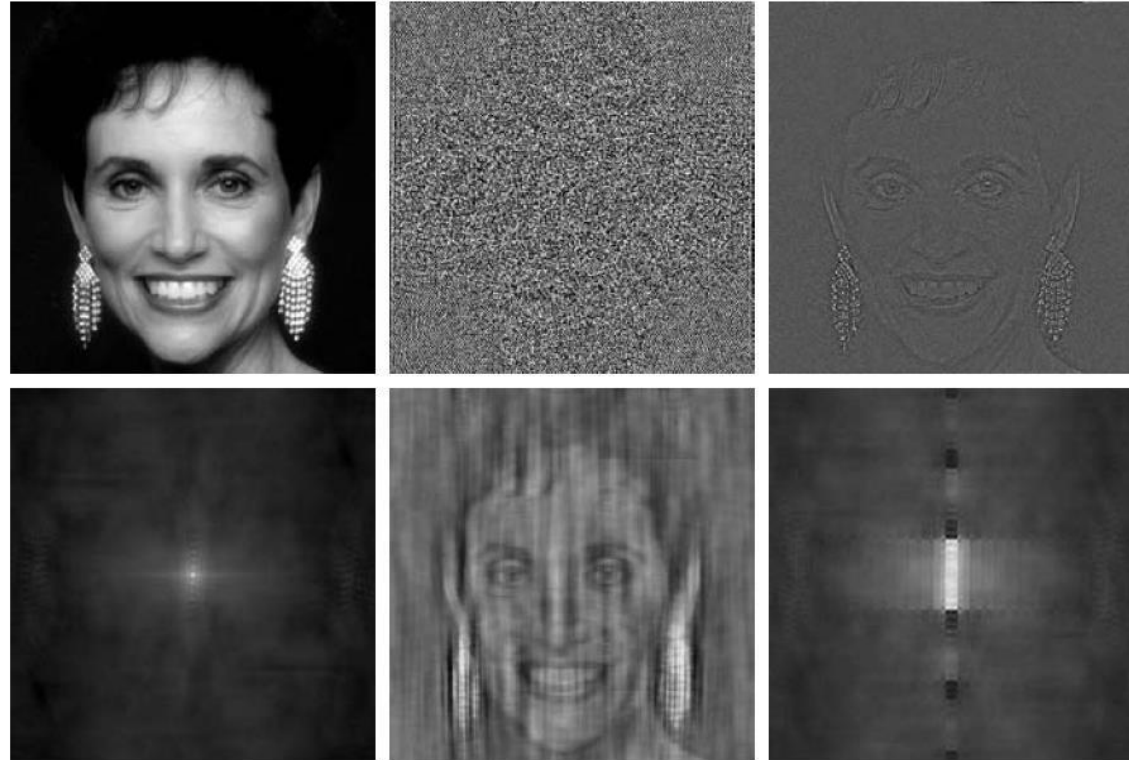


a b c

FIGURE 4.26 Phase angle array corresponding (a) to the image of the centered rectangle in Fig. 4.24(a), (b) to the translated image in Fig. 4.25(a), and (c) to the rotated image in Fig. 4.25(c).

基于二维DFT频谱与相位的反变换

- 仅用相角信息，可恢复原图的**关键形状特性**
- 仅用谱信息，直流项占支配地位，难以恢复原图的形状信息



a	b	c
d	e	f

FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.

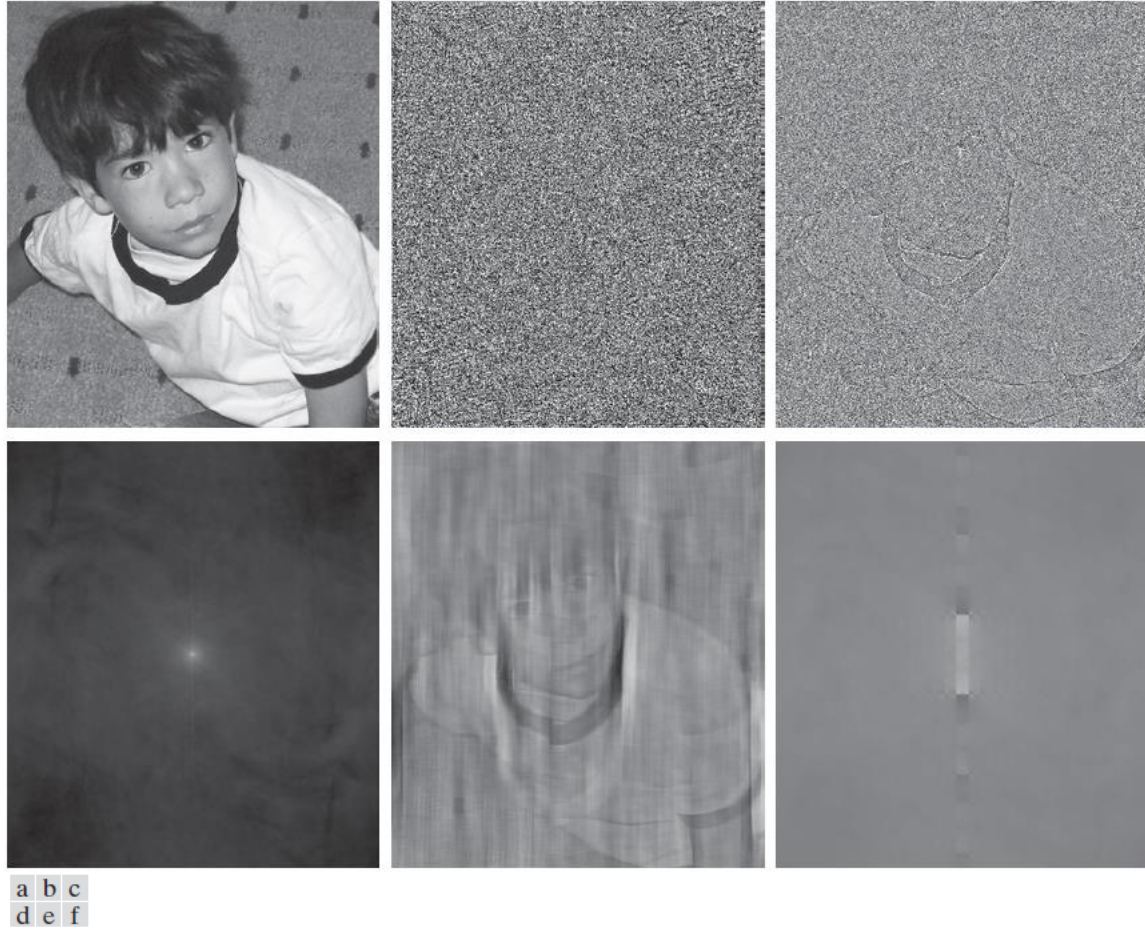
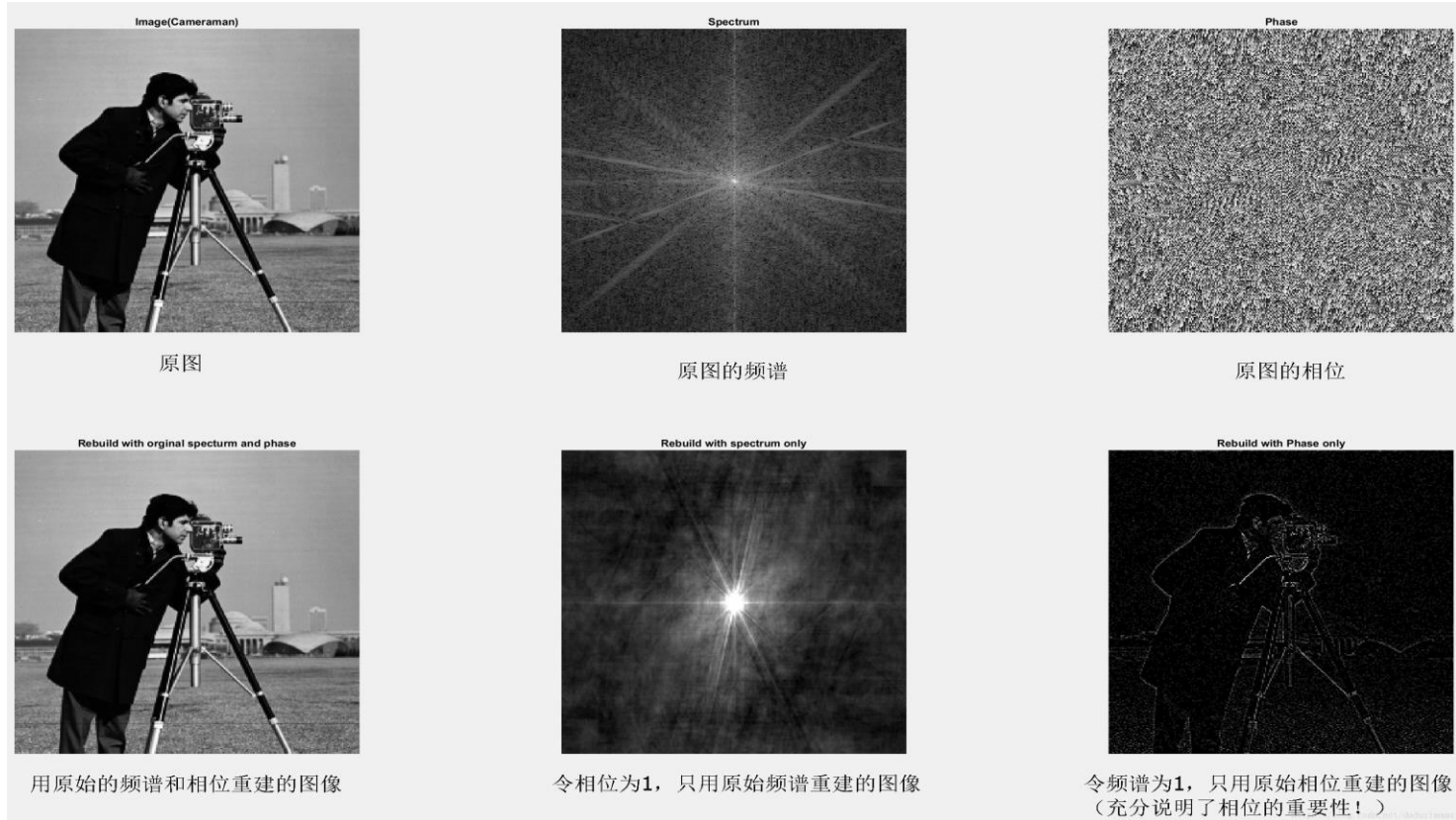
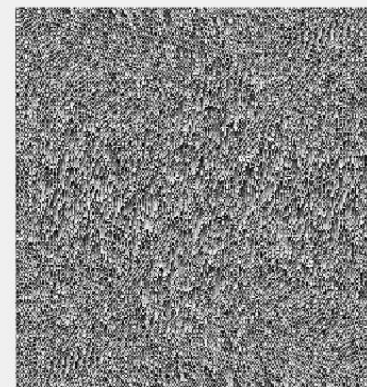
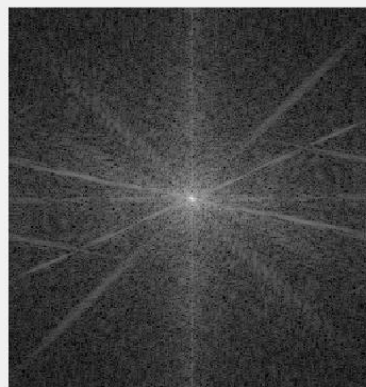


FIGURE 4.26 (a) Boy image. (b) Phase angle. (c) Boy image reconstructed using only its phase angle (all shape features are there, but the intensity information is missing because the spectrum was not used in the reconstruction). (d) Boy image reconstructed using only its spectrum. (e) Boy image reconstructed using its phase angle and the spectrum of the rectangle in Fig. 4.23(a). (f) Rectangle image reconstructed using its phase and the spectrum of the boy's image.

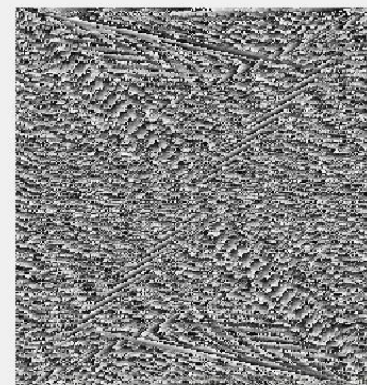
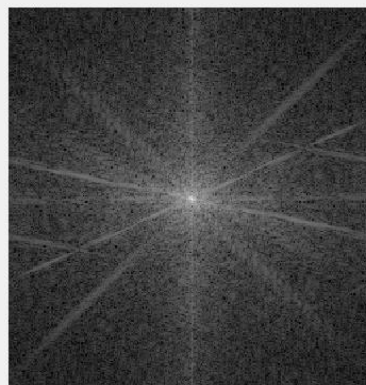


- 图像的频域分析和滤波中，相位是常常被忽略的。
- 虽然相位分量的贡献很不直观，但是它恰恰很重要。
- 相位是频谱中各正弦分量关于原点的位移的度量。





对调图像四个象限，图像的频谱不变，但图像的相位发生了改变！

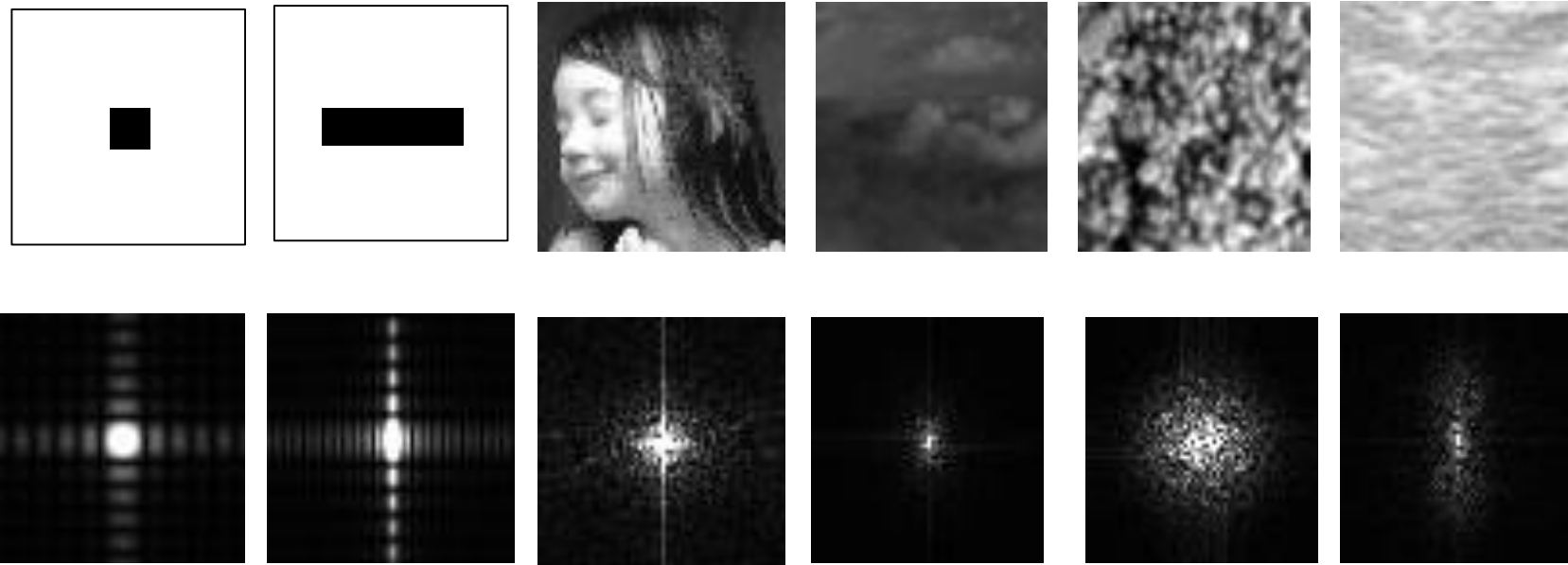




中间最小的那个圆圈内包含了大约85%的能量
中间那个圈包含了大约93%的能量
最外面那个圈则包含了几乎99%的能量

数字图像的DFT变换

典型图象的频谱



空域卷积 vs. DFT

$$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$$

$$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$$

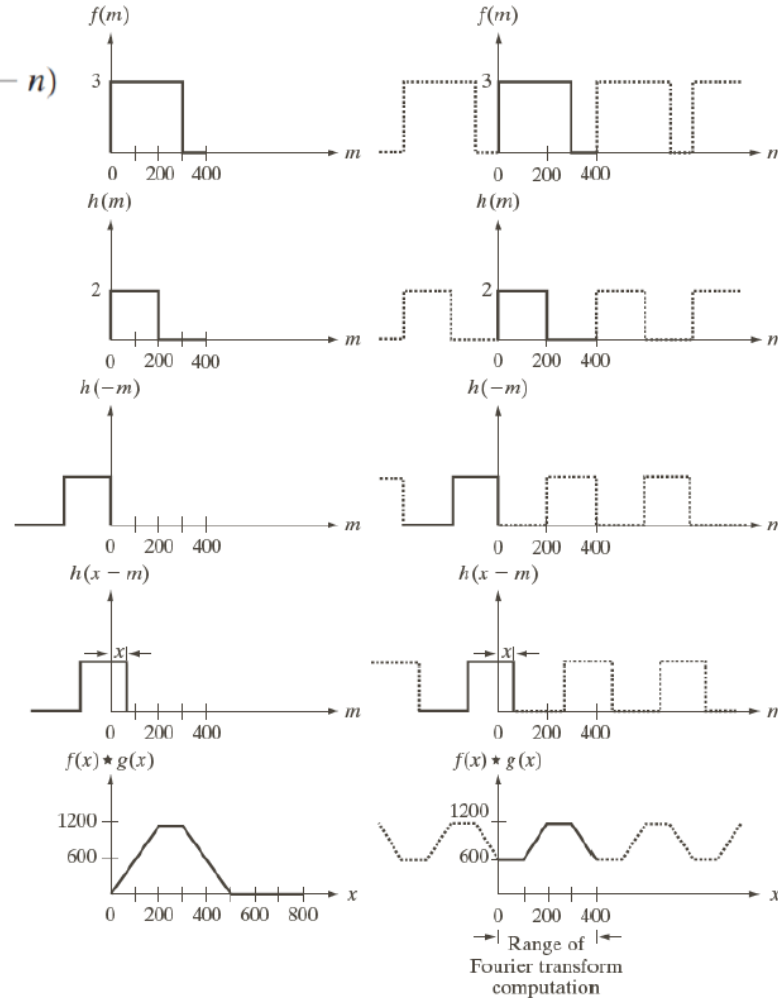
$$f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$$

$$f(x) \star h(x) = \sum_{m=0}^{399} f(x)h(x - m)$$

解决缠绕错误：添加0补齐

$$P \geq A + B - 1$$

$$P=400+400-1=799$$



a	f
b	g
c	h
d	i
e	j

FIGURE 4.28 Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.



DFT定义及相关表达式小结

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$
3) Polar representation	$F(u, v) = F(u, v) e^{j\phi(u,v)}$
4) Spectrum	$ F(u, v) = [R^2(u, v) + I^2(u, v)]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = F(u, v) ^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

(Continued)

- 直流项决定了图像的平均灰度，因此将它设置为零时，输出图像的平均灰度将为0



DFT定义及相关表达式小结

8) Periodicity (k_1 and k_2 are integers)

$$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N) \\ = F(u + k_1M, v + k_2N)$$

$$f(x, y) = f(x + k_1M, y) = f(x, y + k_2N) \\ = f(x + k_1M, y + k_2N)$$

9) Convolution

$$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$$

10) Correlation

$$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)$$

11) Separability

The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.

12) Obtaining the inverse Fourier transform using a forward transform algorithm.

$$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)e^{-j2\pi(ux/M+vy/N)}$$

This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$. Taking the complex conjugate and dividing by MN gives the desired inverse. See Section 4.11.2.



第4章 频率域滤波

1. 离散傅立叶变换(DFT)
2. 频率域滤波基础
3. 频率域滤波器平滑图像
4. 频率域滤波器锐化图像
5. 选择性滤波
6. 其它变换(离散余弦变换)
7. 小波变换

4.2 频率域滤波基础

频率域滤波基本步骤:

- 1、 $(-1)^{x+y} \times$ 原图像
- 2、 $F(u, v)$
- 3、 $H(u, v) \times F(u, v)$
- 4、 反DFT
- 5、 实部
- 6、 用 $(-1)^{x+y} \times$ (5) 结果。

被滤波图像 = $\mathcal{F}^{-1} [G(u, v)]$

$$G(u, v) = H(u, v)F(u, v)$$

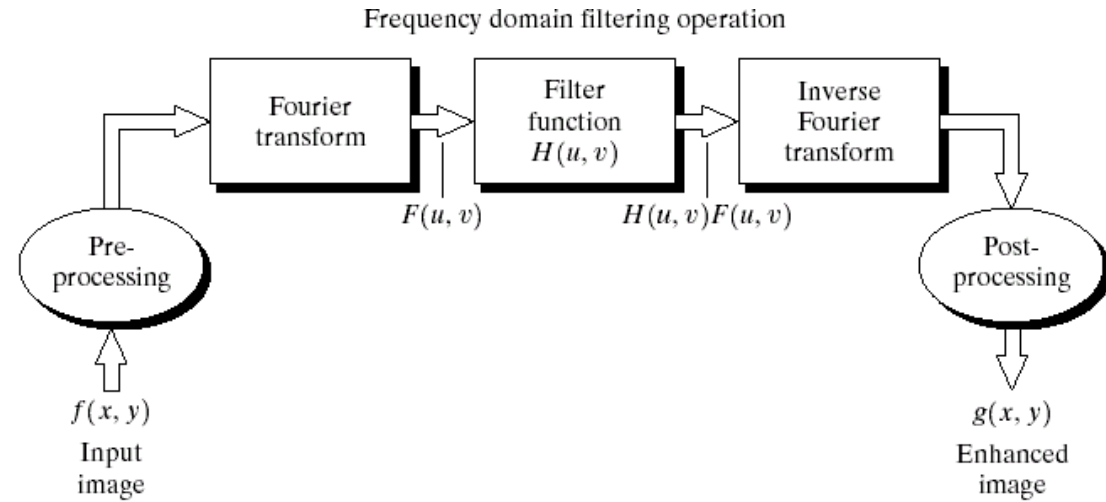


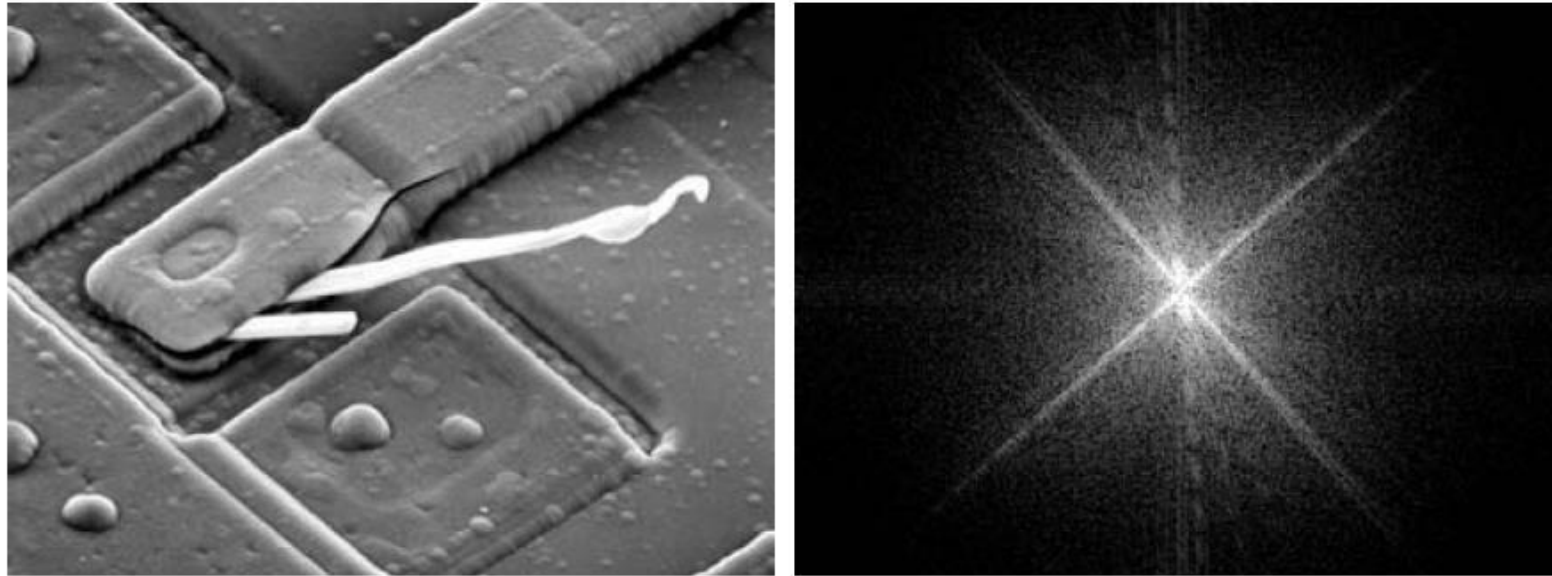
FIGURE 4.5 Basic steps for filtering in the frequency domain.

$$g(x, y) = \text{Real} \left\{ \mathcal{F}^{-1} [H(u, v)F(u, v)] \right\}$$

- IDFT操作应该得出实函数，然而计算精度不足通常会使得IDFT中出现寄生复数项，因此在结果中取实部来解决

图像空域与频域的定性分析

□ 集成电路SEM图像及其傅里叶谱



a b

FIGURE 4.29 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

4.2 频率域滤波基础

□ 一些基本的滤波器及其性质

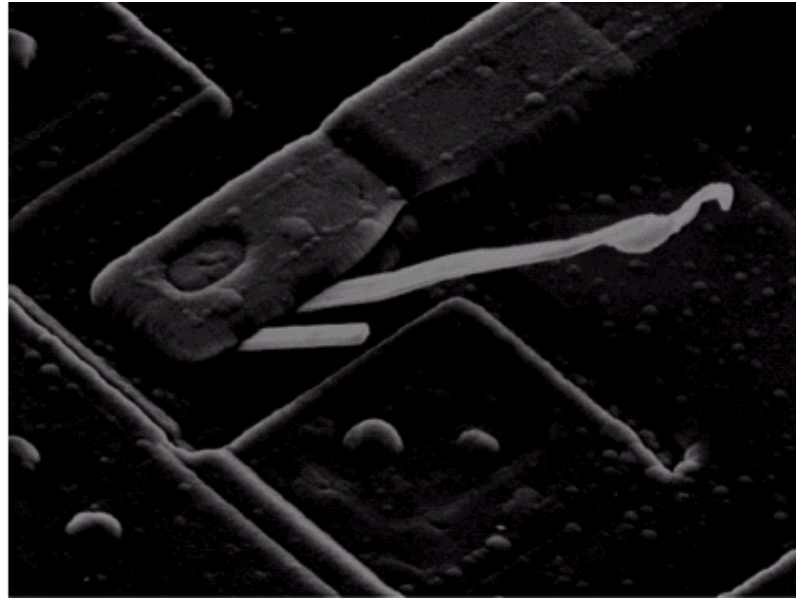
陷波滤波器: $F(0,0) \rightarrow$ 平均值

如果想使其平均灰度为0, 则可:

$$H(u,v) = \begin{cases} 0 & (u,v) = (\frac{M}{2}, \frac{N}{2}) \\ 1 & \text{其它} \end{cases}$$

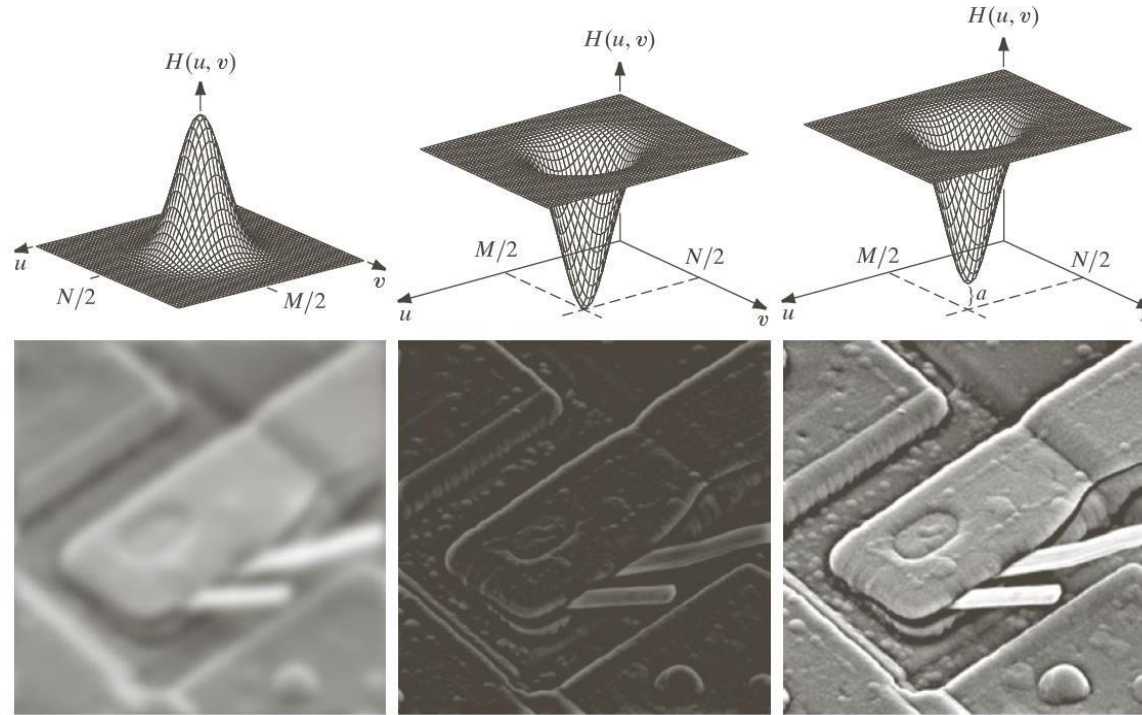
具体见如下图示 (存在负灰度, 但该图中所有负值被置为零):

FIGURE 4.6
Result of filtering
the image in
Fig. 4.4(a) with a
notch filter that
set to 0 the
 $F(0,0)$ term in
the Fourier
transform.



4.2 频率域滤波基础

□ 低通滤波与高通滤波



a	b	c
d	e	f

FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

使用DFT时二维图像固有的周期性

□ 边缘填充对滤波结果的影响



a b c

FIGURE 4.32 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with padding. Compare the light area of the vertical edges in (b) and (c).

SUMMARY OF STEPS FOR FILTERING IN THE FREQUENCY DOMAIN

The process of filtering in the frequency domain can be summarized as follows:

1. Given an input image $f(x, y)$ of size $M \times N$, obtain the padding sizes P and Q using Eqs. (4-102) and (4-103); that is, $P = 2M$ and $Q = 2N$.
2. Form a padded[†] image $f_p(x, y)$ of size $P \times Q$ using zero-, mirror-, or replicate padding (see Fig. 3.39 for a comparison of padding methods).
3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center the Fourier transform on the $P \times Q$ frequency rectangle.
4. Compute the DFT, $F(u, v)$, of the image from Step 3.
5. Construct a real, symmetric filter transfer function, $H(u, v)$, of size $P \times Q$ with center at $(P/2, Q/2)$.
6. Form the product $G(u, v) = H(u, v)F(u, v)$ using elementwise multiplication; that is, $G(i, k) = H(i, k)F(i, k)$ for $i = 0, 1, 2, \dots, M - 1$ and $k = 0, 1, 2, \dots, N - 1$.
7. Obtain the filtered image (of size $P \times Q$) by computing the IDFT of $G(u, v)$:

$$g_p(x, y) = \left(\text{real} \left[\mathcal{F}^{-1} \{ G(u, v) \} \right] \right) (-1)^{x+y}$$

8. Obtain the final filtered result, $g(x, y)$, of the same size as the input image, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x, y)$.

a	b	c
d	e	f
g	h	

FIGURE 4.35

(a) An $M \times N$ image, f .

(b) Padded image, f_p of size $P \times Q$.

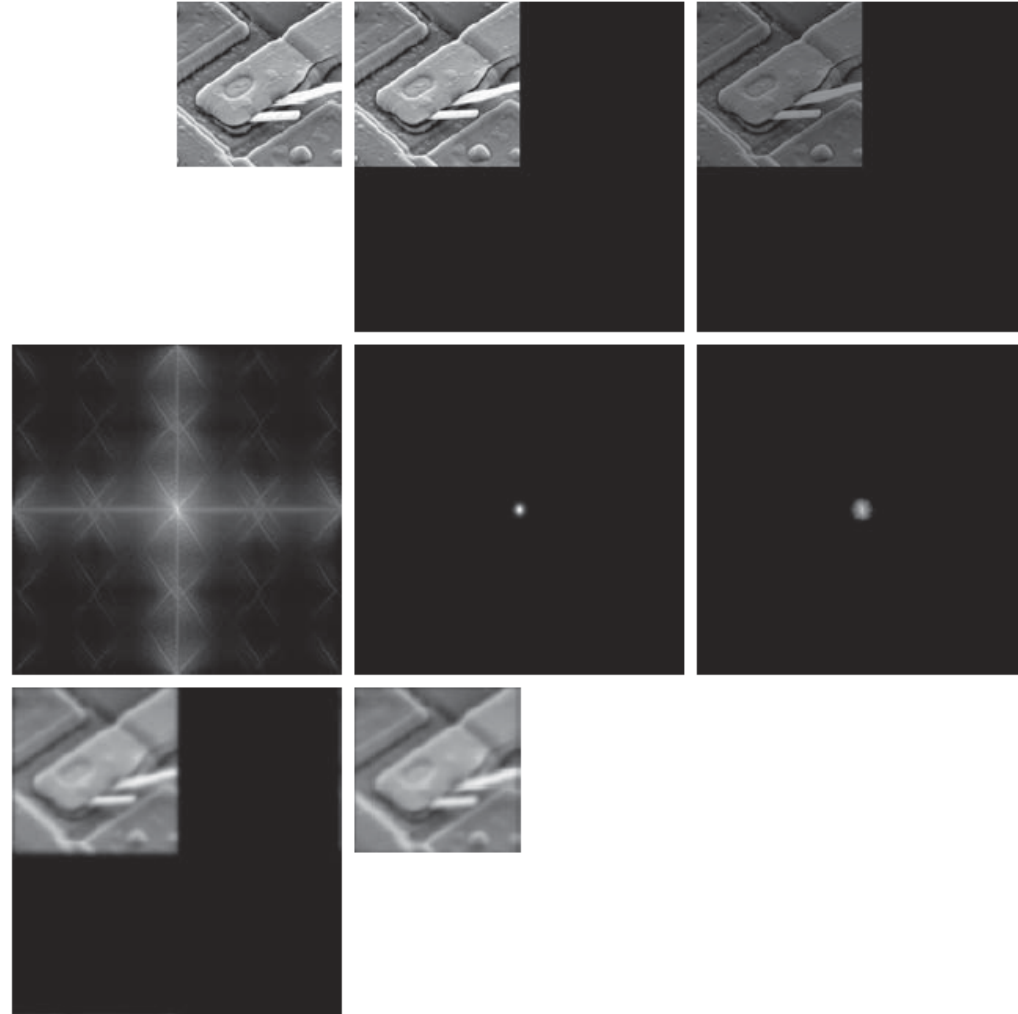
(c) Result of multiplying f_p by $(-1)^{x+y}$.

(d) Spectrum of F . (e) Centered Gaussian lowpass filter transfer function, H , of size $P \times Q$.

(f) Spectrum of the product HF .

(g) Image g_p , the real part of the IDFT of HF , multiplied by $(-1)^{x+y}$.

(h) Final result, g , obtained by extracting the first M rows and N columns of g_p .



空间和频率域滤波间的对应

空间域滤波和频率域滤波之间的对应关系

例如：高斯滤波函数

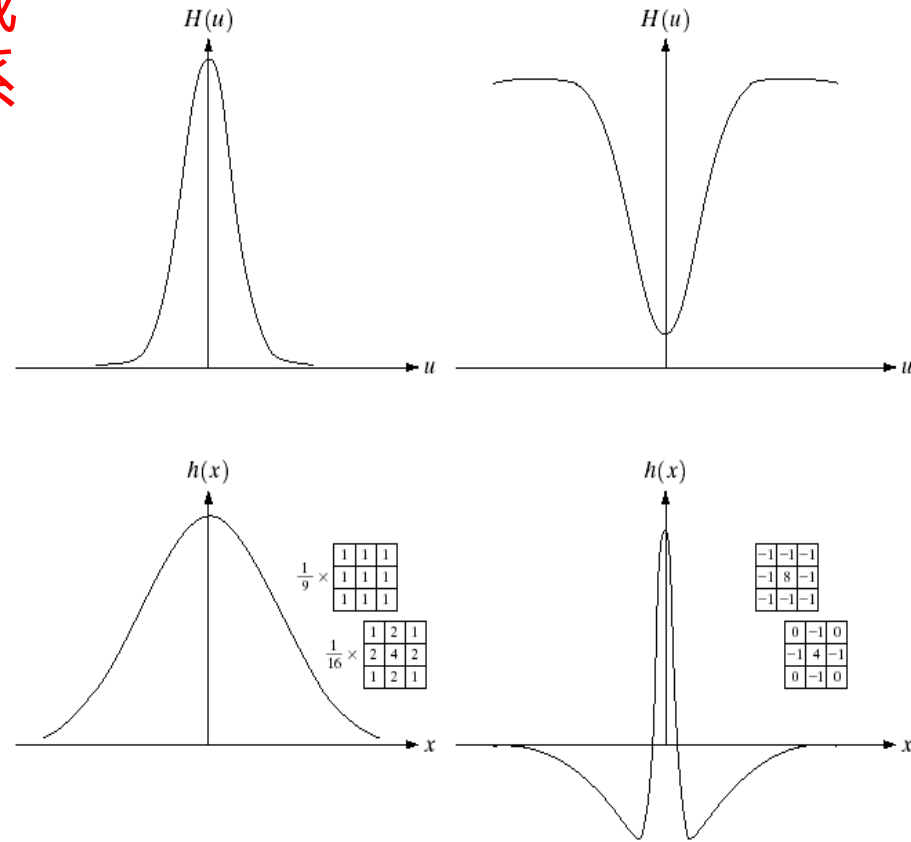
$$H(u) = Ae^{-u^2/2\sigma^2}$$

σ 为标准差：

$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

傅立叶变换及其反变换
因为实数。

曲线形状如右图所示：



a b
c d

FIGURE 4.9
(a) Gaussian frequency domain lowpass filter.
(b) Gaussian frequency domain highpass filter.
(c) Corresponding lowpass spatial filter.
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.



第4章 频率域滤波

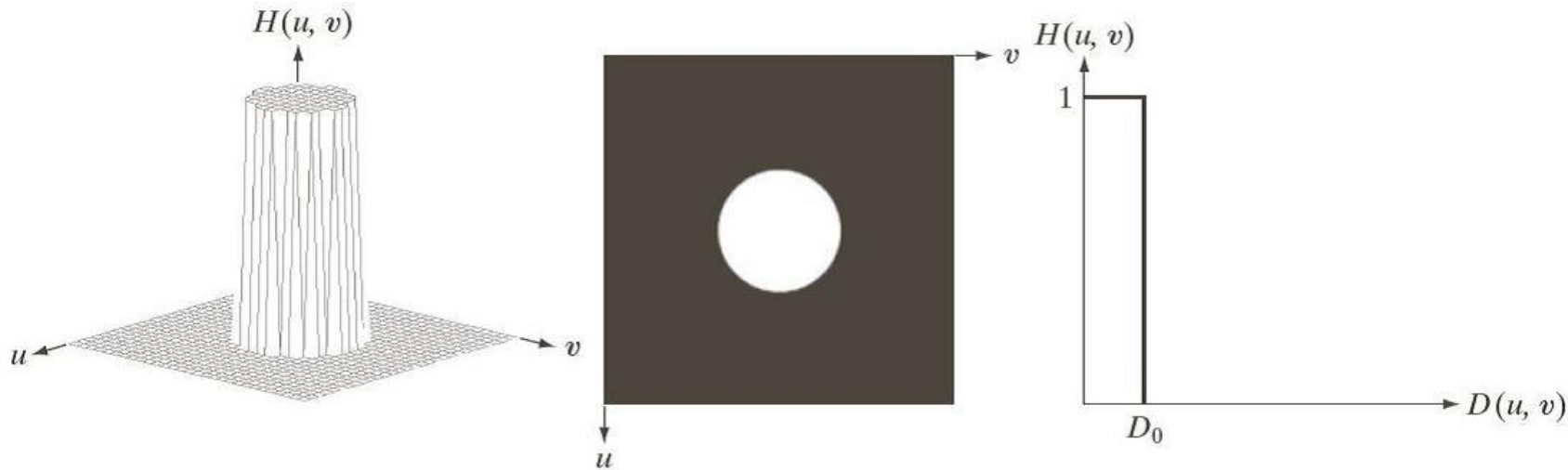
1. 离散傅立叶变换(DFT)
2. 频率域滤波基础
3. 频率域滤波器平滑图像
4. 频率域滤波器锐化图像
5. 选择性滤波
6. 其它变换(离散余弦变换)
7. 小波变换

4.3 频率域滤波平滑

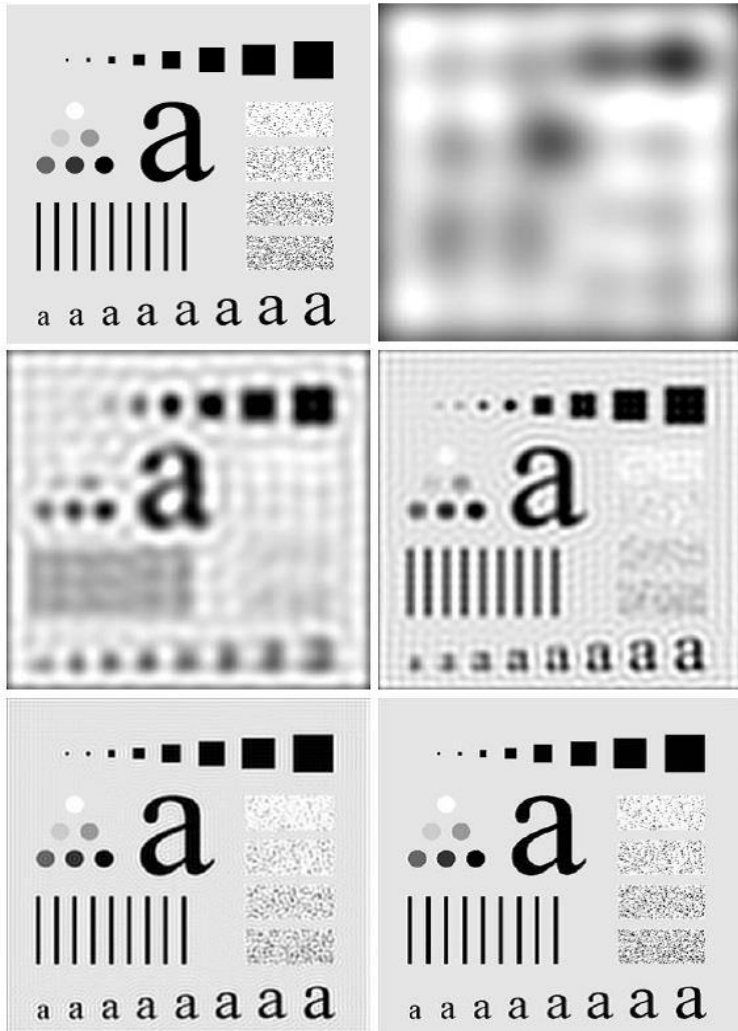
理想低通滤波器

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

其中， $D(u, v)$ 是频率域中点 (u, v) 与频率矩形中心的距离



理想低通滤波器示例



a	b
c	d
e	f

(a) 原图

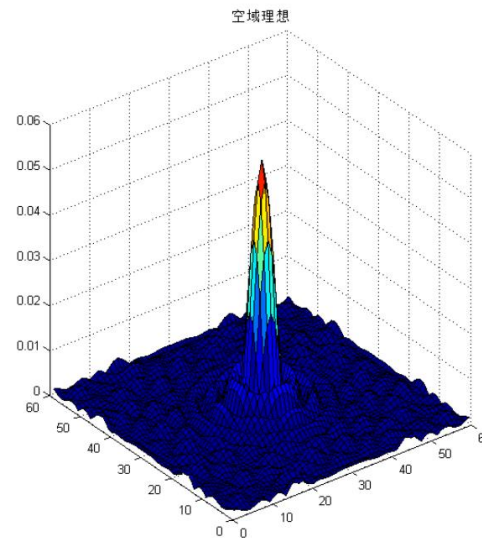
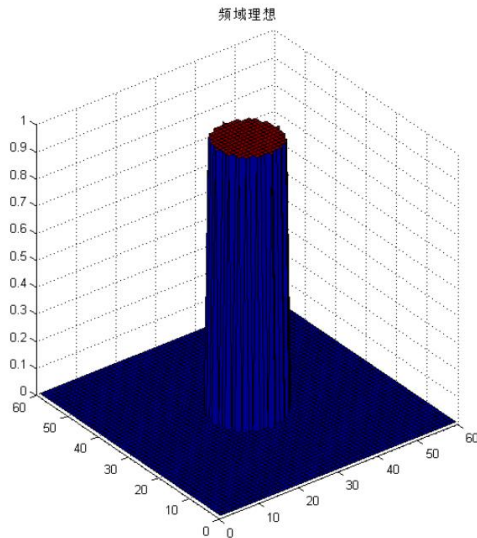
(b)-(f) 使用理想低通滤波器，截止频率设置10,30,60,160和460。这些滤波器移除的功率分别为总功率的13%，6.9%，4.3%，2.2%和0.8%。

振铃效应解释



a b

FIGURE 4.43
 (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000×1000 .
 (b) Intensity profile of a horizontal line passing through the center of the image.

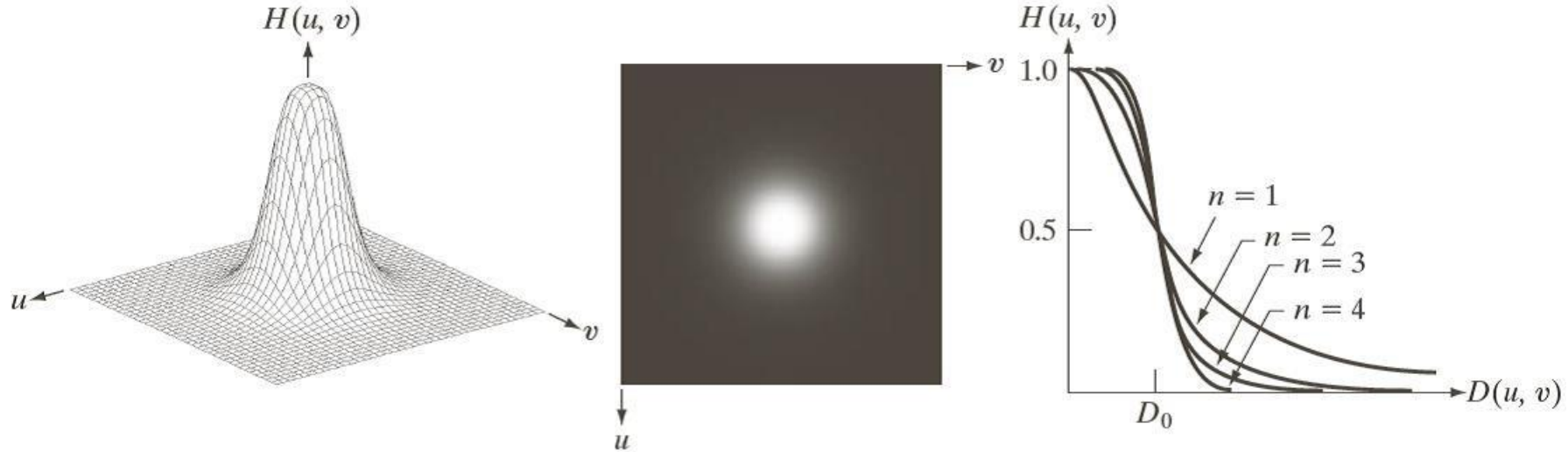


若频域滤波函数具有陡峭变化，则傅里叶逆变换得到的空域滤波函数会在外围出现震荡

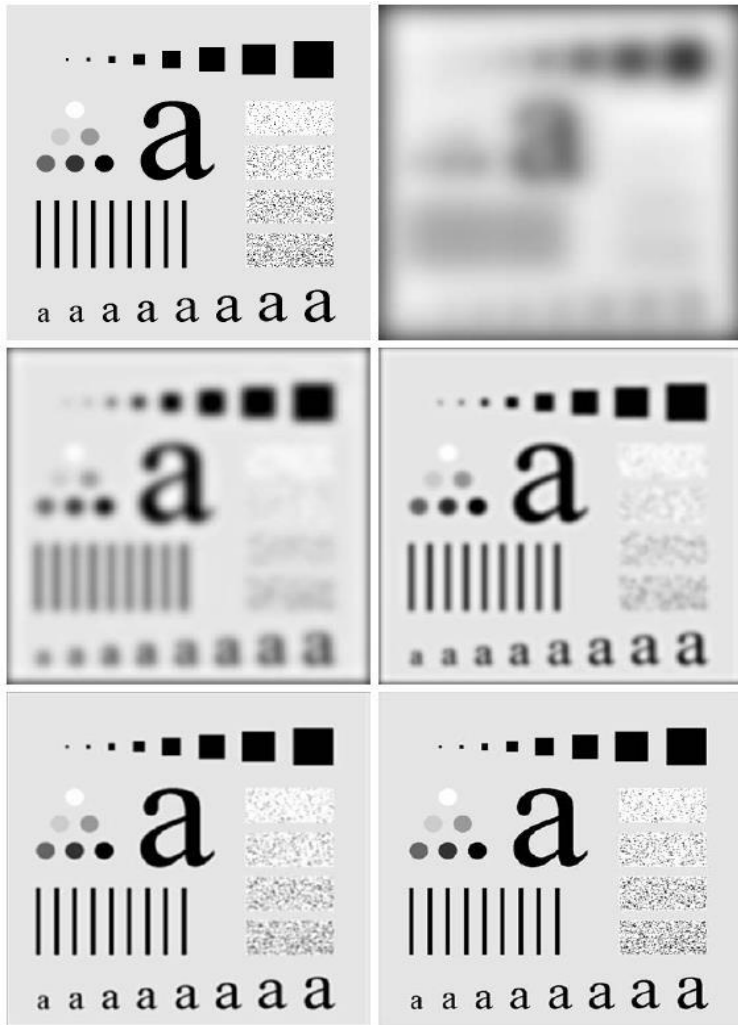
4.3 频率域滤波平滑

n 阶布特沃斯低通滤波器

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$



布特沃斯滤波器示例

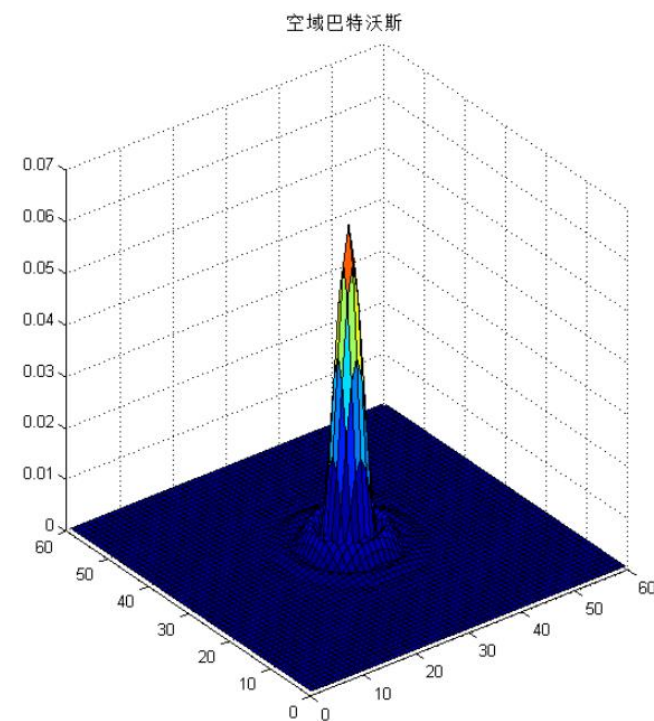
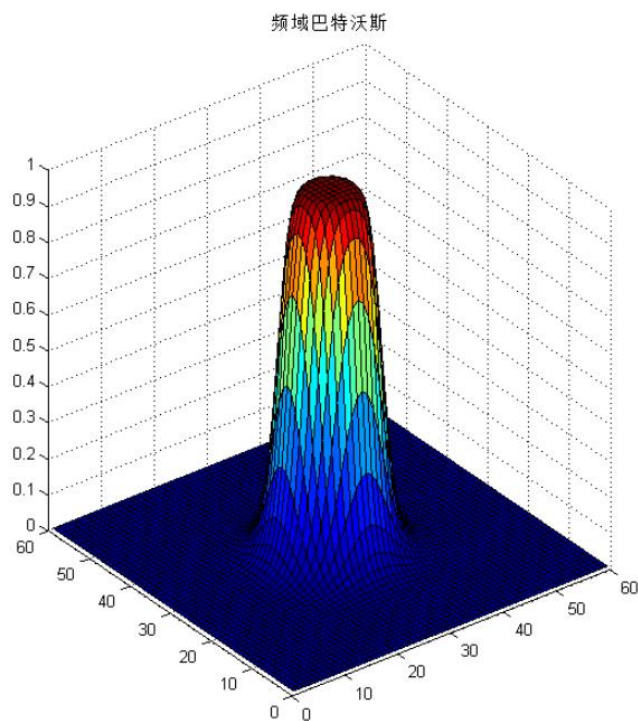


a	b
c	d
e	f

(a) 原图

(b)-(f) 使用二阶布特沃斯低通滤波器的结果，截止频率仍为10, 30, 60, 160和460。

1阶巴特沃斯没有“振铃“，随着阶数增大，振铃现象越发明显。下图取 $n=2$ ，可以看出空域函数外围部分出现震荡



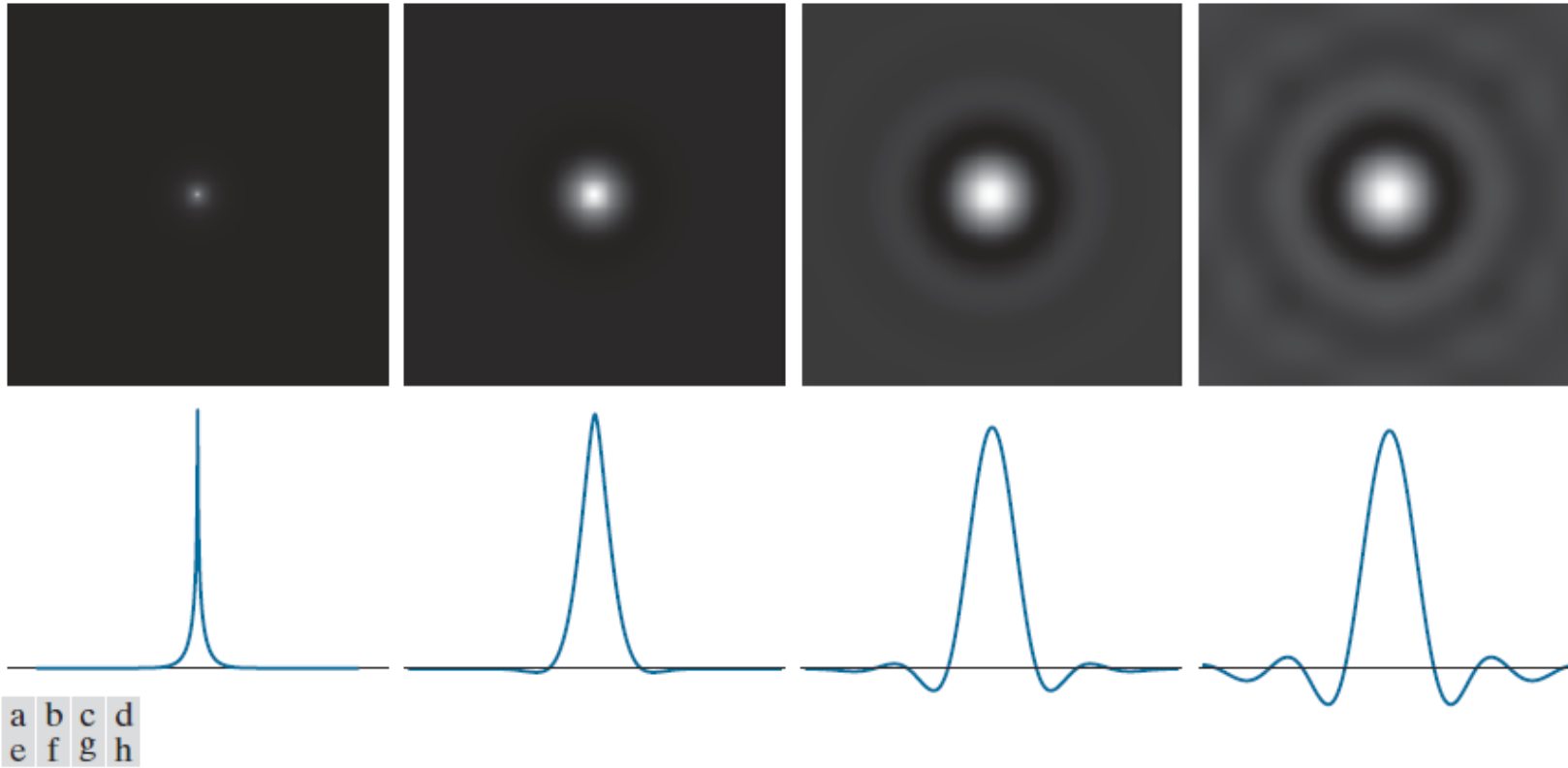
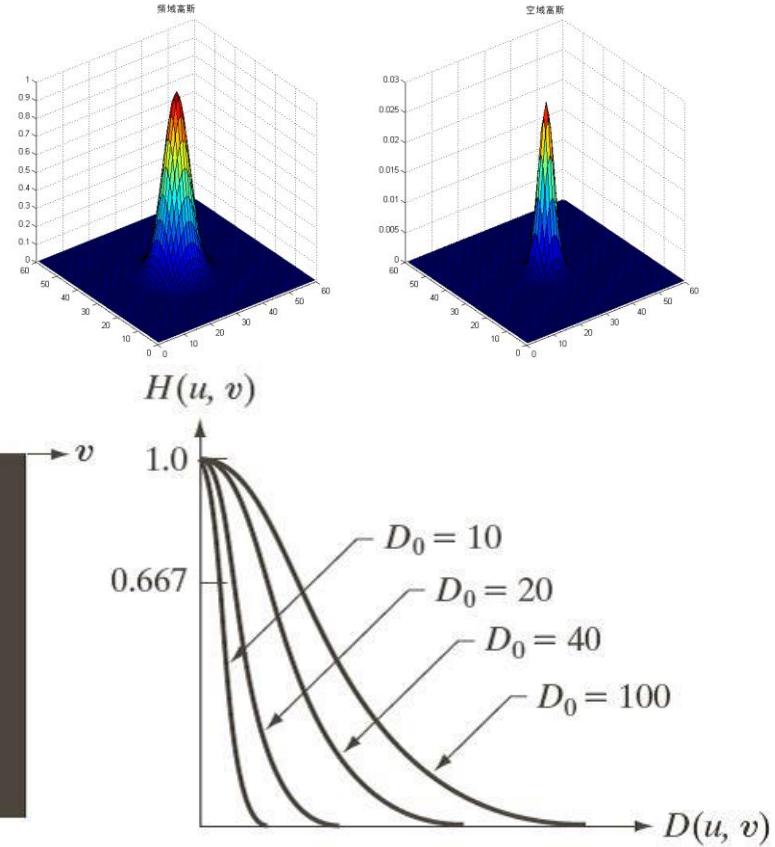
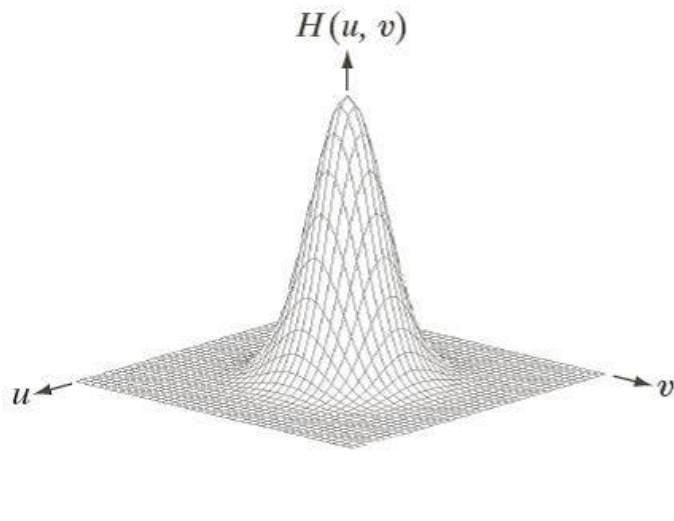


FIGURE 4.47 (a)–(d) Spatial representations (i.e., spatial kernels) corresponding to BLPF transfer functions of size 1000×1000 pixels, cut-off frequency of 5, and order 1, 2, 5, and 20, respectively. (e)–(h) Corresponding intensity profiles through the center of the filter functions.

4.3 频率域滤波平滑

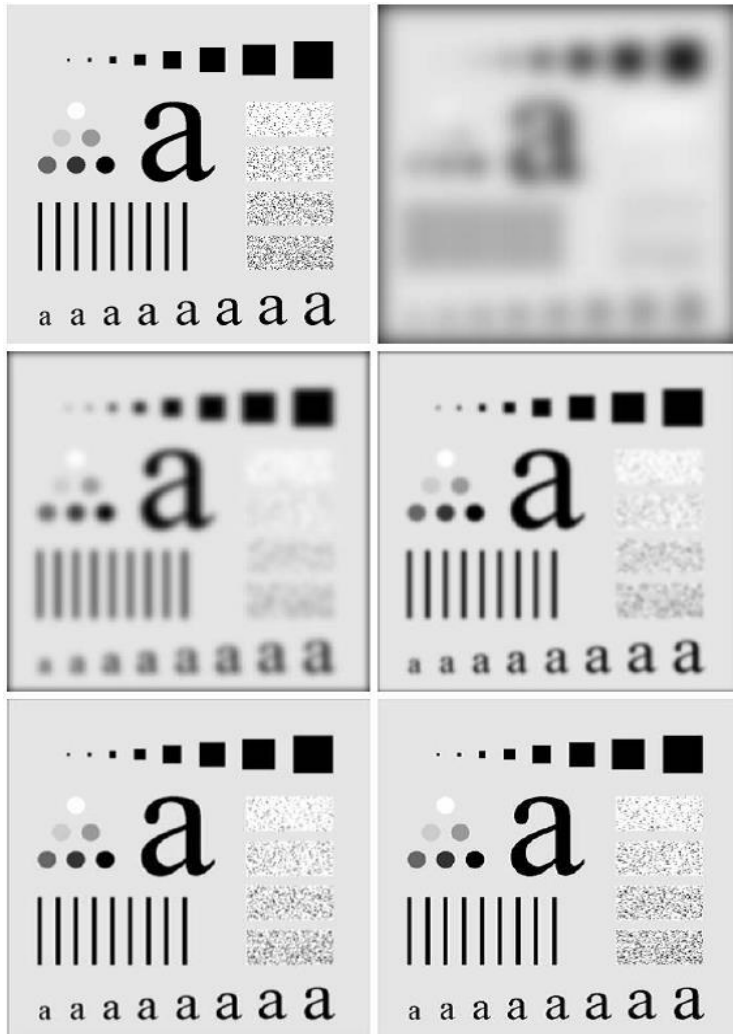
高斯低通滤波器

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$



高斯低通滤波器 (GLPF) 的傅里叶反变换也是高斯的, 因此通过上式的IDFT得到的空间高斯滤波器没有振铃。

高斯低通滤波器示例



a b
c d
e f

(a) 原图

(b)-(f) 使用高斯低通滤波器的结果，截止频率仍为10，30，60，160和460。

低通滤波的其他例子

1. 字符识别:

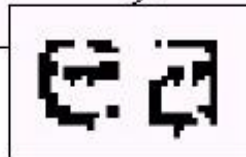
下图: 断裂现象

a b

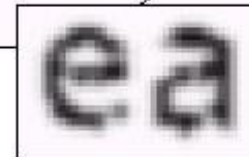
FIGURE 4.19

(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



低通滤波的其他例子

2. 印刷和出版业：预处理

下图：减少面部细纹



a b c

FIGURE 4.20 (a) Original image (1028×732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

低通滤波的其他例子

3. 卫星和航空图像:

下图：墨西哥湾和佛罗里达图像存在“扫描线”
(用高斯低通来处理)



a b c

FIGURE 4.21 (a) Image showing prominent scan lines. (b) Result of using a GLPF with $D_0 = 30$. (c) Result of using a GLPF with $D_0 = 10$. (Original image courtesy of NOAA.)



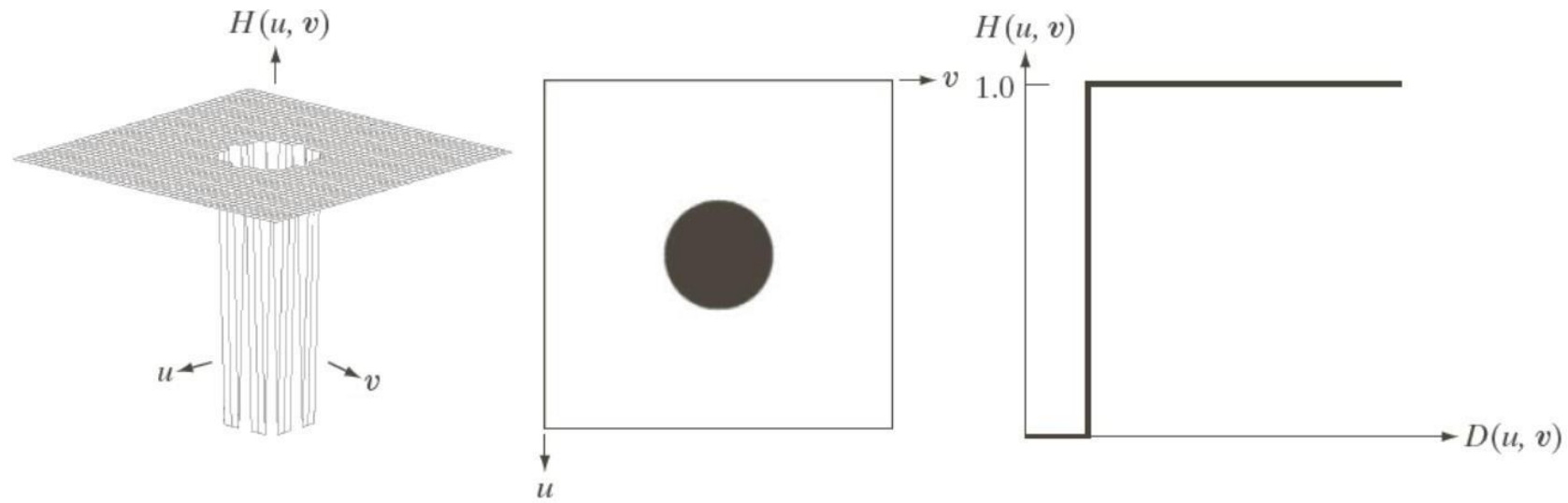
第4章 频率域滤波

1. 离散傅立叶变换(DFT)
2. 频率域滤波基础
3. 频率域滤波器平滑图像
4. 频率域滤波器锐化图像
5. 选择性滤波
6. 其它变换(离散余弦变换)
7. 小波变换

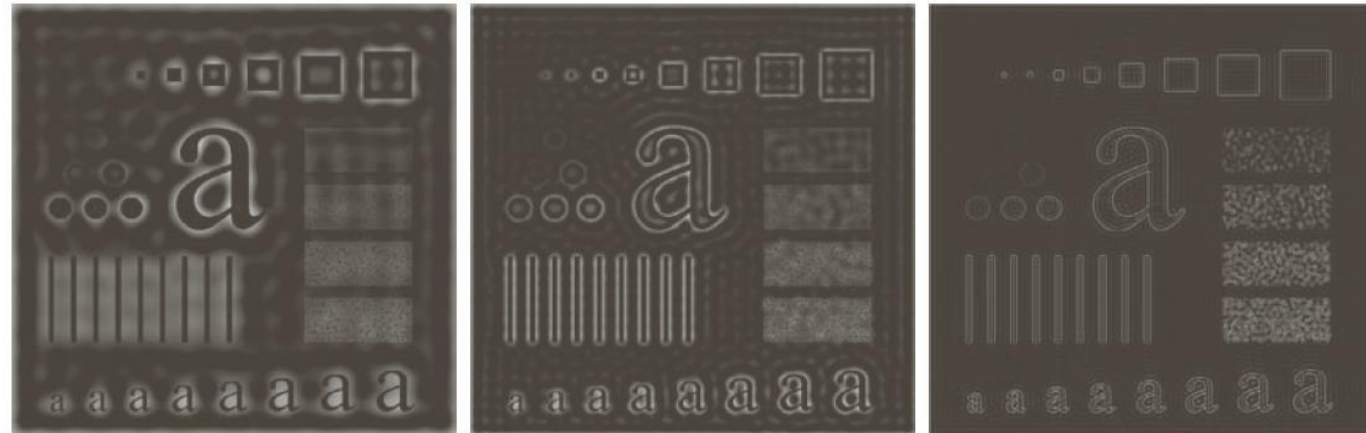
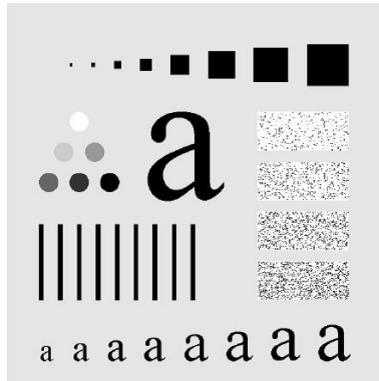
4.4 频率域滤波锐化

理想高通滤波器

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$



理想高通滤波器示例

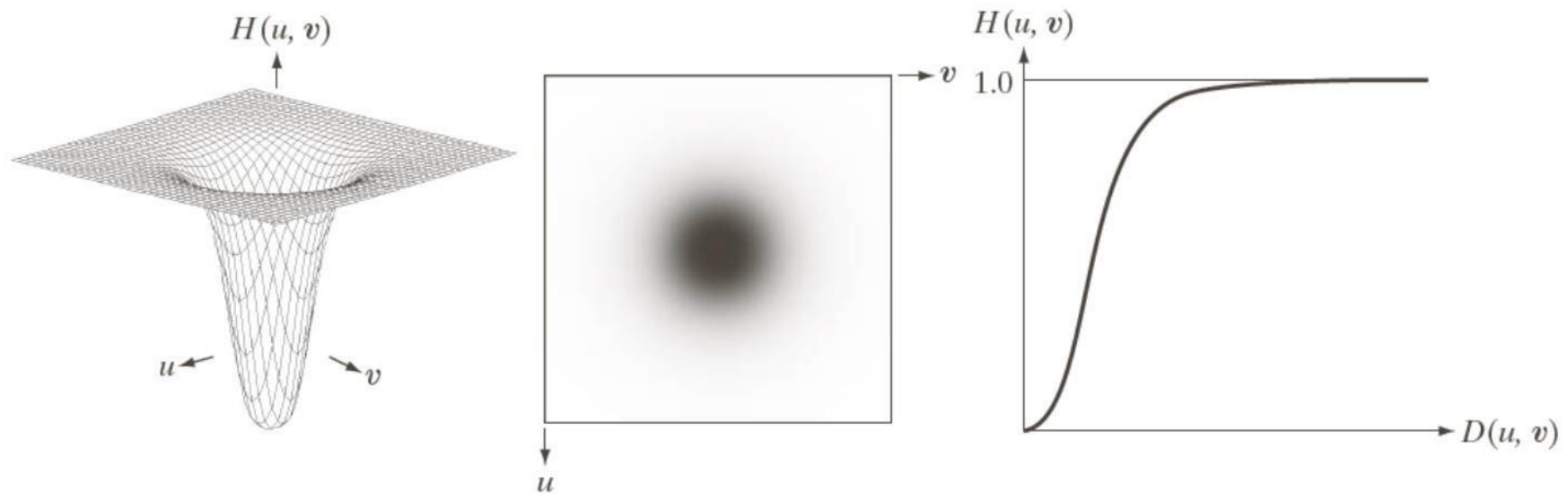


$D_0 = 30, 60, 100$ 的理想高通滤波器结果

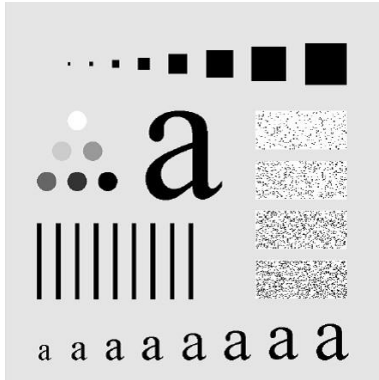
4.4 频率域滤波锐化

布特沃斯高通滤波器

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$



布特沃斯高通滤波器示例



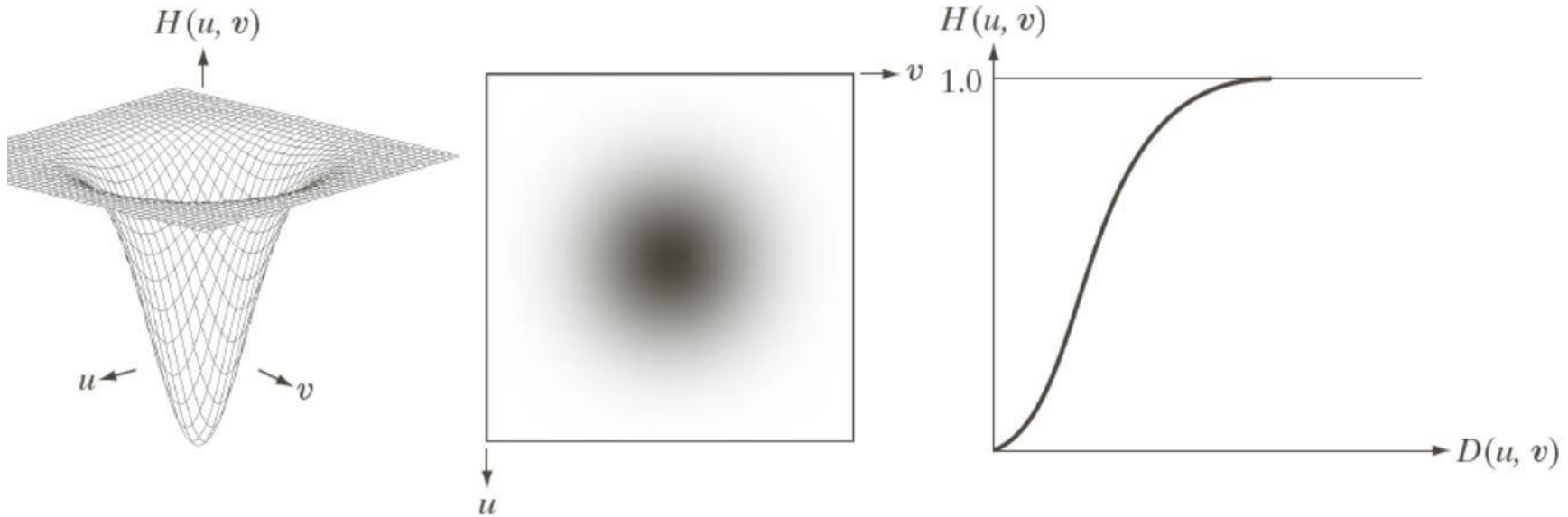
a b c

$D_0 = 30, 60, 100$ 的2阶布特沃斯高通滤波器结果

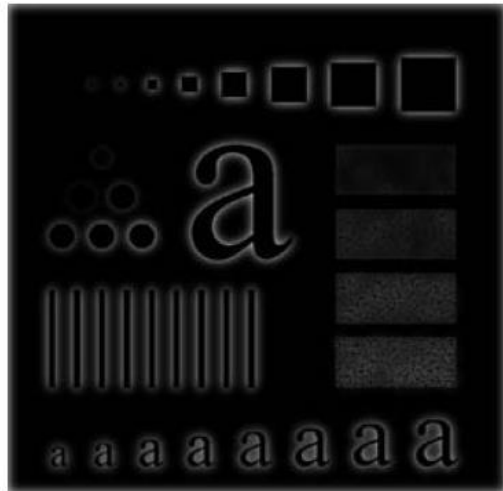
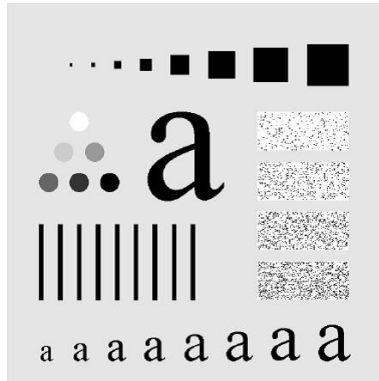
4.4 频率域滤波锐化

高斯高通滤波器

$$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2}$$



高斯高通滤波器示例



a b c

$D_0 = 30, 60, 100$ 的高斯高通滤波器结果



4.4 频率域滤波锐化：同态滤波

- 一幅图像 $f(x,y)$ 可以表示为照射分量和反射分量的乘积。

$$f(x,y) = i(x,y)r(x,y)$$

- 然而上式不能用来直接对两部分分量分别进行操作，因为两个函数乘积的傅立叶变换是不可分的。

$$F\{f(x,y)\} \neq F\{i(x,y)\}F\{r(x,y)\}$$

- 我们对图像函数两边取对数，则可以将两个分量分开。

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$$

$$F\{\ln f(x,y)\} = F\{\ln i(x,y)\} + F\{\ln r(x,y)\}$$



同态滤波步骤

1. 两边取对数: $f(x,y) = \ln i(x,y) + \ln r(x,y)$

2. 两边取傅氏变换: $F(u,v) = I(u,v) + R(u,v)$

3. 用频域函数 $H(u,v)$ 处理 $F(u,v)$:

$$H(u,v)F(u,v) = H(u,v) I(u,v) + H(u,v) R(u,v)$$

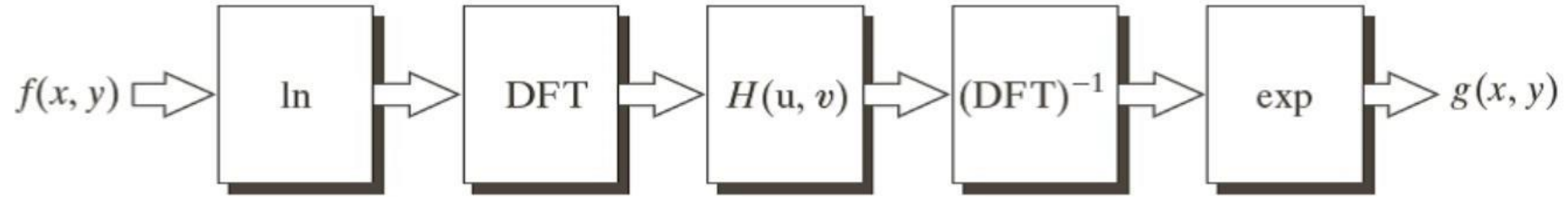
4. 反变换到空域:

$$s(x,y) = i'(x,y) + r'(x,y)$$

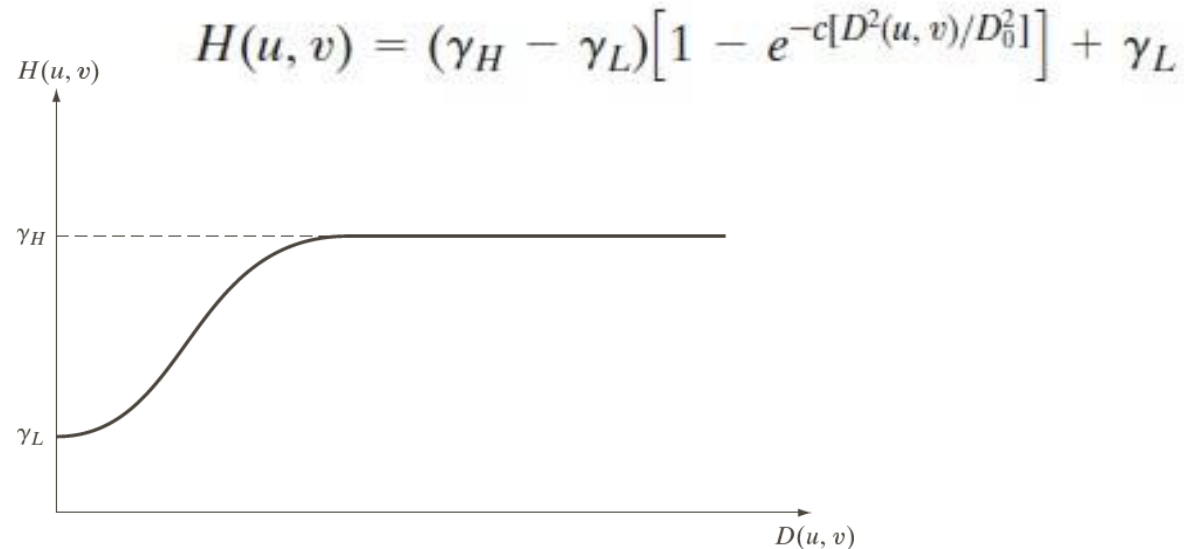
5. 两边取指数:

$$g(x,y) = \exp\{i'(x,y)\} \exp\{r'(x,y)\} = i_0(x,y) r_0(x,y)$$

同态滤波步骤及滤波器径向剖面图

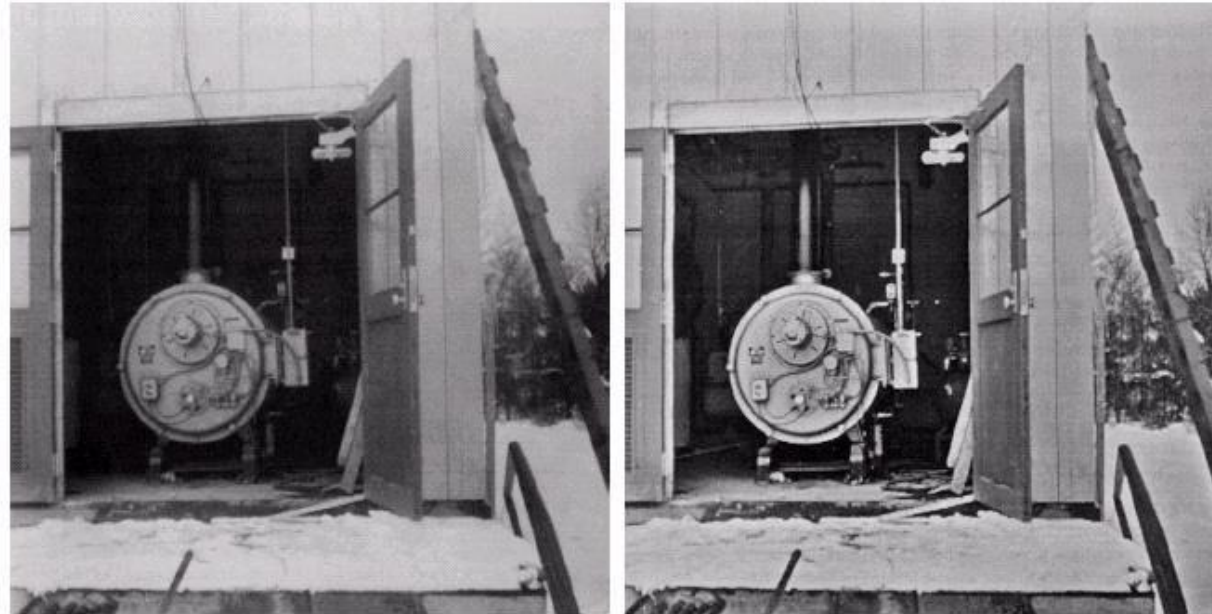


同态滤波器函数径向剖面图：**滤波器函数趋向于衰减低频的贡献，而增强高频的贡献。**



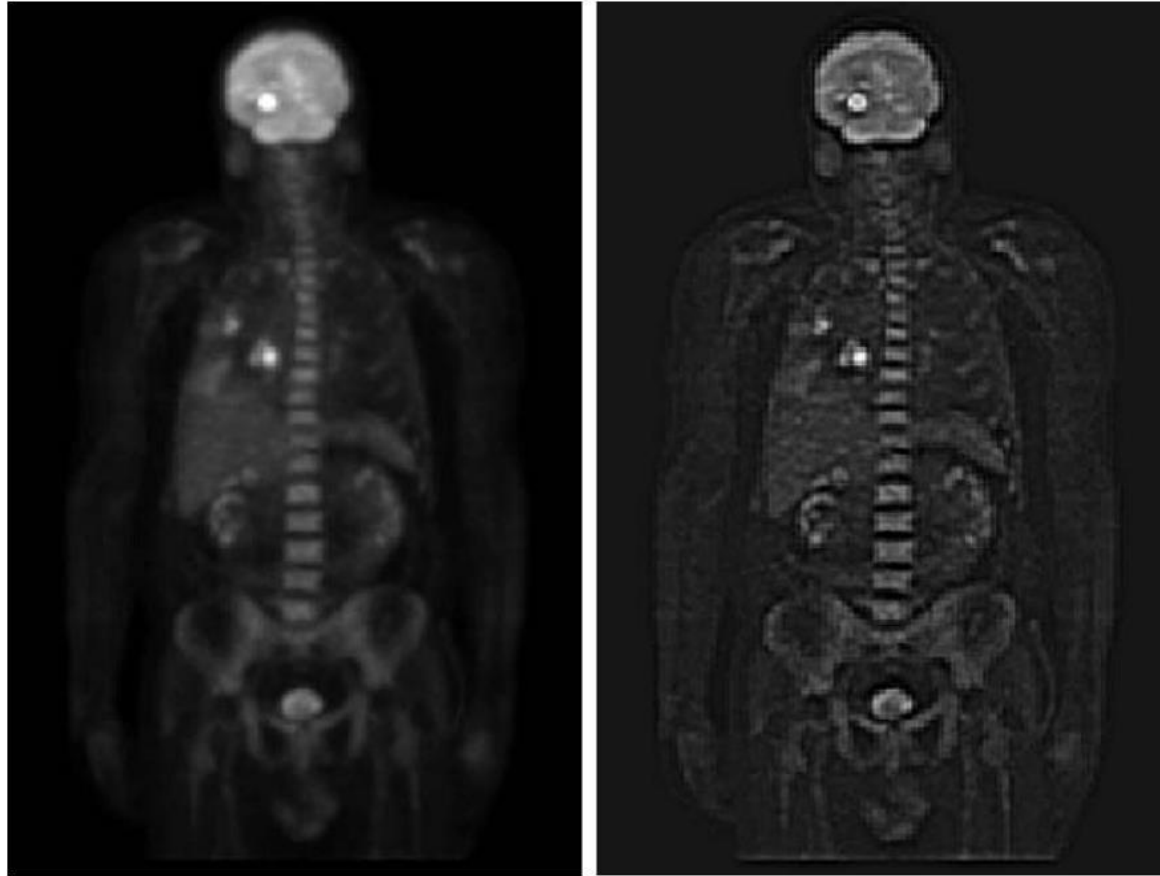
同态滤波示例

- **特点：**能消除乘性噪声，能同时压缩图象的整体动态范围和增加图象中相邻区域间的对比度
- 示例一：



同态滤波示例

同态滤波示例二：





第4章 频率域滤波

1. 离散傅立叶变换(DFT)
2. 频率域滤波基础
3. 频率域滤波器平滑图像
4. 频率域滤波器锐化图像
5. 选择性滤波
6. 其它变换(离散余弦变换)
7. 小波变换

带阻滤波器和带通滤波器

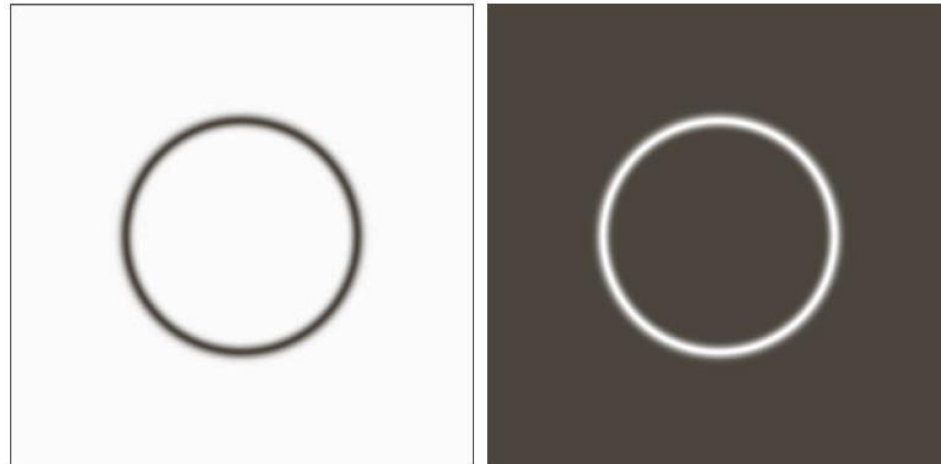
□ 理想、巴特沃斯、高斯带阻滤波器

TABLE 4.6

Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u, v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u, v) = \frac{1}{1 + \left[\frac{DW}{D^2 - D_0^2} \right]^{2n}}$	$H(u, v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW} \right]^2}$

□ 理想带阻、带通滤波器



a b

FIGURE 4.63

(a) Bandreject Gaussian filter.
 (b) Corresponding bandpass filter.
 The thin black border in (a) was added for clarity; it is not part of the data.

陷波滤波器

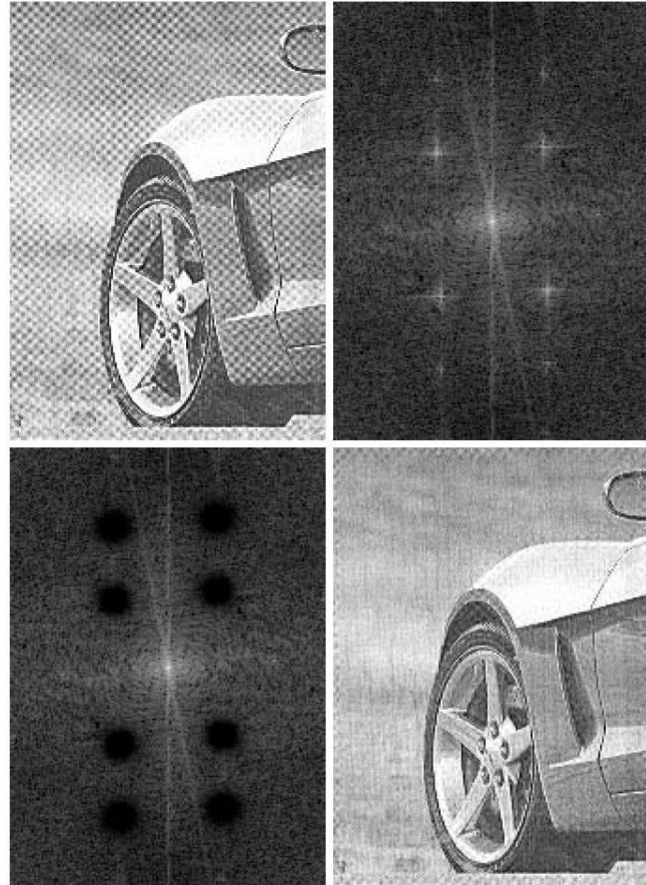
- 陷波滤波器：拒绝（或通过）事先定义的频率矩形中心的一个邻域的频率
- 一般设计为零相移滤波器，其关于原点对称
 - 零相移：不改变原图像频域表达式中的相角的滤波器
 - 一个中心位于 (u_0, v_0) 的陷波在位置 $(-u_0, -v_0)$ 处必须有一个对应的陷波
- 一般形式：

$$H_{NR}(u, v) = \prod_{k=1}^Q H_k(u, v) \cdot H_{-k}(u, v)$$

陷波滤波器可以在某一个频率点迅速衰减输入信号，以达到阻碍此频率信号通过的滤波效果的滤波器。陷波滤波器属于带阻滤波器的一种，只是它的阻带非常狭窄

陷波滤波器

□ 使用陷波滤波器减少莫尔纹



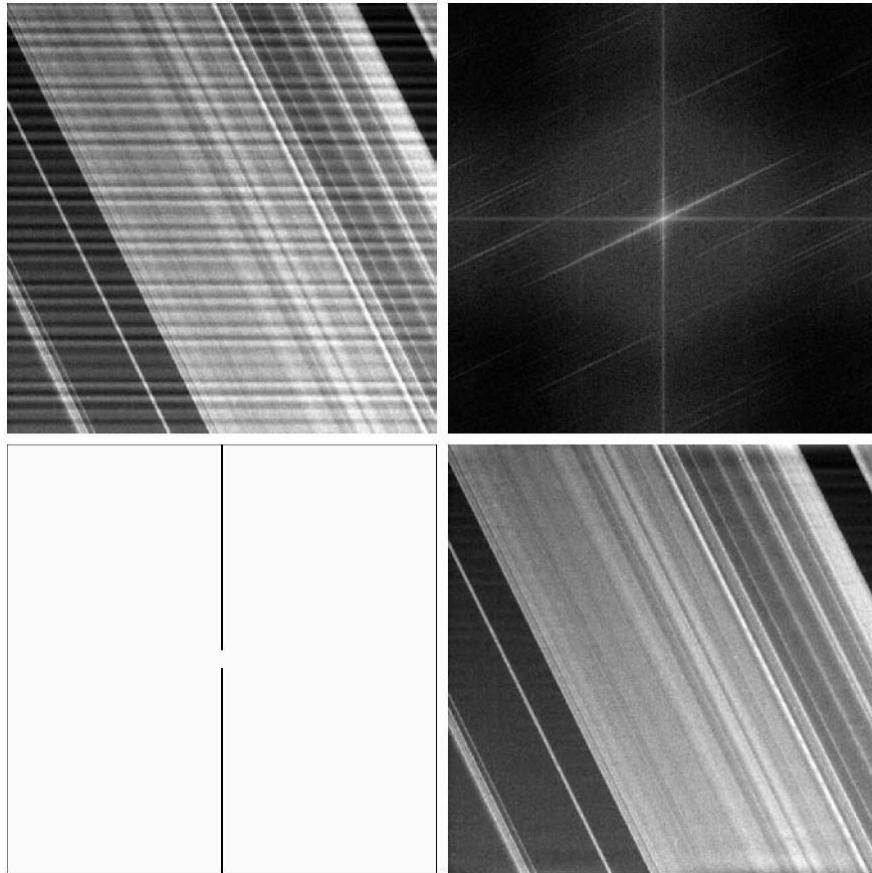
a b
c d

FIGURE 4.64

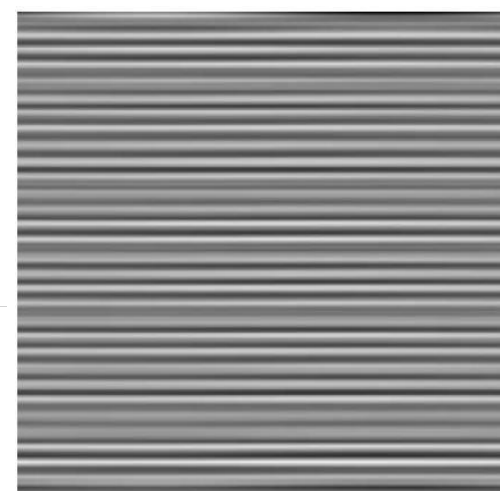
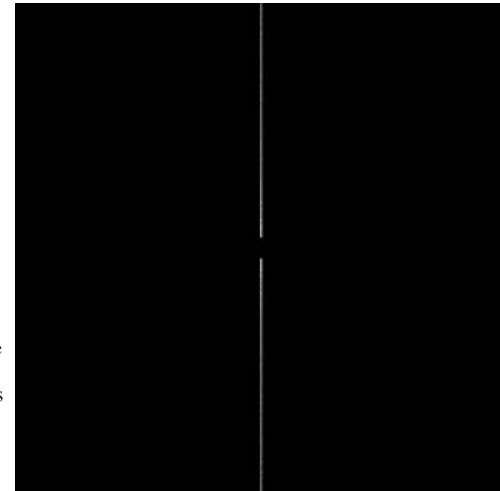
(a) Sampled newspaper image showing a moiré pattern.
(b) Spectrum.
(c) Butterworth notch reject filter multiplied by the Fourier transform.
(d) Filtered image.

陷波滤波器

□ 使用陷波滤波器增强土星图像



a b
 c d
FIGURE 4.65
 (a) 674×674 image of the Saturn rings showing nearly periodic interference. (b) Spectrum: The bursts of energy in the vertical axis near the origin correspond to the interference pattern. (c) A vertical notch reject filter. (d) Result of filtering. The thin black border in (c) was added for clarity; it is not part of the data. (Original image courtesy of Dr. Robert A. West, NASA/JPL.)



a b
FIGURE 4.66
 (a) Result (spectrum) of applying a notch pass filter to the DFT of Fig. 4.65(a). (b) Spatial pattern obtained by computing the IDFT of (a).





数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners





第4章 频率域滤波

1. 离散傅立叶变换(DFT)
 2. 频率域滤波基础
 3. 频率域滤波器平滑图像
 4. 频率域滤波器锐化图像
 5. 选择性滤波
 6. 其它变换(离散余弦变换)
 7. 小波变换
-



离散余弦变换(DCT)

- 问题的提出：傅里叶变换的一个最大问题是它的参数都是复数，在数据的描述上相当于实数的两倍。为此，我们希望有一种能够达到相同功能但数据量又不大的变换。

离散傅里叶变换(DFT)

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] \cdot e^{-\frac{i2\pi}{N}kn} \\ &= \sum_{n=0}^{N-1} x[n] \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right) \right] \end{aligned}$$

一般的实数序列的傅里叶变换为复数

由于许多要处理的信号都是实信号，在使用DFT时由于傅里叶变换时由于实信号傅立叶变换的共轭对称性导致DFT后在频域中有一半的数据冗余

离散余弦变换(DCT)

- 由傅里叶变换性质

- 当 $f(x)$ 为实的偶函数时，傅里叶变换域中得到实的偶函数

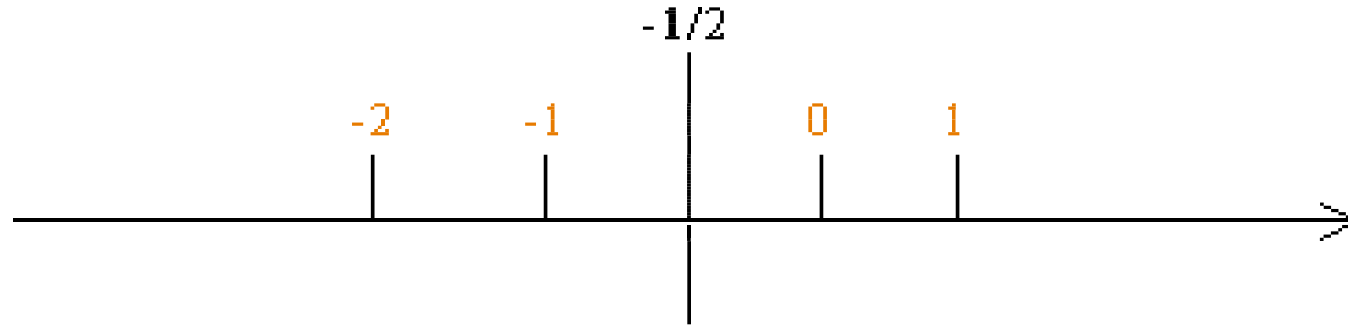
$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j \frac{2\pi ux}{N}} = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \left(\cos\left(\frac{2\pi ux}{N}\right) \right)$$

- 当 $f(x)$ 为偶函数时，傅里叶变换的计算公式虚部为零，只有余弦项

离散余弦变换（DCT）是对实信号定义的一种变换，变换后在频域中得到的也是一个实信号，相比DFT而言DCT可以减少一半以上的计算

■ 任意函数离散余弦变换

- 一个任意函数采样从 $0, 1, 2, \dots, N-1$, 若向负方向折叠形成 $2N$ 采样的偶函数, 就可以进行 $2N$ 的偶函数傅立叶变换



此时可采用离散余弦变换进行

余弦变换是简化傅立叶变换的一种方法

■ 一维离散余弦变换定义

$$F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi}{2N} (2x+1)u \right]$$

$$F(0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} f(x)$$

DCT是实值操作，计算较DFT快速、简单。

FDCT算法

■ 一维离散余弦反变换定义

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} c(u) F(u) \cos \left[\frac{\pi}{2N} (2x+1)u \right]$$

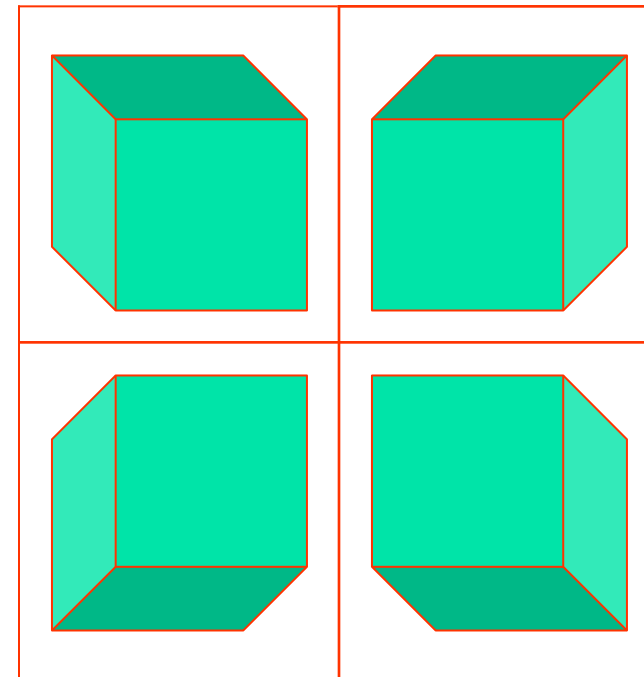
$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u = 1, 2, \dots, N-1 \end{cases}$$

■ 二维离散余弦变换

➤ 二维偶函数:

$$f(x, y) = \begin{cases} f(x, y) & x \geq 0, y \geq 0 \\ f(-x, y) & x < 0, y \geq 0 \\ f(x, -y) & x \geq 0, y < 0 \\ f(-x, -y) & x < 0, y < 0 \end{cases}$$

➤ 为折叠镜像序列





➤ 余弦变换的性质

- ✓ 序列的余弦变换是DFT的对称扩展形式
- ✓ 核可分离，可以用两次一维变换来执行
- ✓ 余弦变换的能量向低频集中
- ✓ 余弦变换有快速变换，和傅立叶变换一样，分奇偶组
- ✓ 余弦变换为实的正交变换，变换核的基函数正交

$$F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi}{2N} (2x+1)u \right] \quad \Rightarrow \quad T(u) = \sum_{x=0}^{N-1} f(x) r(x,u)$$

正变换核

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} c(u) F(u) \cos \left[\frac{\pi}{2N} (2x+1)u \right] \quad \Rightarrow \quad f(x) = \sum_{u=0}^{N-1} T(u) s(x,u)$$

反变换核

$$s(x,u) = \alpha(u) \cos \left(\frac{(2x+1)u\pi}{2N} \right)$$

反变换核

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \dots, N-1 \end{cases}$$

$$T(u) = \sum_{x=0}^{N-1} f(x) r(x,u)$$

正变换核

$$f(x) = \sum_{u=0}^{N-1} T(u) s(x,u)$$

反变换核



矩阵表示

$$\mathbf{t} = \mathbf{A} \mathbf{f} \quad \text{可逆变换对} \quad \mathbf{f} = \mathbf{A}^T \mathbf{t}$$

A的基向量是实的和正交的，又称为正交变换

$$\mathbf{t} = \begin{bmatrix} T(0) \\ T(1) \\ \vdots \\ T(N-1) \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{N-1} \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} s_0^T \\ s_1^T \\ \vdots \\ s_{N-1}^T \end{bmatrix} = \begin{bmatrix} s_0 & s_1 & \dots & s_{N-1} \end{bmatrix}^T$$

$$\mathbf{A} \mathbf{A}^T = \mathbf{A}^T \mathbf{A} = \mathbf{I}$$

N个基变换向量

$$\mathbf{s}_u = \begin{bmatrix} s(0,u) \\ s(1,u) \\ \vdots \\ s(N-1,u) \end{bmatrix} = \begin{bmatrix} s_{u,0} \\ s_{u,1} \\ \vdots \\ s_{u,N-1} \end{bmatrix} \quad \text{for } u = 0, 1, \dots, N-1$$

$$\text{例: } \mathbf{s}_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{s}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\mathbf{A} = [\mathbf{s}_0 \quad \mathbf{s}_1]^T = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- 变换矩阵A是实的，正交的，但不是对称的

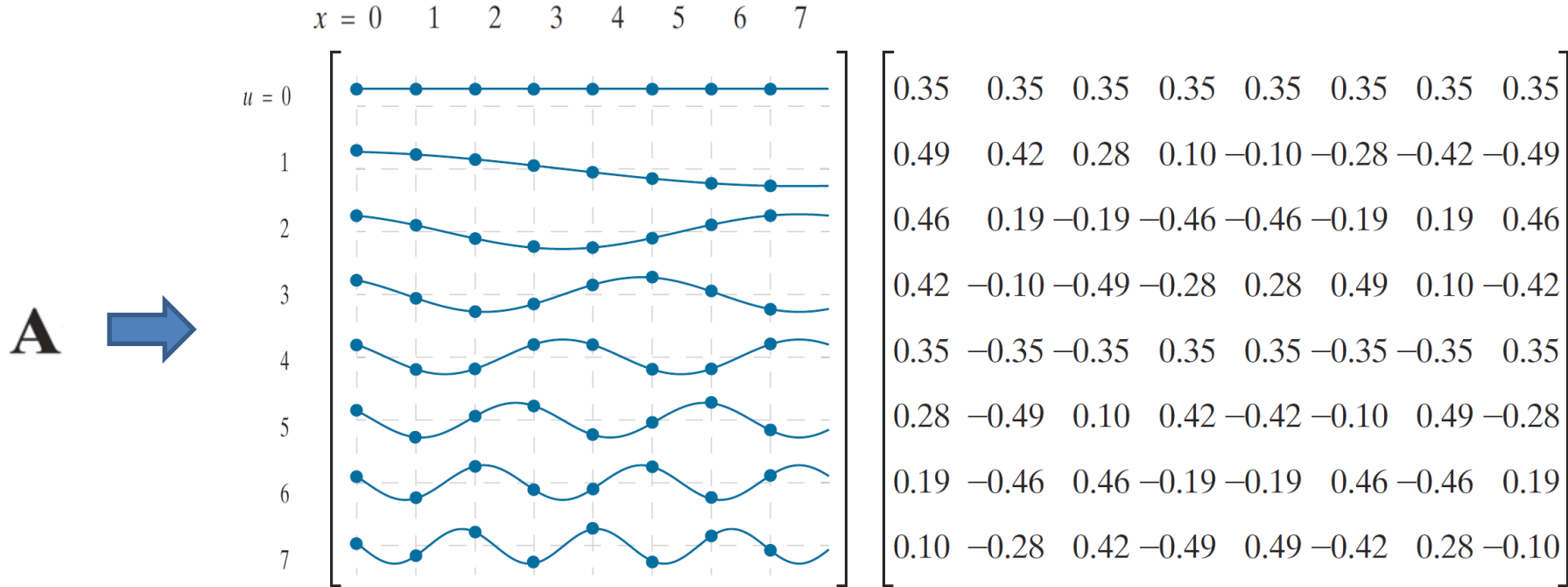


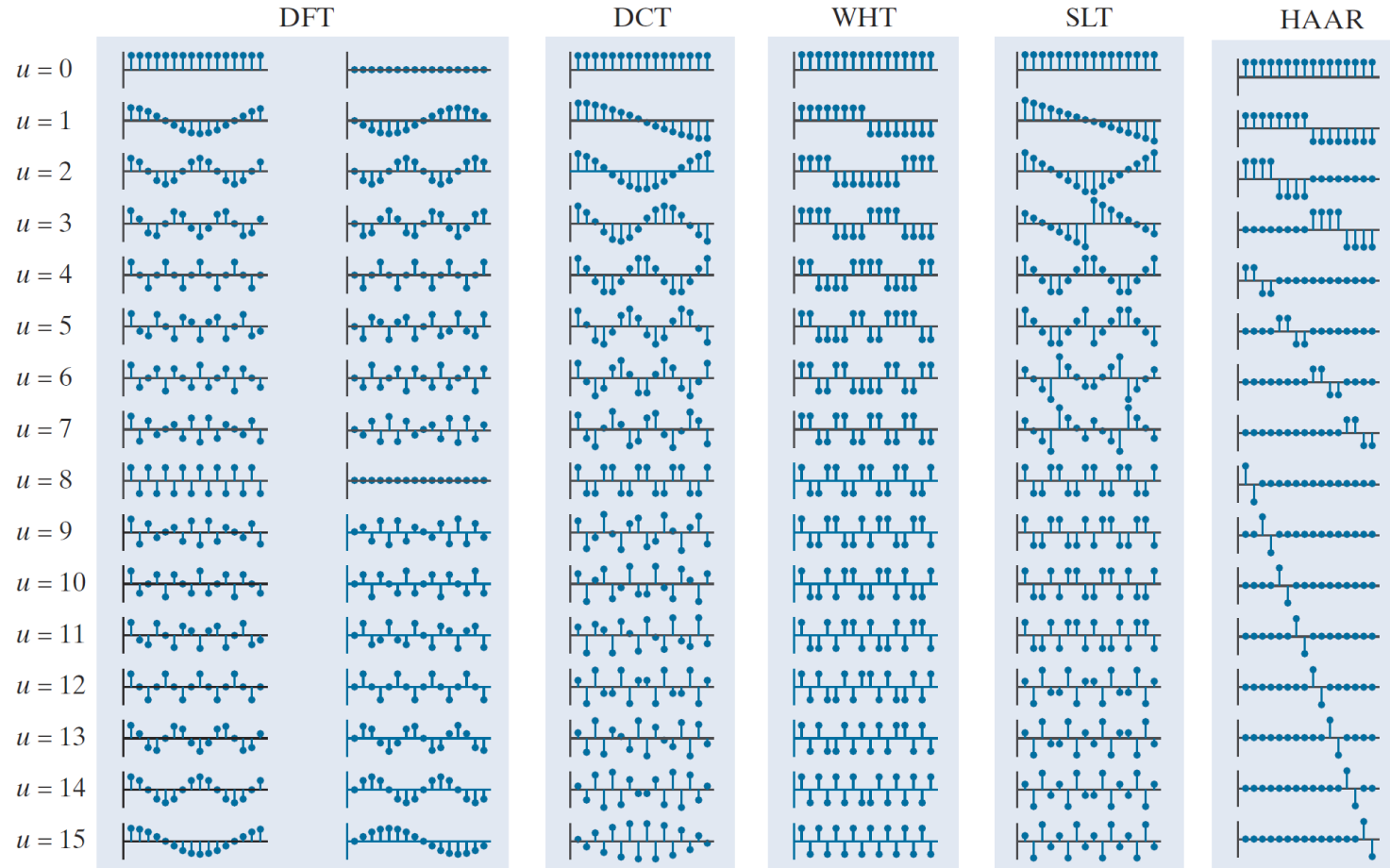
FIGURE 7.10 The transformation matrix and basis images of the discrete cosine transform for $N = 8$. (a) Graphical representation of orthogonal transformation matrix \mathbf{A}_C . (b) \mathbf{A}_C rounded to two decimal places, and (c) basis images.

$$s_u(x) \Leftrightarrow s(x, u) \text{ or } \mathbf{A} = [s_0 \ s_1 \ \dots \ s_7]^T$$

\mathbf{A} 里的每一项表达 \rightarrow \mathbf{A}^T 里的每一项表达

a	b	c	d
e	f	g	h

FIGURE 7.3
Basis vectors
(for $N = 16$) of
some commonly
encountered
transforms:
(a) Fourier basis
(real and imagi-
nary parts),
(b) discrete
Cosine basis,
(c) Walsh-Had-
amard basis,
(d) Slant basis,
(e) Haar basis,



➤ 二维离散余弦变换定义

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

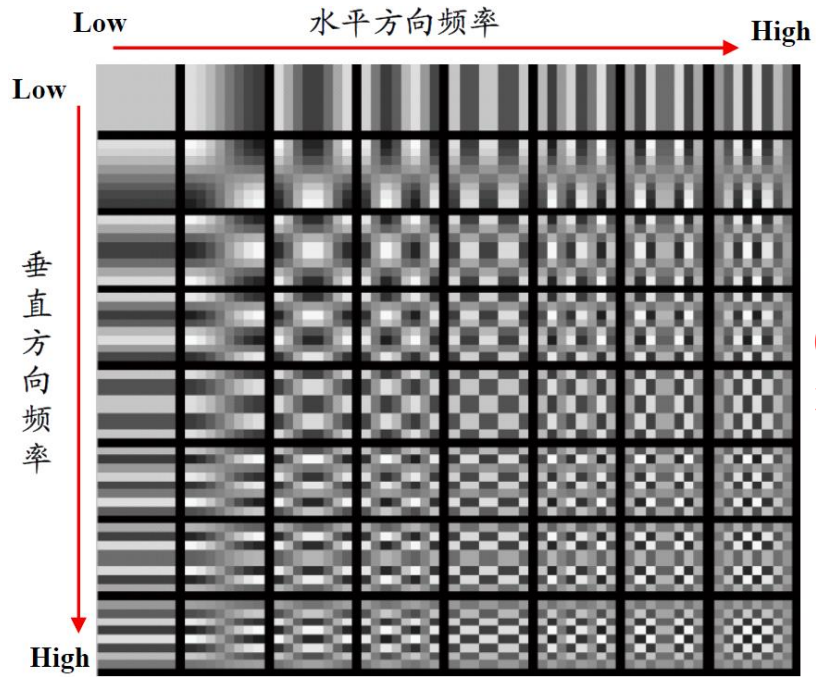
$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) \underbrace{F(u, v)}_{\text{权值}} \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

$$C(u) = C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u, v = 0 \\ 1 & u, v = 1, \dots, N-1 \end{cases}$$

考虑大小为 $N \times N$ 的子图象 $f(x, y)$ ，其离散变换 $T(u, v)$ 可以表示为如下的关系

$$T(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) r(x, y, u, v)$$

其中 $u, v = 0, 1, 2, \dots, n-1$ ， $r(x, y, u, v)$ 被称作是基函数或者是基图像，并且不能直接用 AA^T 来表示

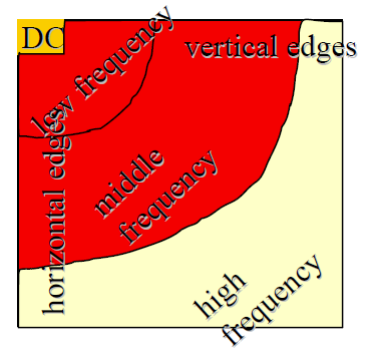


基图像

64个基础
线性组合

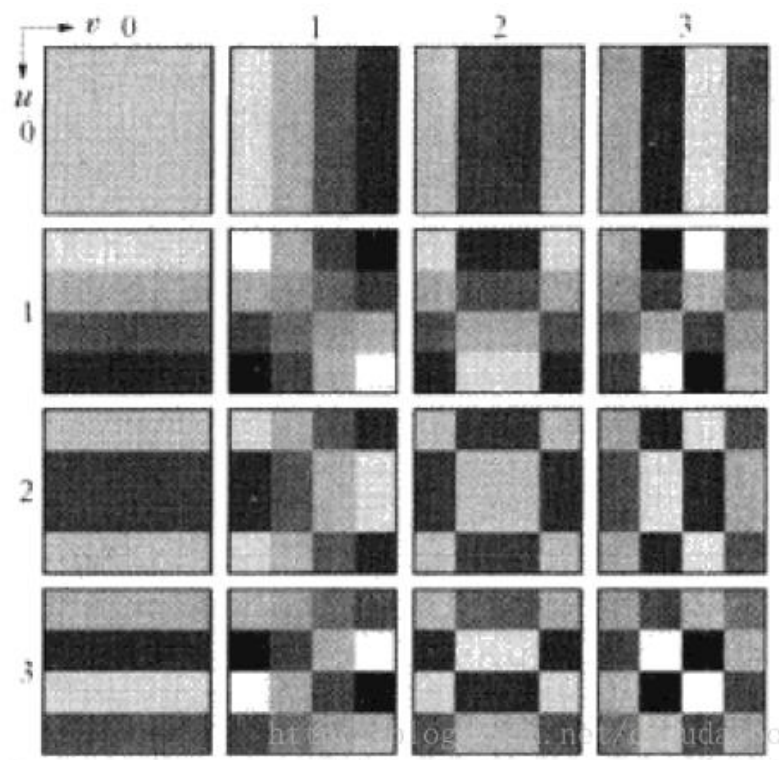
8x8的图像 $f(x, y)$ 是由64个基本图像的线性组合构成，如(0, 0)位置的元素表示的是直流分量

8x8 DCT
basis
images



可分离性：二维DCT可以用一维DCT来实现

4 × 4 DCT变换的基图像



这是根据基函数绘制出的由16个小块组成的基图像，其中每个小块由 4×4 个元素（子方块）组成。为了得到左上角块，我们令 $u = v = 0$ ，并画出 $x = y = 0, 1, 2, 3$ 时 $r(x, y, 0, 0)$ 的值。顶部行中的第二块是 $r(x, y, 0, 1)$ 在 $x = y = 0, 1, 2, 3$ 时的值的图像，依次类推可以画出其余所有的图像。

代数表达式变换对:

$$F(u,v) = \sum_x \sum_y^{M-1, N-1} f(x,y)g(x,y,u,v) \quad (\text{式 1})$$

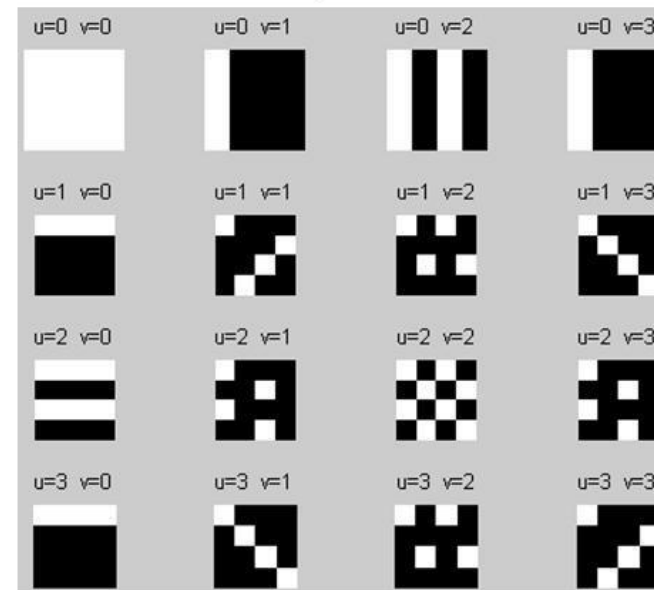
$$f(x,y) = \sum_u \sum_v^{M-1, N-1} F(u,v)h(x,y,u,v) \quad (\text{式 2})$$

$g(x,y,u,v)$ 为正变换核。反变换核 $h(x,y,u,v)$ 即是基函数，对于不同 u,v 下的 $M \times N$ 幅图像即是所谓基图像。

利用 $h(u,v)$ ，即可计算出基图像

$$\text{基图像: } h(u,v) = \begin{matrix} h(0,0,u,v) & h(0,1,u,v) & h(0,2,u,v) & h(0,3,u,v) \\ h(1,0,u,v) & h(1,1,u,v) & h(1,2,u,v) & h(1,3,u,v) \\ h(2,0,u,v) & h(2,1,u,v) & h(2,2,u,v) & h(2,3,u,v) \\ h(3,0,u,v) & h(3,1,u,v) & h(3,2,u,v) & h(3,3,u,v) \end{matrix}$$

$$\text{基图像: } h(u,v) = \begin{matrix} h(0,0,u,v) & h(0,1,u,v) & h(0,2,u,v) & h(0,3,u,v) \\ h(1,0,u,v) & h(1,1,u,v) & h(1,2,u,v) & h(1,3,u,v) \\ h(2,0,u,v) & h(2,1,u,v) & h(2,2,u,v) & h(2,3,u,v) \\ h(3,0,u,v) & h(3,1,u,v) & h(3,2,u,v) & h(3,3,u,v) \end{matrix}$$



DFT基图像



在图像的二维变换中，如果图像本身是**正方形**的（图像的长与宽相等），并且图像的变换核满足**可分离性**和**对称性**，则此时图像的二维变换可以用两个一维变换代替，并有如下的矩阵表示形式

$$\mathbf{T} = \mathbf{AFA}^T \quad \mathbf{F} = \mathbf{A}^T \mathbf{TA}$$

其中 \mathbf{A} 是进行图像一维变化时的矩阵，但 \mathbf{A} 不是图像二维变换的基函数

基图像就是固定 u, v 之后得到的变换核在有可能取值的 x, y 上构成的图像

$$\begin{array}{c}
 0 \\
 \downarrow \\
 \begin{bmatrix}
 \frac{1}{\sqrt{N}} [& 1 & 1 & \dots & 1] \\
 \sqrt{\frac{2}{N}} [& \cos \frac{\pi}{2N} & \cos \frac{3\pi}{2N} & \dots & \cos \frac{(2N-1)\pi}{2N}] \\
 \sqrt{\frac{2}{N}} [& \cos \frac{2\pi}{2N} & \cos \frac{6\pi}{2N} & \dots & \cos \frac{(2N-1)2\pi}{2N}] \\
 \vdots & \vdots & \vdots & \vdots & \vdots \\
 \sqrt{\frac{2}{N}} [& \cos \frac{(N-1)\pi}{2N} & \cos \frac{(N-1)3\pi}{2N} & \dots & \cos \frac{(N-1)(2N-1)2\pi}{2N}] \\
 \end{bmatrix} \\
 \downarrow \\
 u
 \end{array}
 \begin{array}{c}
 x \\
 \rightarrow
 \end{array}$$

<http://blog.csdn.net/dagudache>

根据一维变换核 $r(x, u)$ 得到的变换矩阵 A

图2. DCT变换矩阵及两坐标轴意义

$$\begin{array}{c}
 0 \\
 \downarrow \\
 \begin{bmatrix}
 \frac{1}{\sqrt{N}} & \sqrt{\frac{2}{N}} & \sqrt{\frac{2}{N}} & \dots & \sqrt{\frac{2}{N}} \\
 \left(\cos \frac{\pi}{2N} \right) & \left(\cos \frac{2\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)\pi}{2N} \right) \\
 \left(\cos \frac{3\pi}{2N} \right) & \left(\cos \frac{6\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)3\pi}{2N} \right) \\
 \vdots & \vdots & \vdots & \vdots \\
 \left(\cos \frac{(2N-1)\pi}{2N} \right) & \left(\cos \frac{(2N-1)2\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)(2N-1)2\pi}{2N} \right) \\
 \left(\cos \frac{(2N-1)\pi}{2N} \right) & \left(\cos \frac{(2N-1)2\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)(2N-1)2\pi}{2N} \right) \\
 \vdots & \vdots & \vdots & \vdots \\
 \left(\cos \frac{(2N-1)\pi}{2N} \right) & \left(\cos \frac{(2N-1)2\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)(2N-1)2\pi}{2N} \right) \\
 \left(\cos \frac{(2N-1)\pi}{2N} \right) & \left(\cos \frac{(2N-1)2\pi}{2N} \right) & \dots & \left(\cos \frac{(N-1)(2N-1)2\pi}{2N} \right)
 \end{bmatrix} \\
 \downarrow \\
 y
 \end{array}
 \begin{array}{c}
 v \\
 \rightarrow
 \end{array}$$

<http://blog.csdn.net/dagudache>

同样变换矩阵 A^T

图3. DCT变换矩阵的转置及两坐标轴意义

因此我们可以看到，当求图1 左上角的基图像时，只需要将图3 中的第一列乘以图2 中的第一行即可；当求图1 顶部行中的第二块的基图像 ($u = 0, v = 1$) 时，只需要图3 中的第二列乘以图2 中的第一行即可，依次便可以计算整幅基图像。

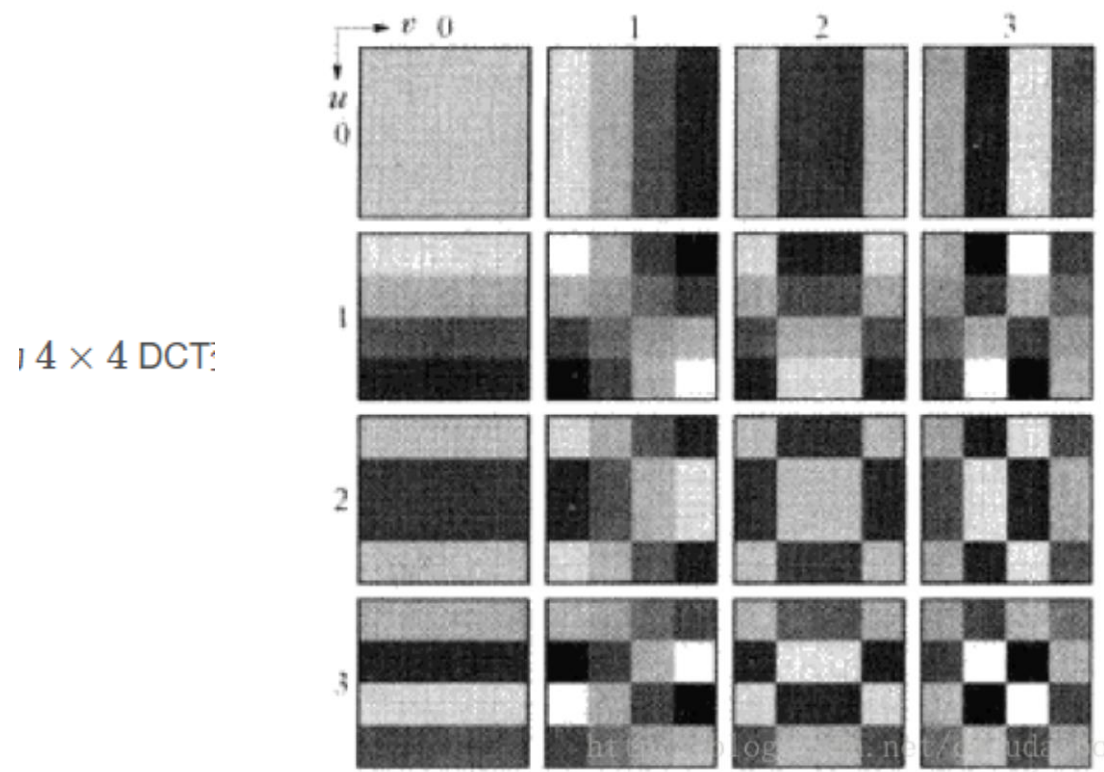
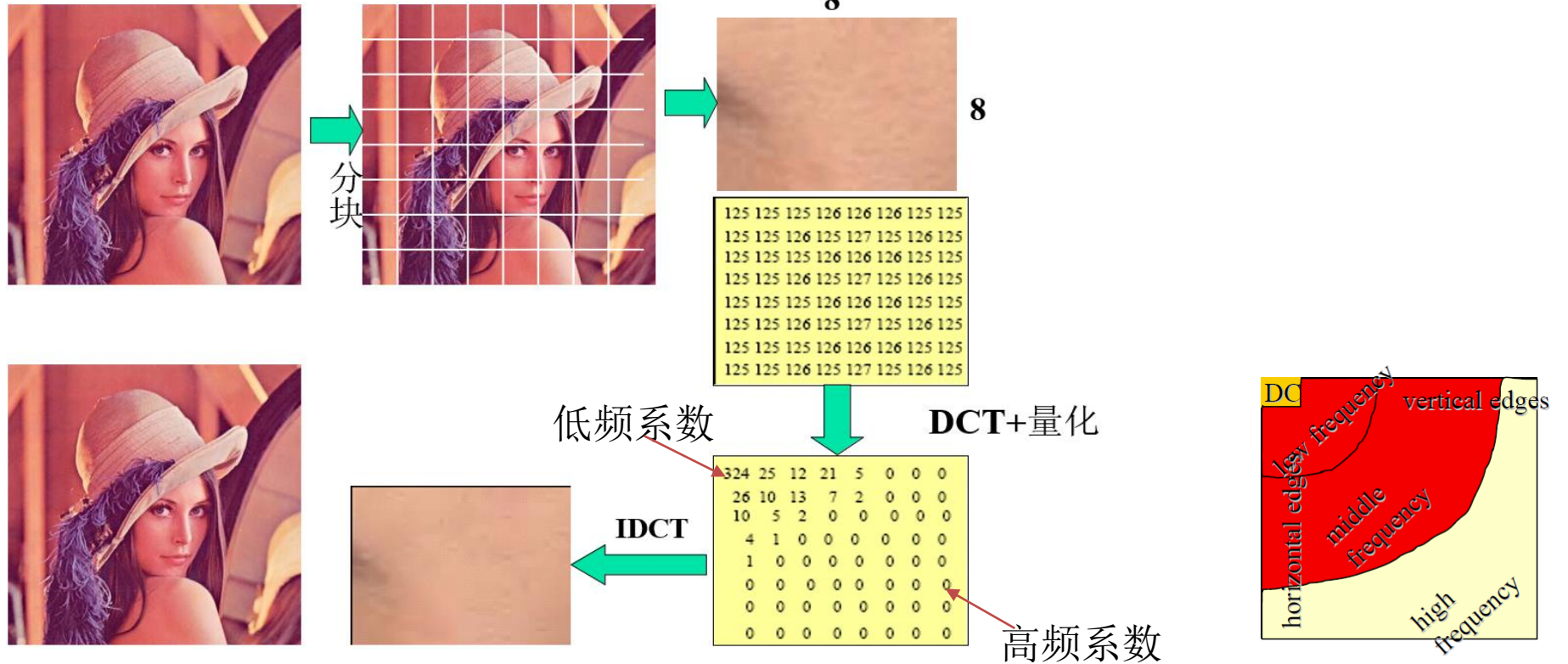


图1. $n=4$ 时的DCT基函数对应的基图像

➤ 余弦变换的应用—压缩编码

JPEG 图像压缩算法中，先分解图像为 8×8 或 16×16 的块，对每一块进行 DCT 变换，所得量化 DCT 系数进行编码和传送，形成压缩后的图像格式。



- (1) 小块图像的变换计算容易；
- (2) 距离较远的像素之间的相关性比距离较近的像素之间的相关性小。

通常变换后的图像，大多数的DCT系数值非常接近0，舍弃这些系数对重构图像质量影响不大。

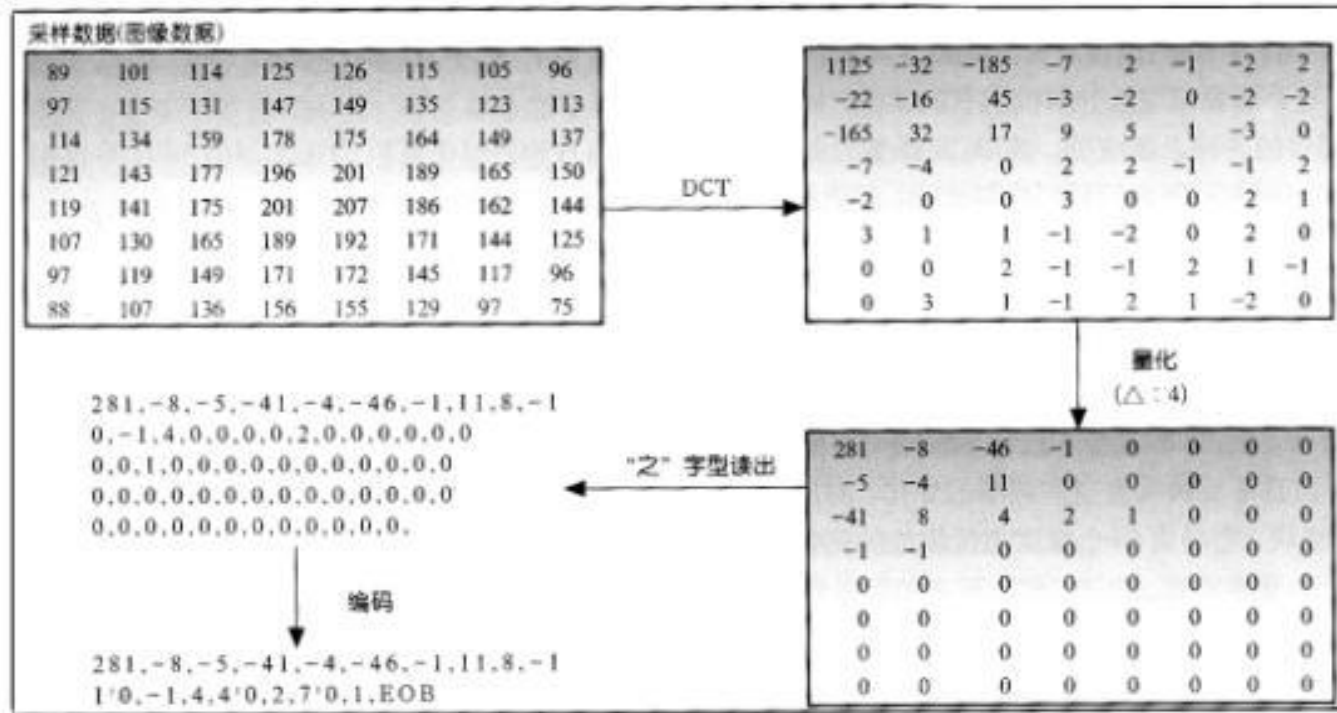


图 3

提高传输速度，节约图像存储空间

沃尔什-哈达玛变换

■ 沃尔什—哈达玛变换

- 能否进一步找到计算更简单的变换？
- 方法一：构建更为简单的正交函数集，只要满足正交关系：

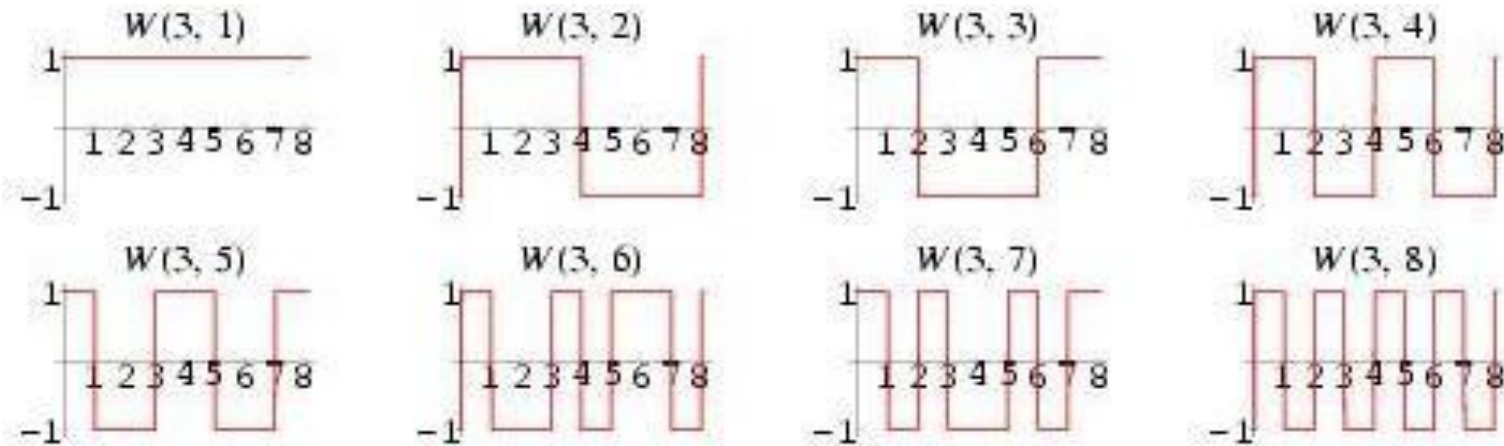
$$[A][A]^T = I$$

- 对正弦函数集进行深入研究，发现不考虑函数值，仅考数值的过零点位置分布时，可形成包含+1和-1极值状态下的正交函数集

- **1893**年法国数学家哈达玛总结前人研究只包含**+1**和**-1**的正交矩阵结果，形成哈达玛矩阵，既简单又有规律

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} \quad H_3 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} \quad \dots$$

- 1923年，美国数学家沃尔什提出**Walsh**函数，具有特点
 - 函数取值仅有两个（**0,1** 或**-1,+1**）
 - 由**Walsh**函数构成的**Walsh**函数集，具备正交性和完备性



- 按照哈达玛构造矩阵的排列方式，对Walsh函数进行排列，形成的函数集既满足正交性和完备性，又特别容易记忆和产生，将该正交函数集应用于信号变换，形成沃尔什—哈达玛变换 (Walsh-Hadamard)

反变换核 $s(x,u) = \frac{1}{\sqrt{N}} (-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$

优点

- 实对称正交函数变换，核仅取+1和-1，只需进行加、减运算
- 正反变换核相同，算法简单

- 二维沃尔什—哈达玛变换定义

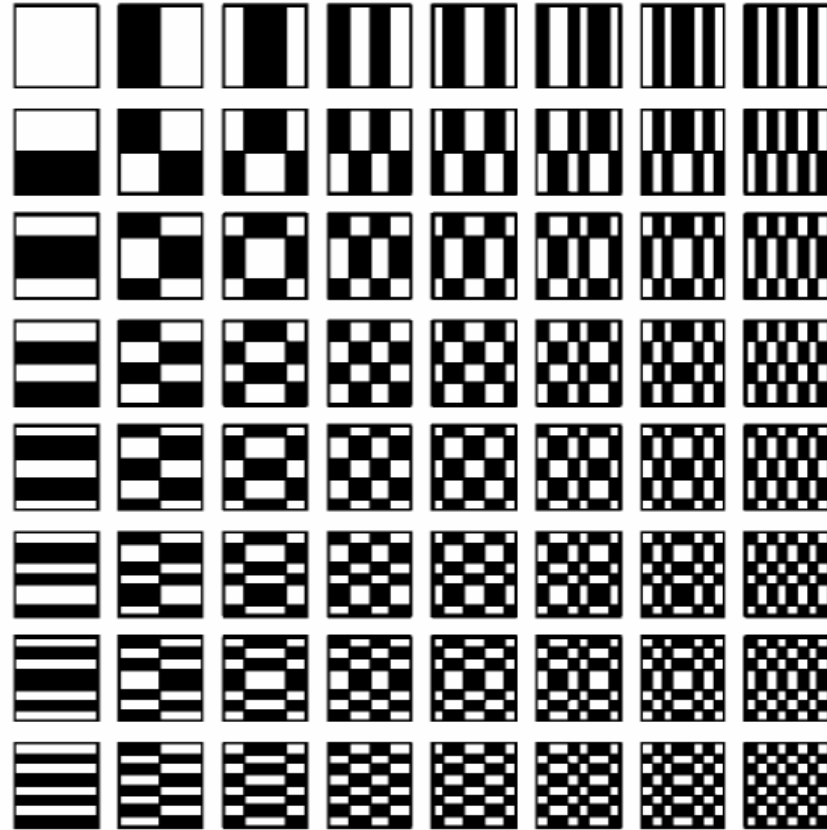
$$F_H = HfH \quad f = HF_H H$$

- 变换矩阵**H**具有递推公式：

$$H_1 = [1] \quad \mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{H}_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix}$$

■ 沃尔什——哈达玛变换 (Walsh-Hadamard)

$$H_8 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$



8x8 沃尔什—哈达玛变换基图像

■ 哈尔变换—Haar Transform

- 与沃尔什—哈达玛变换的构成方法相似，寻找其他可用的正交函数
- Haar函数于1910年提出，其矩阵只有+1，-1和另一个以 $\sqrt{2}$ 为基础的系数。它是正交**稀疏**矩阵，可实现快速计算
- 哈尔函数具备完备性与归一化正交性

$$N = 2, H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$N = 4, H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$N = 8, H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

■ 哈尔变换的定义

- 利用哈尔函数作为基函数的对称、可分离酉变换，要求 $N=2^n$

$$T = HFH$$

反变换核 $s(x, u) = \frac{1}{\sqrt{N}} h_u(x/N)$ for $x = 0, 1, \dots, N - 1$

$N \times N$ Haar matrix

$$\mathbf{H}_N = \begin{bmatrix} h_0(0/N) & h_0(1/N) & \dots & h_0(N-1/N) \\ h_1(0/N) & h_1(1/N) & & \vdots \\ \vdots & & \ddots & \\ h_{N-1}(0/N) & \dots & & h_{N-1}(N-1/N) \end{bmatrix}$$

$$\mathbf{A}_H = \frac{1}{\sqrt{N}} \mathbf{H}_N$$

For example, if $N = 2$, $\mathbf{A}_H = \frac{1}{\sqrt{2}} \begin{bmatrix} h_0(0) & h_0(1/2) \\ h_1(0) & h_1(1/2) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

- 令整数 $0 \leq k \leq N-1$ 由其他两个整数 p 和 q 唯一决定，索引 p 规定了 **尺度**， q 决定了 **平移量**

$$k = 2^p + q - 1$$

- 哈尔函数定义为：

$$h_0(x) = \frac{1}{N} \quad h_k(x) = \frac{1}{N} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \leq x < \frac{q-0.5}{2^p} \\ -2^{p/2} & \frac{q-0.5}{2^p} \leq x < \frac{q}{2^p} \\ 0 & \text{其他} \end{cases}$$

- 可以计算 $N=8$ 时 p 和 q 的关系：

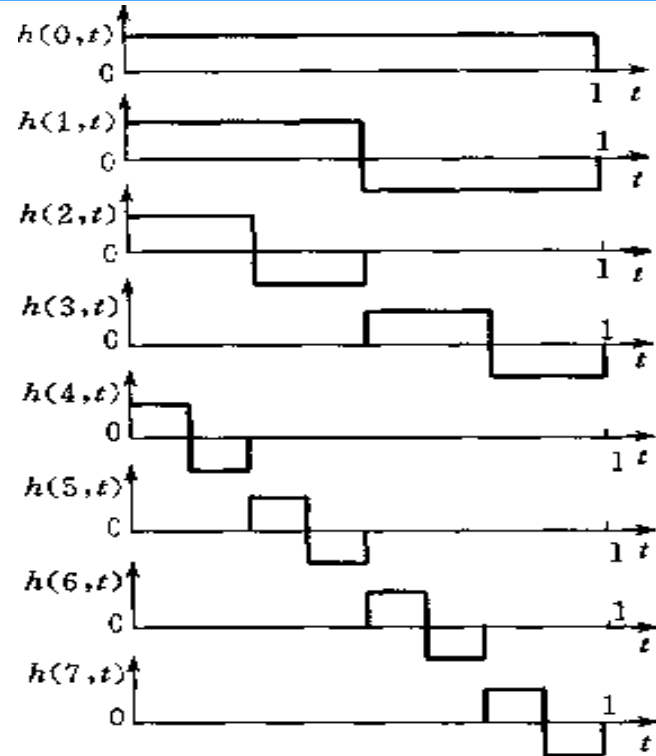
$N=8$	k	0	1	2	3	4	5	6	7
$n=3$	p	0	0	1	1	2	2	2	2
	q	0	1	1	2	1	2	3	4

■ N = 8 哈尔函数及波形

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

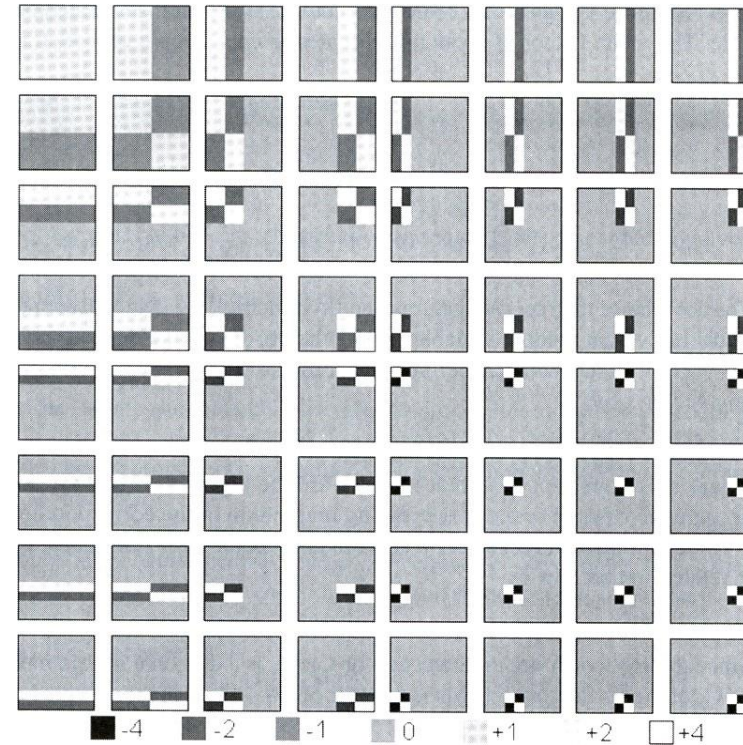
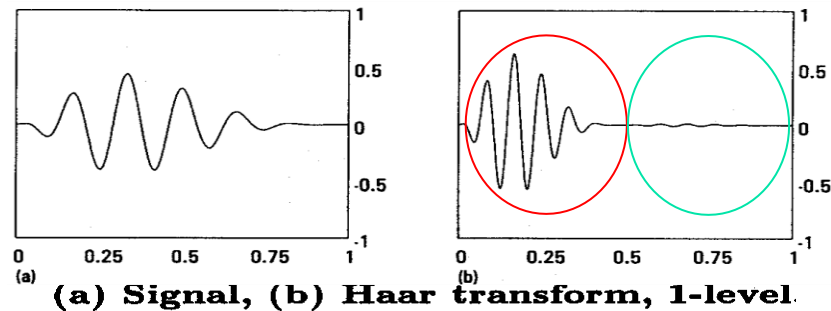
自身特征:

- 哈尔矩阵既反映**整体**，又反映**局部** $T = HFH$
- 第一行反映一种整体变换，第二行反映两个半幅变换，下两行是N/4之间的变换，最后是相邻两像元之间的变换
- 如果一个信号、或信号中的一部分，可以近似地匹配上某一基函数，则在变换后，会产生一个对应那个基函数的较大的变换系数，而对于其他正交基函数产生较小的系数。由此，哈尔变换可以给出一些线和边的尺寸和位置信息



■ $N=8$ 哈尔基图像

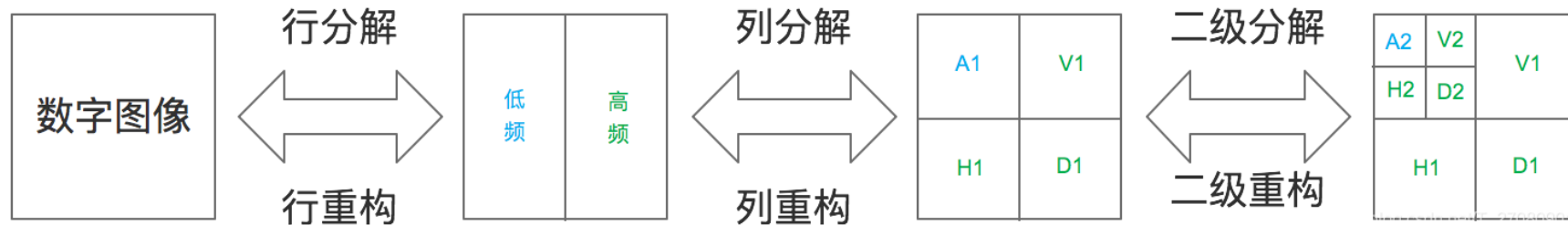
- 傅里叶变换的基函数仅是频率不同，哈尔函数在尺度和位置上都不同
- 哈尔变换具有尺度和位置的双重性
- 全域特性和区域特性：哈尔函数系列可分为全域部分和区域部分。全域部分作用于整个变换区间，区域部分作用于局部区域
- 小波特性：



注意：右下象限部分可以用来搜索图像中不同位置的小特征

■ 哈尔变换步骤:

- 首先, 沿着矩阵的每一行做一维的Haar变换;
- 然后, 沿着矩阵的每一列做一维的哈尔变换;
- 对于每个低频分量矩阵 (近似信息) 重复前两个步骤直到完成指定的等级划分



A表示近似信息 (approximation coefficients)

H表示水平细节信息 (horizontal detail coefficients)

V表示垂直细节信息 (vertical detail coefficients)

D表示对角线细节信息 (diagonal detail coefficients)

- 示例: 给定一个二维信号, 假设是一个 4×4 图像

$$f = \begin{bmatrix} 2 & 1 & 5 & 6 \\ 7 & 6 & 5 & 8 \\ 2 & 1 & 5 & 5 \\ 7 & 7 & 2 & 10 \end{bmatrix}$$

第一次行分解得到低频信息

$$L = \begin{bmatrix} \frac{3}{\sqrt{2}} & \frac{11}{\sqrt{2}} \\ \frac{13}{\sqrt{2}} & \frac{13}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} & 5\sqrt{2} \\ 7\sqrt{2} & 6\sqrt{2} \end{bmatrix}$$

第一次行分解得到高频信息

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & -4\sqrt{2} \end{bmatrix}$$

对L进行列高频分解得到 $A_1 = \begin{bmatrix} 8 & 12 \\ 8.5 & 11 \end{bmatrix}$

对L进行列低频分解得到 $H_1 = \begin{bmatrix} -5 & -1 \\ -5.5 & -1 \end{bmatrix}$

对H进行列高频分解得到 $V_1 = \begin{bmatrix} 1 & -2 \\ 0.5 & -4 \end{bmatrix}$

对H进行列低频分解得到 $D_1 = \begin{bmatrix} 0 & 1 \\ 0.5 & 4 \end{bmatrix}$

对 A_1 继续进行二层分解.....

原始图像



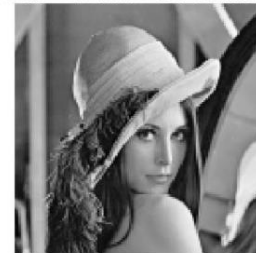
小波分解系数



压缩一 (去掉第一层高频信息)



压缩二 (去掉第二层高频信息)



- 对图片做Haar变换，然后我们去掉其高频信息部分，实现对图像的**压缩**



经过二次分解，然后分别过滤掉第一层的高频信息和第一层兼第二层的高频信息的效果！过滤掉第一层的高频信息，图像压缩为原来的四分之一。过滤掉第二层和第三层的高频信息以后，可以看到图片稍微有点模糊了。



第4章 频率域滤波

1. 离散傅立叶变换(DFT)
2. 频率域滤波基础
3. 频率域滤波器平滑图像
4. 频率域滤波器锐化图像
5. 选择性滤波
6. 其它变换(离散余弦变换)
7. 小波变换

傅里叶变换的局限

- 傅里叶变换的引入，使人类对于自然界复杂信号或图像的分析与处理，转变为对于其相对较为简单的频域特征的分析与处理

➤ 回忆调和信号与傅里叶变换 $e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$

$$F(\omega) = \int f(t)e^{-j\omega t} dt \quad f(t) = \frac{1}{2\pi} \int F(\omega)e^{j\omega t} d\omega$$

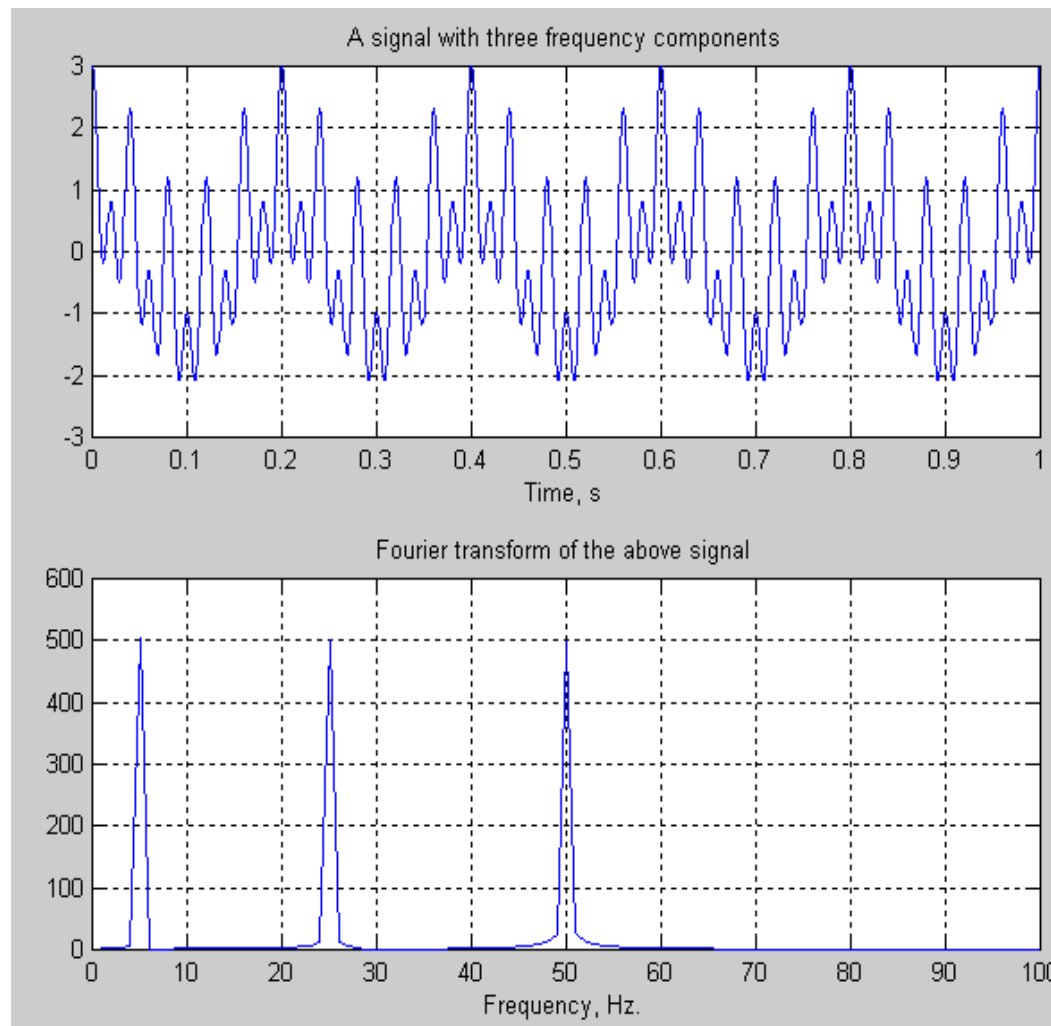
- 实际上相当于将调和信号频率与实际信号相比较
- 若实际信号中含有对应频率，则具有较大的变换系数
- 若实际信号中没有某特定频率，则该频率对应的系数较小或为零

- 傅里叶变换可以准确地知道信号中含有哪些频率成分，但不知道这些成分发生的时间、位置
- 稳态信号特征
 - 由一系列不随时间变化的频率组成
 - 不需要知道任何频率的开始与停止时间
 - 傅立叶变换基于在时间轴上无限伸展的正弦曲线波作为正交基函数，十分适于表现稳态信号
- 非稳态信号具有随时间变化的频率成分，分析中需要知道
 - 什么频率在什么时候发生
 - 特定频率发生的位置

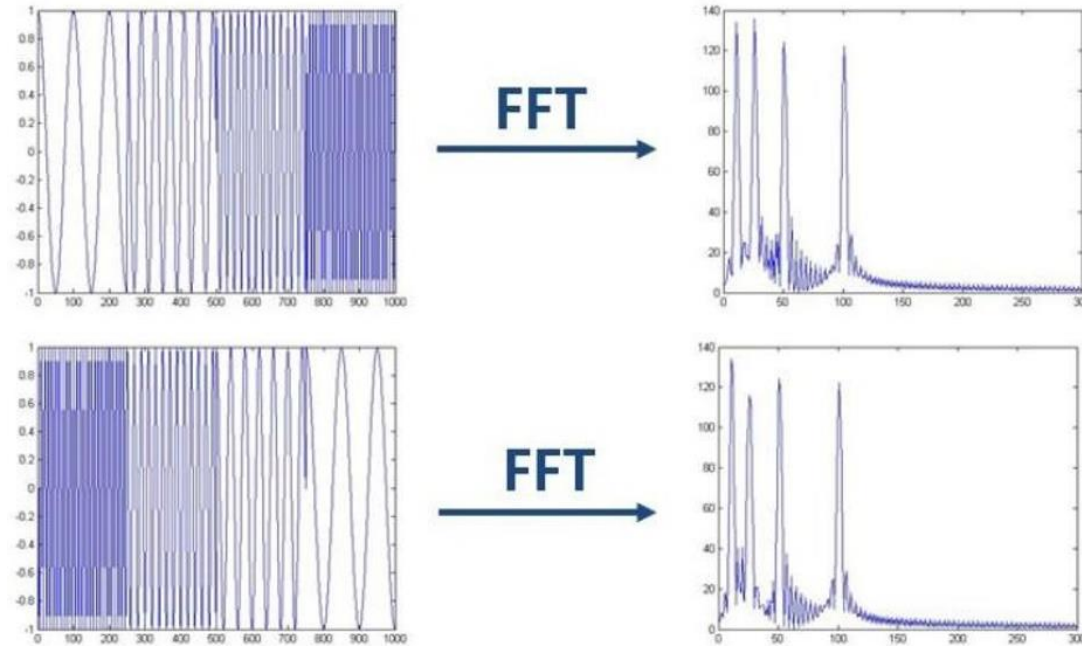
➤ 稳态信号

$$\begin{aligned}x_4(t) &= \cos(2\pi \cdot 5 \cdot t) \\ &+ \cos(2\pi \cdot 25 \cdot t) \\ &+ \cos(2\pi \cdot 50 \cdot t)\end{aligned}$$

缺乏时间信息!



➤ 非稳态信号，频率随着时间变化



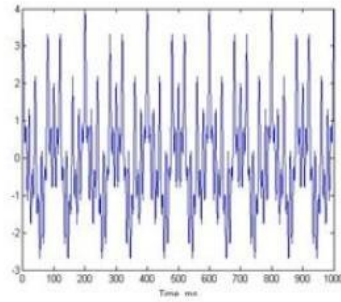
无法根据结果判断一个特定信号是在什么时候发生的



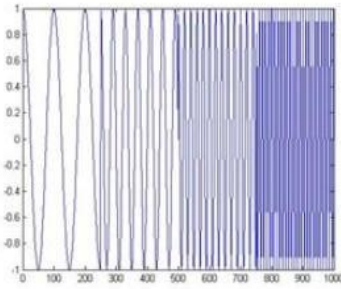
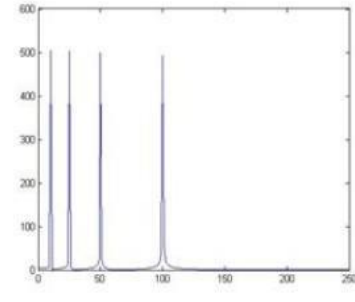
- 平稳信号大多是人为制造出来的，自然界的大量信号几乎都是非平稳的，分析需要提取某一时间段的频域信息或某一频率段所对应的的时间信息

可以准确的知道频率成分，但不知道这些频率出现的位置和延续的时间

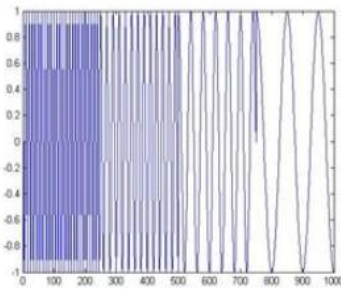
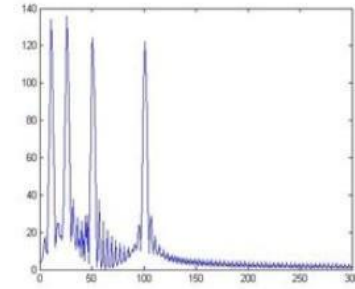
傅里叶变换适用于分析平稳信号



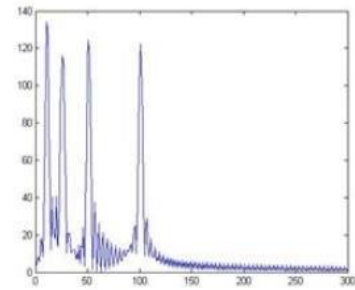
FFT



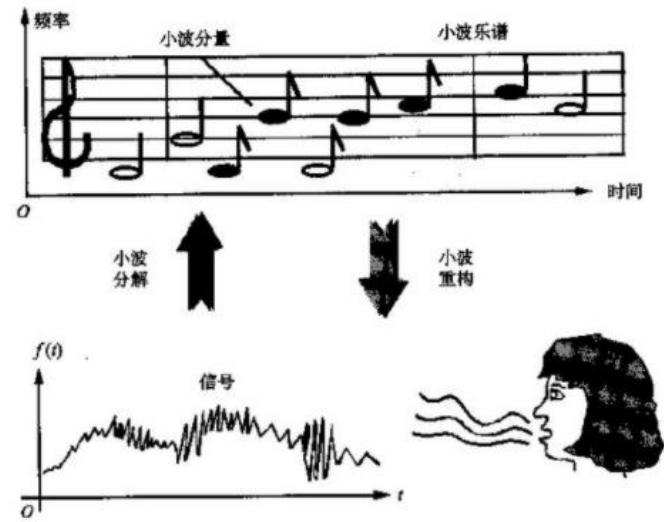
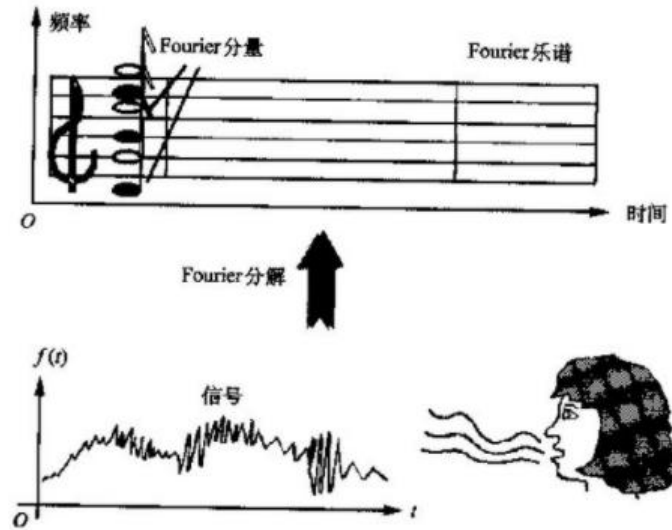
FFT



FFT



- 一个连续乐谱，不光阐明了要演奏的音符或频率，而且需要阐明何时要演奏。而傅里叶变换，只提供了全局的音符或频率信息，局部信息在变换过程中丢失了。





■ 傅里叶变换的局限性

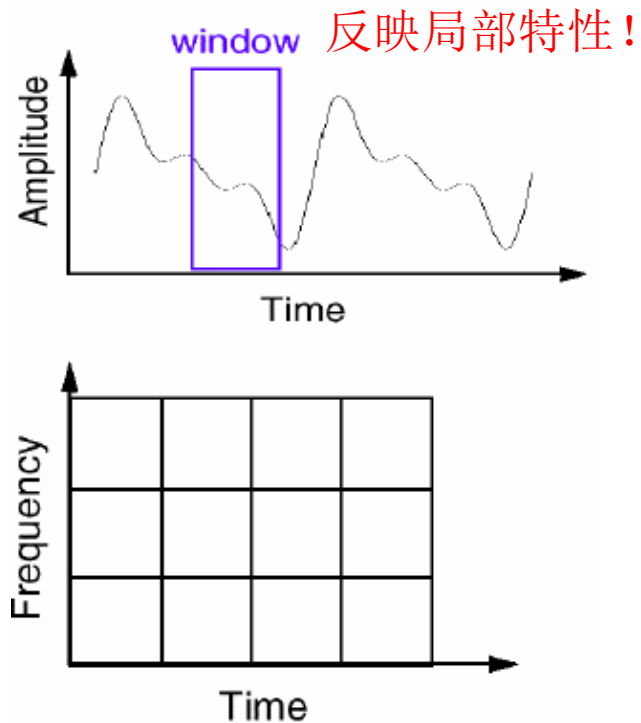
- 时间信息损失：什么时候特定的事件发生？
- 位置信息的损失：傅里叶变换不能确定某一事件的漂移、趋势、突变、起始和结束等
- 傅里叶变换分析是全局性的分析，难以分析局部信号特征



■ 傅里叶变换的其他局限性

- 对于瞬态信号或高度局部化的信号（例如边缘），由于这些成分并不类似于任何一个傅立叶基函数，它们的变换系数（频谱）不是紧凑的，频谱上呈现出一幅相当混乱的构成
- 这种情况下，傅立叶变换是通过复杂的安排，以抵消一些正弦波的方式构造出在大部分区间都为零的函数而实现的
- 对于包含瞬态或局部变化成分的信号，傅里叶变换分析将难于得到最佳的表示

- 为了克服上述缺陷，使用有限宽度基函数的变换方法
- **Gabor**变换（1946），或称之为加窗付里叶变换、短时傅里叶变换（**STFT**）首先产生：



- STFT变换步骤为：

- 1) 选定一个有限窗口
 - 2) 将窗口放置于信号的起点
 - 3) 计算窗口内信号的傅里叶变换
 - 4) 将窗口向右移动一个距离
 - 5) 重复3) 4) 步，直至达到信号的末尾
- 由此，得到每个时间段内信号的频率成分

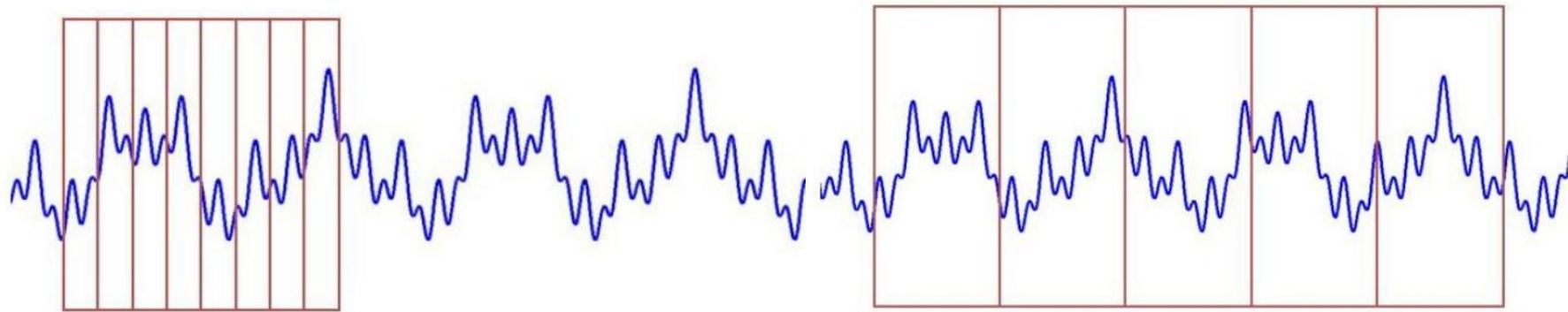


- Gabor变换具有特征:

- 实现了对于信号的频率与时间观察的折衷
- 无论时间还是频率的观察均为有限精度；整体精度取决于窗口尺寸
- 一旦窗口尺寸确定，将作用于所有频率

实际信号需要在时间与频率方面更为灵活的分析

- 窗太窄，窗内的信号太短，会导致频率分析不够精准，频率分辨率差
- 窗太宽，时域上又不够精细，时间分辨率低

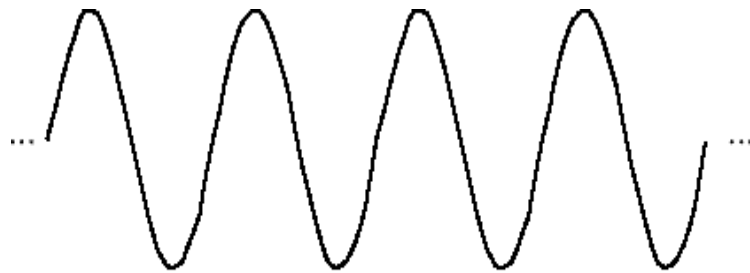


框太窄 → 频率分辨率差

框太宽 → 时间分辨率差

- ◆ 其窗函数的大小和形状均保持固定不变，对于分析时变信号不利。
(高频信号持续时间短，低频信号持续时间长)
- ◆ 不能构成正交基，给数值计算带来不便

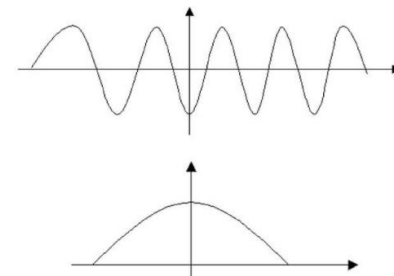
- 进一步的发展—采用频率不同、位置不同、宽度有限的基函数进行变换：小波变换出现，窗口大小随频率变化，且为正交基
- 哈尔变换—最早出现的小波变换实例，其基向量均为一个函数通过不断的平移和伸缩来产生
- 什么是小波？小区域的波，具有有限区间和均值为零的波，波形呈正负交替的波动，也即直流分量为0



Sine Wave



Wavelet (db10)

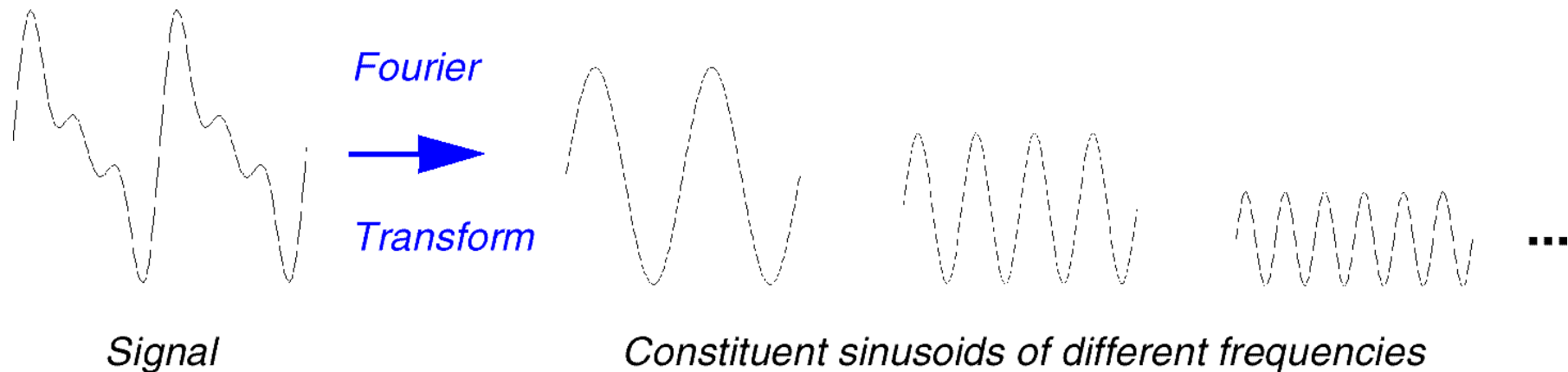


■ 从傅里叶变换到小波变换 (wavelet transforms)

➤ 傅里叶变换

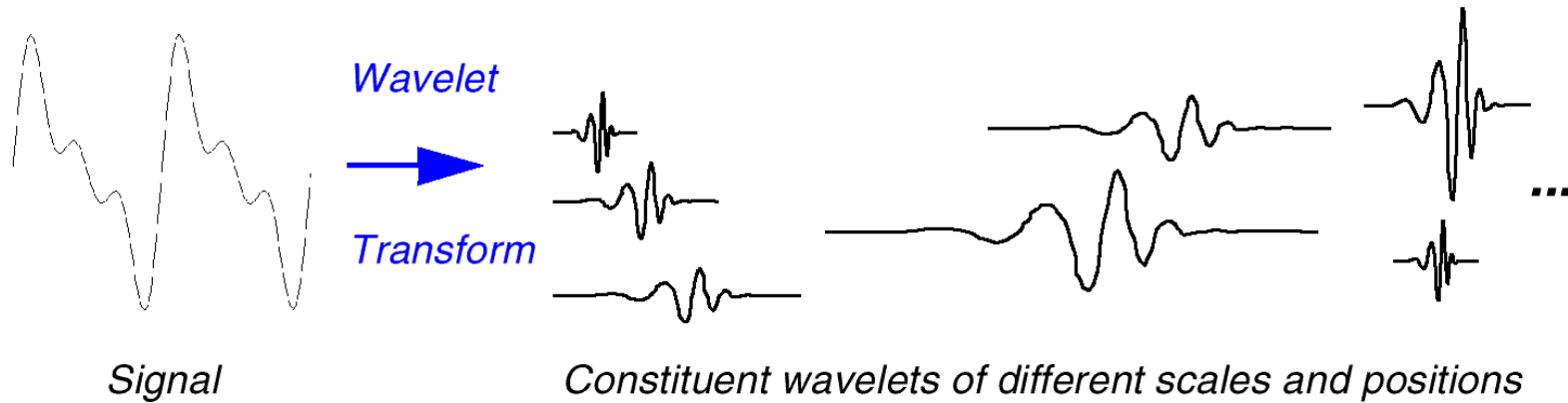
$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

- ✓ 意味着信号在**所有时间区间**与复指数相乘，结果产生傅里叶系数 $F(\omega)$
- ✓ 按照傅里叶系数，可将原信号分解为不同频率的组合



➤ 小波变换

- ✓ 结果产生小波变换系数
- ✓ 按照小波系数，原始信号分解为不同小波的组合



- ✓ 傅里叶变换将信号分解成一系列不同频率正弦波的叠加，小波变换是将信号分解成一系列小波函数的叠加
- ✓ 这些小波函数都是由一个母小波函数经过平移与尺度伸缩得来的

■ 从傅里叶变换到小波变换

➤ 小波变换步骤:

面对高低频信号
窗口大小也不同

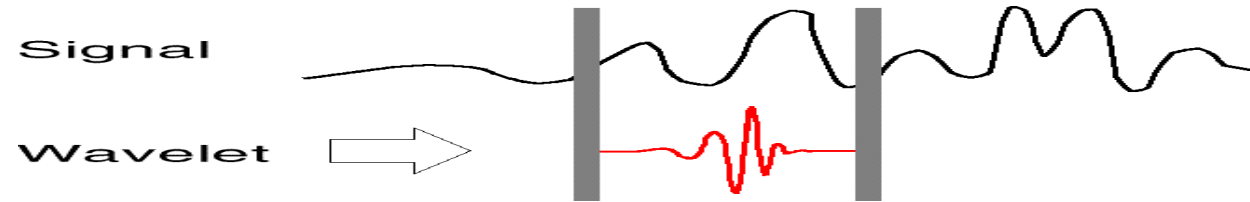
第一步



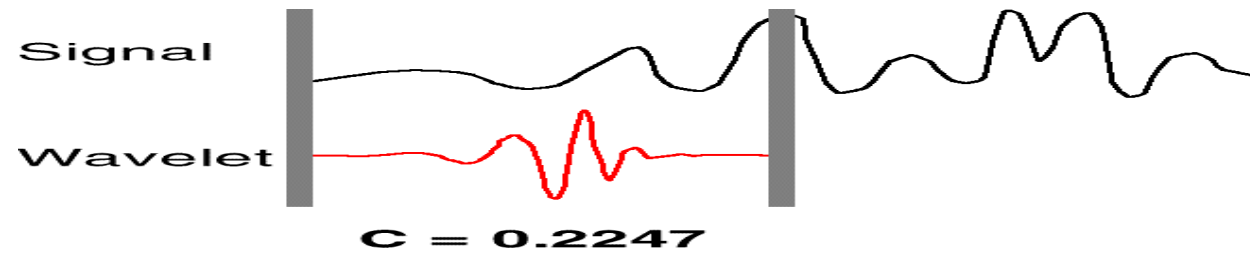
第二步



第三步



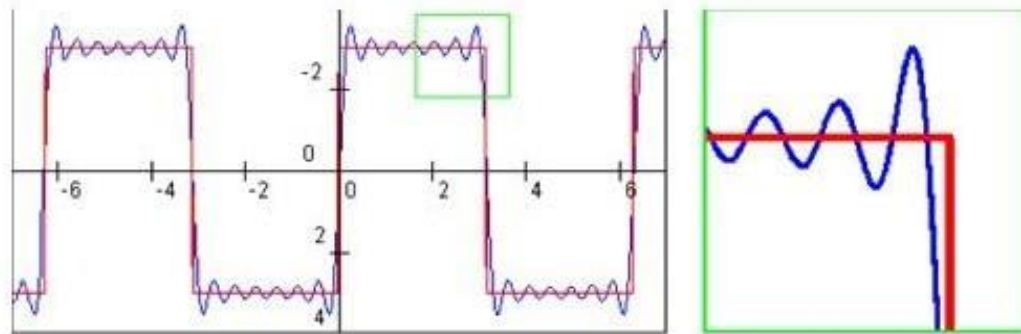
第四步



第☒步: 对所有尺度的小波重复一至四步

对于突变信号

- ✓ 用不规则的小波函数来逼近尖锐变化的信号显然要比光滑的正弦曲线要好
- ✓ 信号局部的特性用小波函数来逼近显然要比光滑的正弦函数来逼近效果好



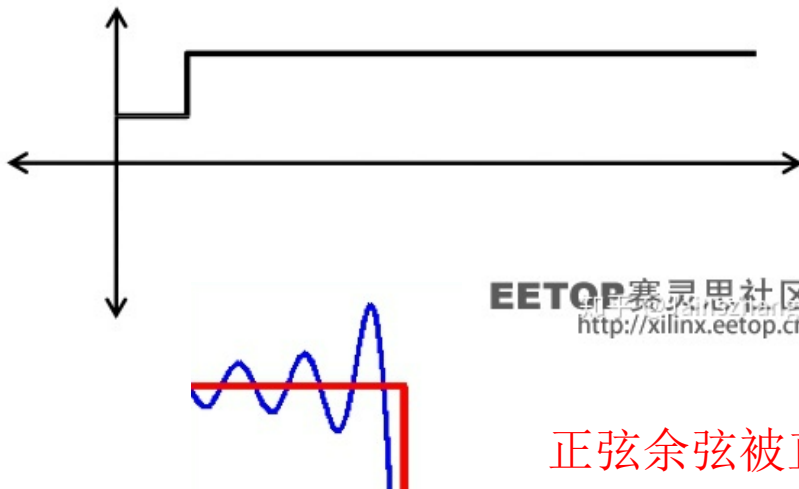
对于这种变换突然而剧烈的信号，即使只有一小段变换，傅立叶也不得不用大量的三角波去拟合。

——吉布斯效应



用傅立叶将其展开，只有一个级数系数不是0，其他所有级数系数都是0

EETOP赛灵思社区
<http://xilinx.eetop.cn>

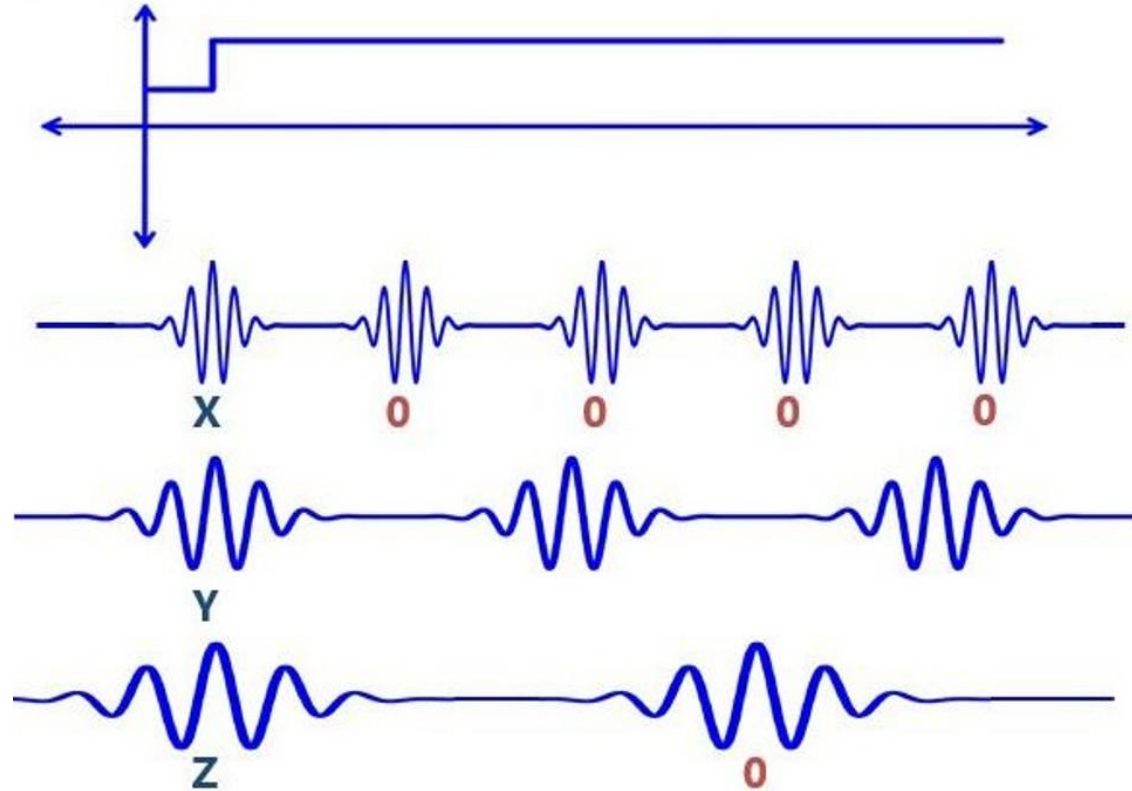


在前一个直流信号上，增加了一个突变。但是，所有的傅立叶级数都为非0了！因为傅立叶必须用三角波来展开信号，对于这种变换突然而剧烈的信号来讲，即使只有一小段变换，傅立叶也不得不用大量的三角波去拟合

EETOP赛灵思社区
<http://xilinx.eetop.cn>

正弦余弦被直角支配的恐怖

只有小波函数与信号突变处重叠时，系数不为0



上图中存在三个基， x,y,z ，可以看出在纵向数值方面， x,y,z 的叠加正好可以满足最初的信号，在断崖处也差不多可以满足。这种表示只需要3个系数即可，相比傅里叶转换大幅降低了计算需要



- 由于小波基的特殊性，任何小波和常量函数的内积都趋近于0。基于小波的设计特性，就需要保证母小波在一个周期的积分趋近于0。因此，小波变换允许更加精确的局部描述以及信号特征的分离。一个傅立叶系数通常表示某个贯穿整个时间域的信号分量，即使是临时的信号，其特征也被强扯到了整个时间周期去描述。而小波展开的系数则代表了对应分量它当下的自己，因此非常容易诠释。
- 对于傅立叶变换以及大部分的信号变换系统，他们的基函数都是固定的正弦波，那么变换后的结果只能按部就班被分析推导出来，没有任何灵活性。总之你就只能用正弦波，没有任何商量的余地。
- 而小波变换的基函数是可变的，可以根据信号来推导或者构建出来，只要符合小波变换的性质和特点即可。这种灵活性是任何别的变换都无法比拟的。总结来说，傅立叶变换适合周期性的，统计特性不随时间变化的信号；而小波变换则适用于大部分信号，尤其是瞬时信号。它针对绝大部分信号的压缩，去噪，检测效果都特别好。

■ 连续小波变换 (continuous wavelet transform)

➤ 基本小波

- ✓ 基本小波是一具有特殊性质的实值函数，它是震荡衰减的，而且通常衰减得很快，在数学上满足积分为零的条件：

$$\int_{-\infty}^{\infty} \psi(t) dt = 0$$

- ✓ 同时其频谱 $\Psi(\omega) = \int_{-\infty}^{\infty} \psi(t) e^{-j\omega t} dt$

- ✓ 满足条件：
$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty$$

两个条件可概括为：**小波应是一个具有振荡性和迅速衰减的波**



■ 连续小波变换

➤ 小波基函数

- ✓ 通过对基本小波进行尺度上的伸缩和位置上的移动，可形成一系列小波函数——**小波基函数**
- ✓ 一组小波基函数是通过尺度因子和位移因子由基本小波来产生：

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right)$$

式中：a—**尺度系数**（伸缩系数）；b—**位移系数**

■ 连续小波变换

▶ 连续小波变换定义（又称之为积分小波变换）：

$$\begin{aligned} W_f(a, b) &= \langle f, \psi_{a,b}(t) \rangle = \\ &= \int_{-\infty}^{\infty} f(t) \psi_{a,b}(t) dt = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \end{aligned}$$

小波变换是信号与被缩放和平移的小波函数之积在信号存在的整个期间里求和的结果

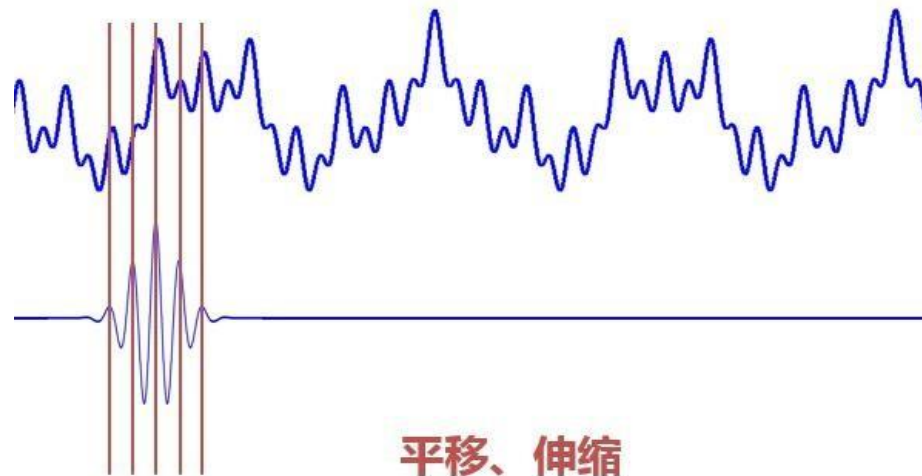
▶ 连续小波变换的逆变换为：

$$f(t) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_f(a, b) \psi_{a,b}(t) db \frac{da}{a^2}$$

- 傅里叶变换，变量只有频率 ω
- 小波变换有两个变量：
 - 尺度 a (scale) 和平移量 b (translation)
 - 尺度 a 控制小波函数的伸缩
 - 平移量 b 控制小波函数的平移
 - 尺度就对应于频率（反比），平移量 b 就对应于时间。

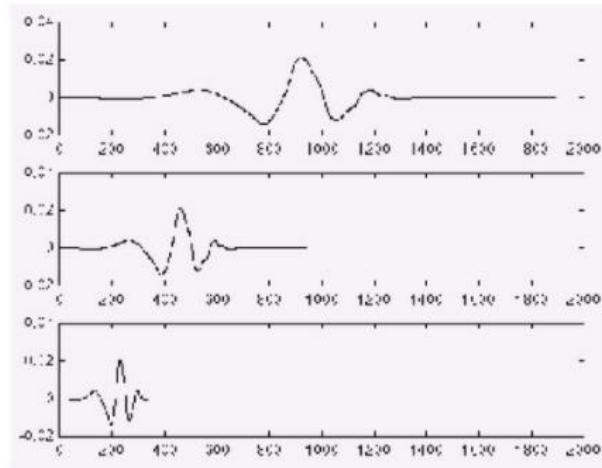
$$WT(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi\left(\frac{t-\tau}{a}\right) dt$$

小波变换结果是许多小波系数WT，这些系数是缩放因子（尺度）和平移的函数



平移、伸缩

- 缩放，就是压缩或伸展基本小波，缩放系数越小，则小波越窄，对应的频率越高



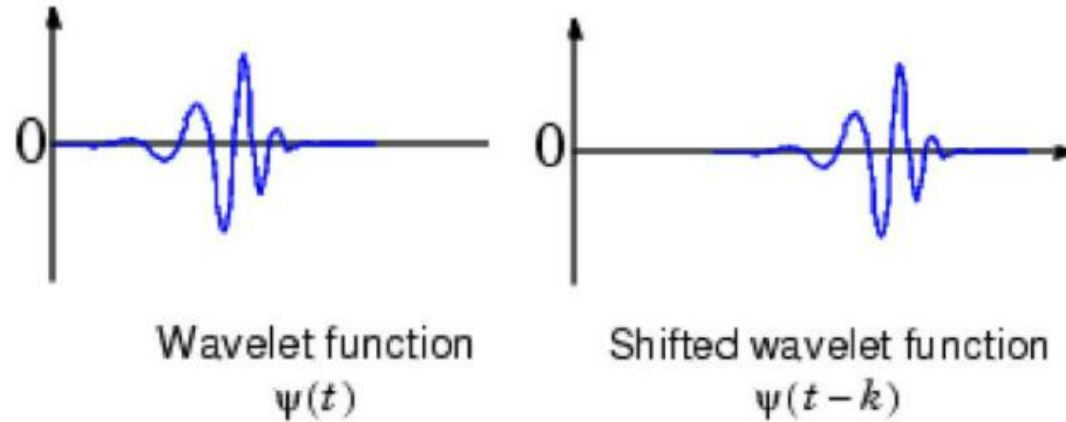
$$f(t) = \psi(t); a = 1$$

$$f(t) = \psi(2t); a = \frac{1}{2}$$

$$f(t) = \psi(4t); a = \frac{1}{4}$$

- 小尺度 a → 压缩的小波 → 快速变换的细节 → 高频部分
- 大尺度 a → 拉伸的小波 → 缓慢变换的粗部 → 低频部分

- 平移，小波的延迟或超前。在数学上，函数 $f(t)$ 延迟 k 的表达式为 $f(t-k)$



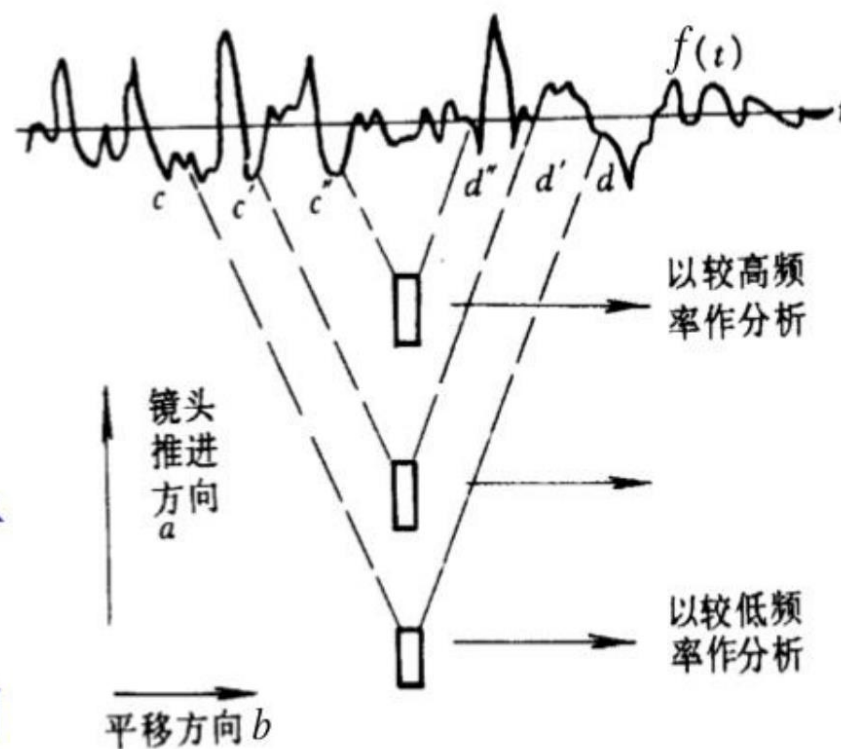
任何复杂的信号 $f(t)$ ，都能由小波的母函数经过伸缩和平移产生的基底的线性组合表示



- 与傅里叶变换相比，小波变换是空间（时间）和频率的局部变换，它通过伸缩平移运算对信号逐步进行多尺度细化，最终达到高频处时间细分，低频处频率细分，能自动适应时频信号分析的要求，从而可聚焦到信号的任意细节，所以被称为数学显微镜
- 打个比喻，我们用镜头观察目标信号 $f(t)$ ， $w(t)$ 代表镜头所起的作用。 b 相当于使镜头相对于目标平行移动， a 的作用相当于镜头向目标推进或远离

多尺度/多分辨率，可以由粗到细地处理信号

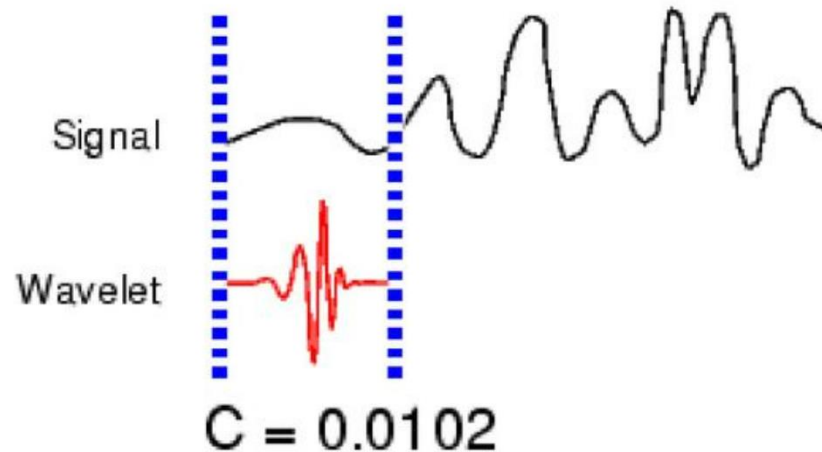
- 用镜头观察目标 $f(t)$ (待分析信号)。
- $\psi(t)$ 代表镜头所起的作用(如滤波或卷积)。
- b 相当于使镜头相对于目标平行移动。
- a 的作用相当于镜头向目标推进或远离。



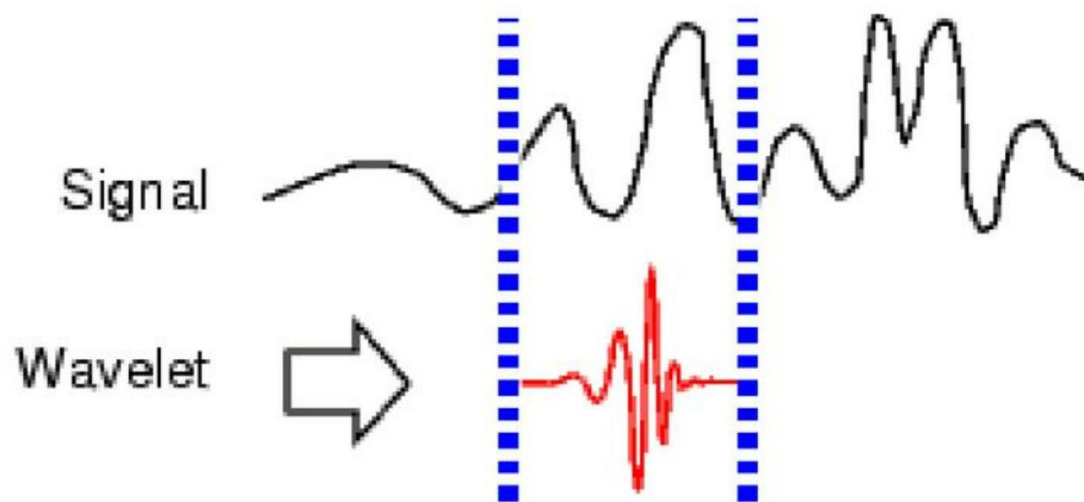
小波变换的粗略解释

■ 小波变换步骤

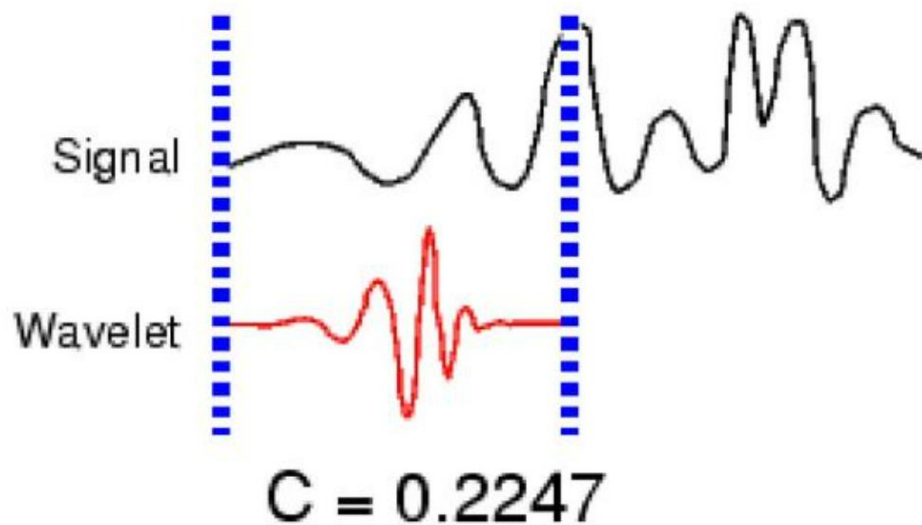
- (1) 选择一个小波函数，并将这个小波与要分析的信号起始点对齐；
- (2) 计算在这一时刻要分析的信号与小波函数的逼近程度，即计算小波变换系数 C ， C 越大，就意味着此刻信号与所选择的小波函数波形越相近，如图所示。



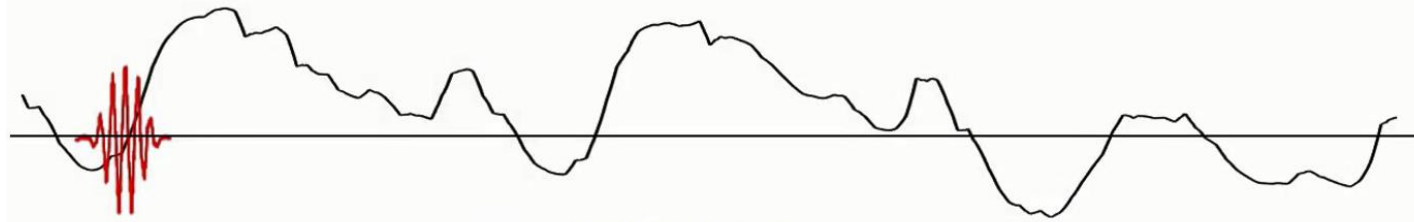
(3) 将小波函数沿时间轴向右移动一个单位时间，然后重复步骤(1)、(2)求出此时的小波变换系数 C ，直到覆盖完整个信号长度，如图所示；



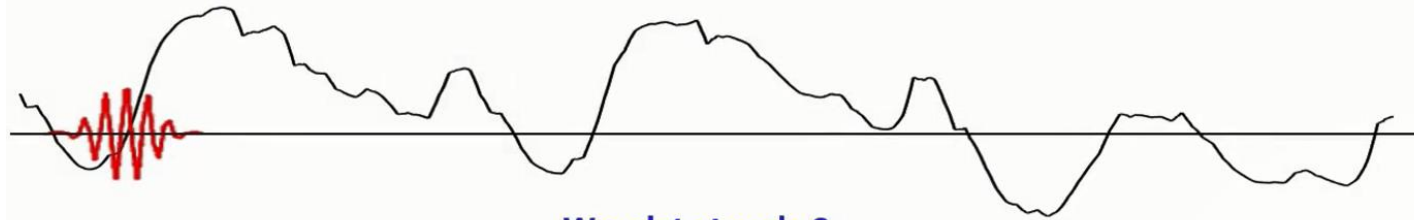
(4) 将所选择的小波函数尺度伸缩一个单位，然后重复步骤(1)、(2)、(3)，如图所示；



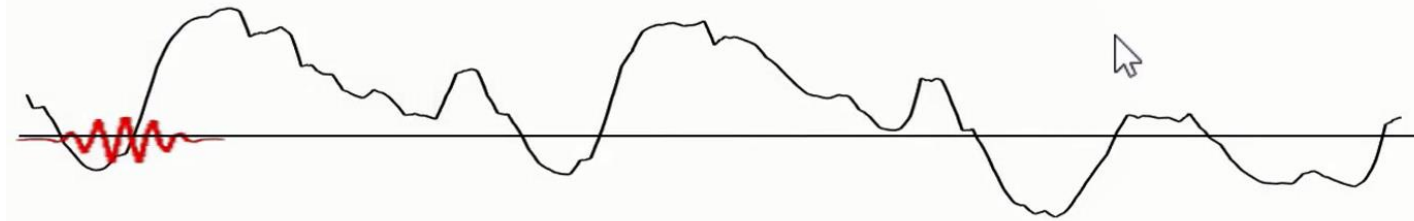
(5) 对所有的尺度伸缩重复步骤(1)、(2)、(3)、(4)。



Wavelet at scale 1



Wavelet at scale 2



Wavelet at scale 3

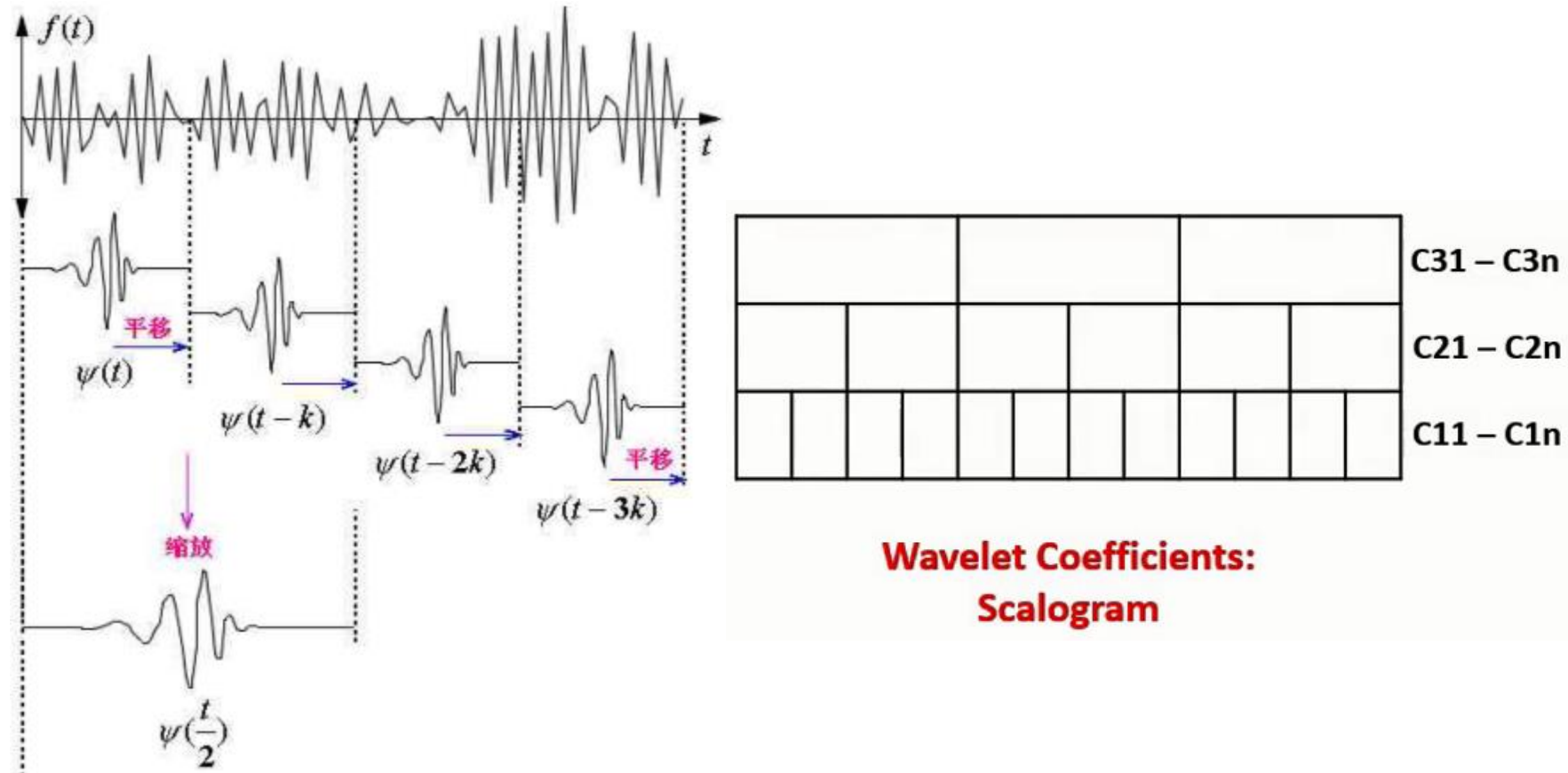
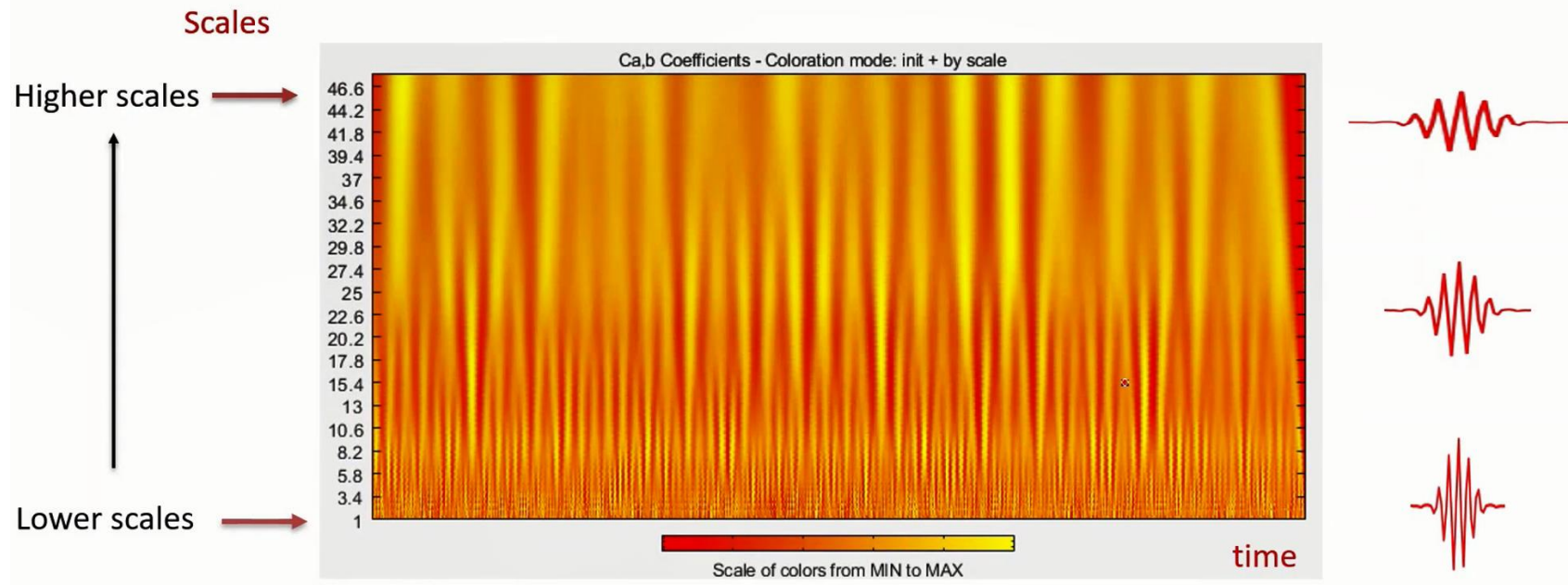
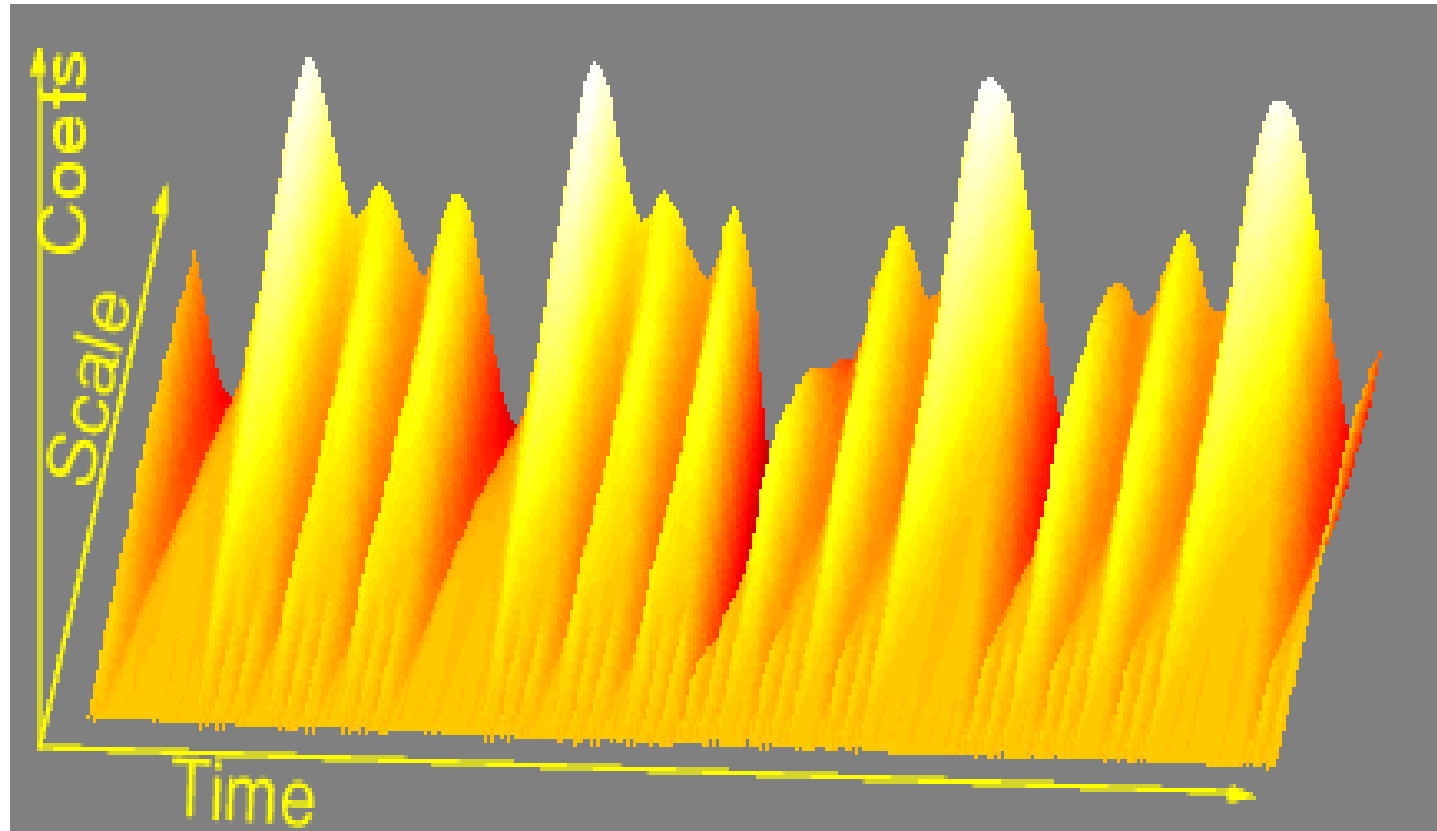


图1-4 连续小波变换的过程

Scalogram: Wavelet Coefficients Representation



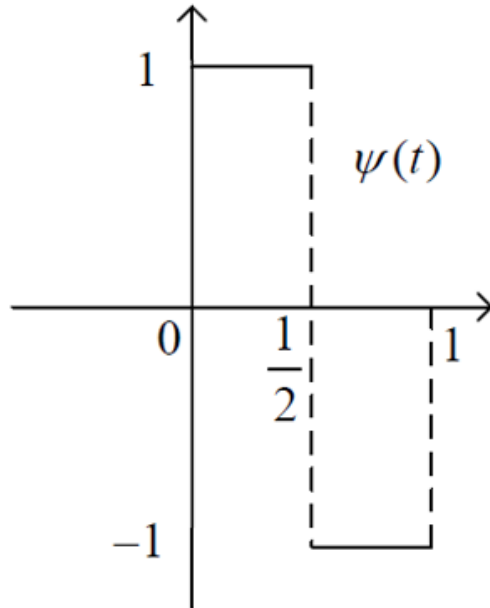
➤ 小波变换系数分布图



常用基本小波

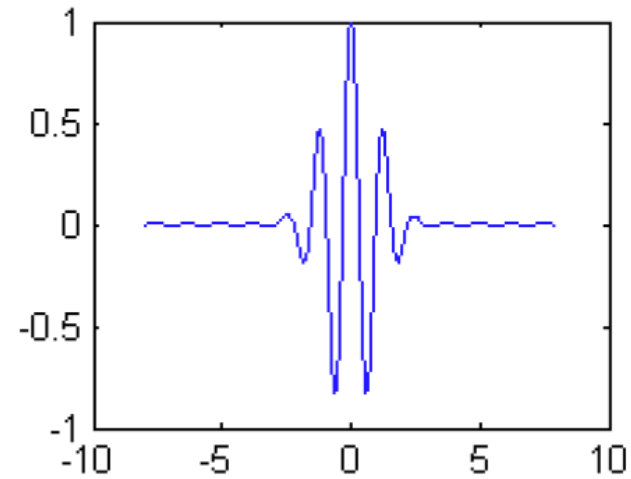
■ 哈尔小波

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{其它} \end{cases} \quad \hat{\psi}(\omega) = i \frac{4}{\omega} e^{-i\omega/2} \sin^2(\omega/4)$$



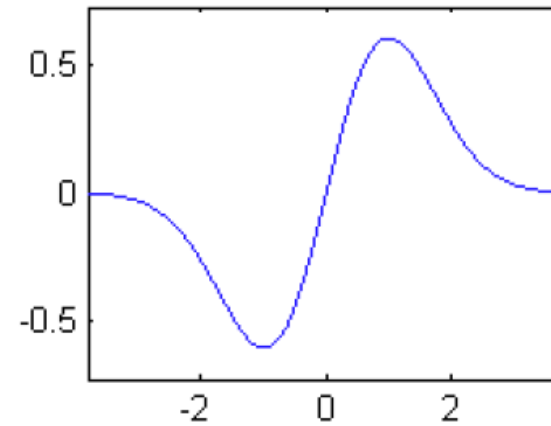
- Morlet小波

$$\psi(t) = e^{-t^2/2} e^{i\omega_0 t}$$



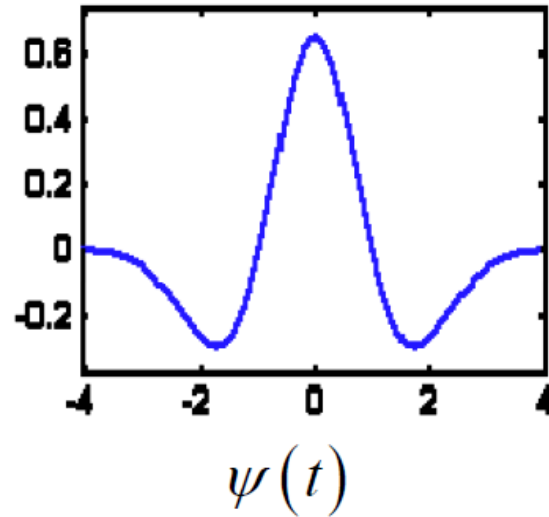
- 高斯小波

$$\psi(t) = -\frac{1}{\sqrt{2\pi}} t e^{-t^2/2}$$



- Marr小波 (墨西哥草帽小波)

$$\psi(t) = \frac{2}{\sqrt{3}\sqrt{\pi}} (1-t^2)e^{-t^2/2}$$



■ 离散小波变换

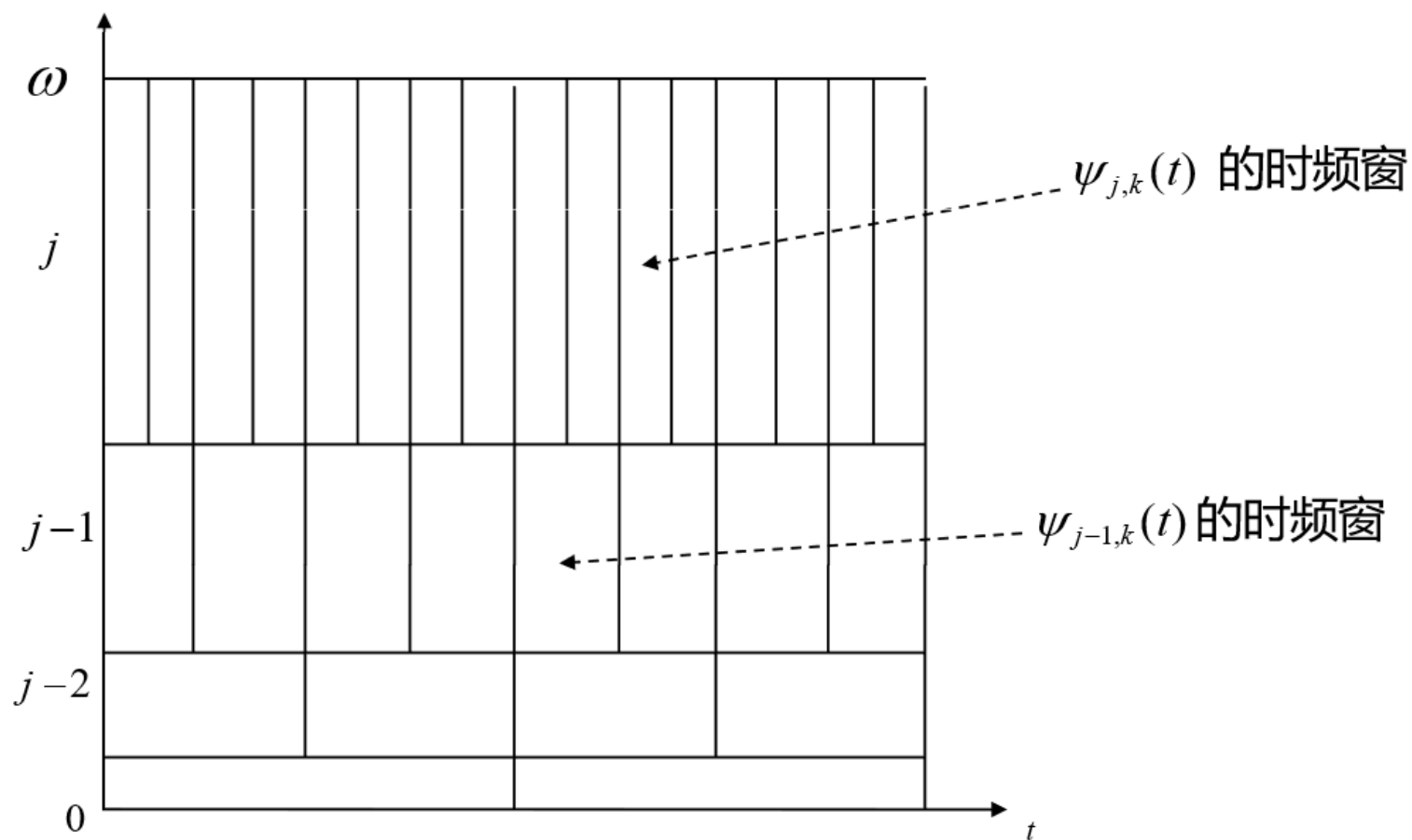
- 连续小波变换中，尺度系数和平移系数连续取值，将产生巨大的计算量，主要用于理论分析
- 仅取尺度与位置的某些离散量，采用离散化的尺度及位移因子，可大量减少计算量，形成离散小波变换
 - ✓ 令 $a = a_0^m$; $b = nb_0 a_0^m$; $a_0 > 1, b_0 \neq 0$; m, n 为整数系列

则有离散小波基函数

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \Psi\left(\frac{t - nb_0 a_0^m}{a_0^m}\right) = a_0^{-\frac{m}{2}} \Psi(a_0^{-m}t - nb_0)$$

及离散小波变换:

$$\langle f, \Psi_{m,n} \rangle = \int_{-\infty}^{\infty} f(t) \Psi_{m,n}(t) dt = a_0^{-\frac{m}{2}} \int_{-\infty}^{\infty} f(t) \Psi(a_0^{-m}t - nb_0) dt$$



离散小波分解自适应时-频窗示意图

■ 二进小波变换

- 若基于2的幂次方选择二进伸缩和二进位移（以2的因子伸缩和平移）构成基函数，即

$$a_0 = 2; \quad b_0 = 1;$$

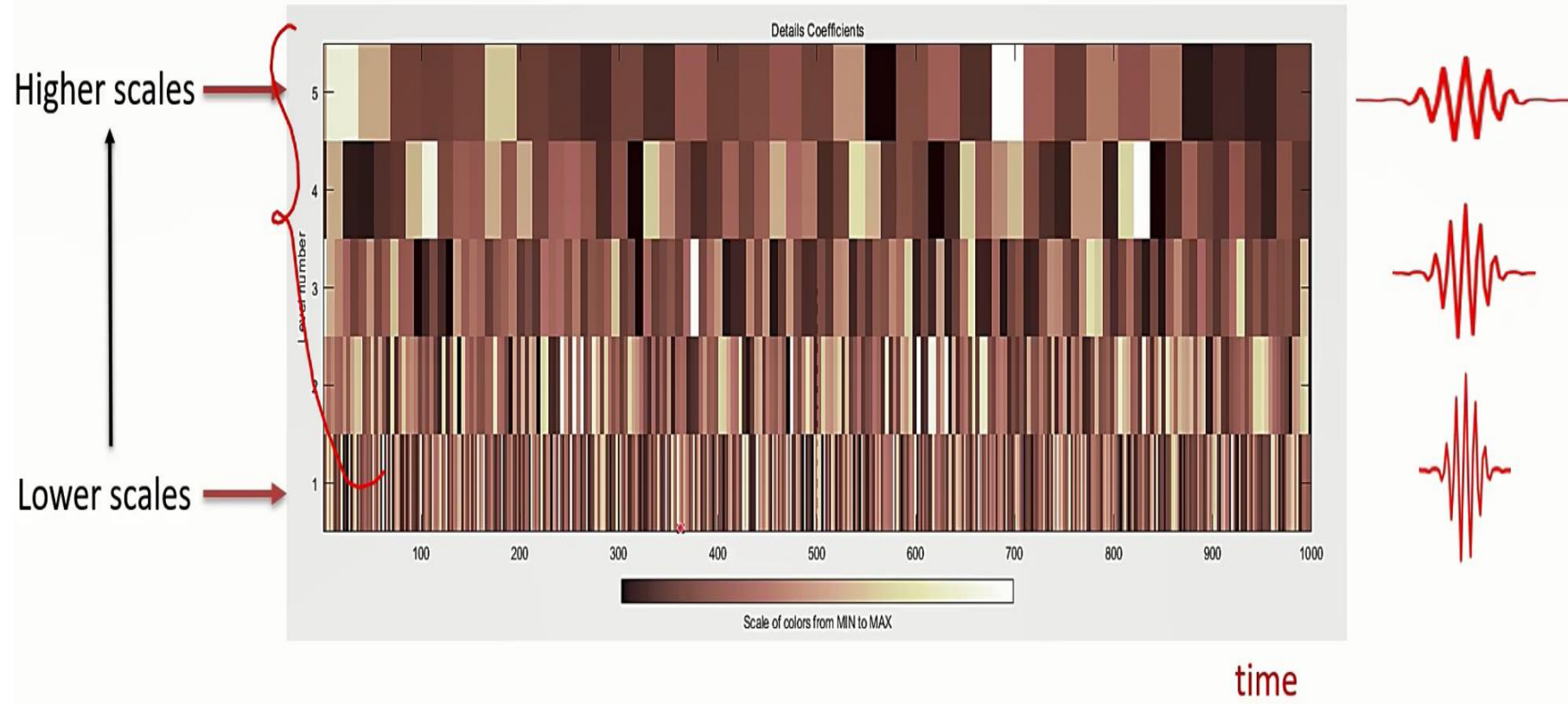
- 则形成二进小波

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}} \psi\left(\frac{t - n2^m}{2^m}\right) = 2^{-\frac{m}{2}} \psi(2^{-m}t - n)$$

- 满足下列条件的二进小波（正交性条件）

$$\langle \psi_{m,n}, \psi_{j,k} \rangle = \delta_{m,j} \delta_{n,k} \quad (\text{Kronecher } \delta \text{ 函数})$$

$$= \begin{cases} 1 & m = j, n = k \\ 0 & \text{其他} \end{cases}$$



■ 二进正交小波变换

- 由二进正交小波可得到信号的任意精度的近似表示：

$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{m,n} \psi_{m,n}(t)$$

- 其中变换系数：

$$c_{m,n} = \langle f(t), \psi_{m,n}(t) \rangle = 2^{-m/2} \int_{-\infty}^{\infty} f(t) \psi(2^{-m}x - n) dt$$

为小波级数展开式

- 快速小波变换算法（**FWT**，**Mallat**算法—塔式算法）

- 回忆线性系统输出表达式



$$y[n] = x[n] * h[n] = h[n] * x[n]$$

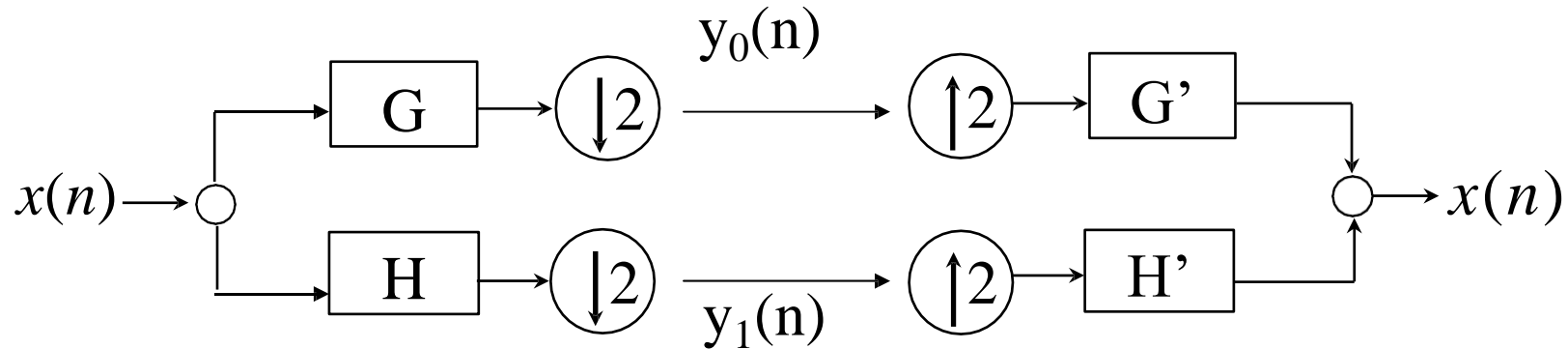
$$= \sum_{k=1}^N x[k] \cdot h[n-k]$$

$$= \sum_{k=1}^N h[k] \cdot x[n-k]$$

- 系统输出相当于对输入信号的滤波

■ 快速小波变换算法

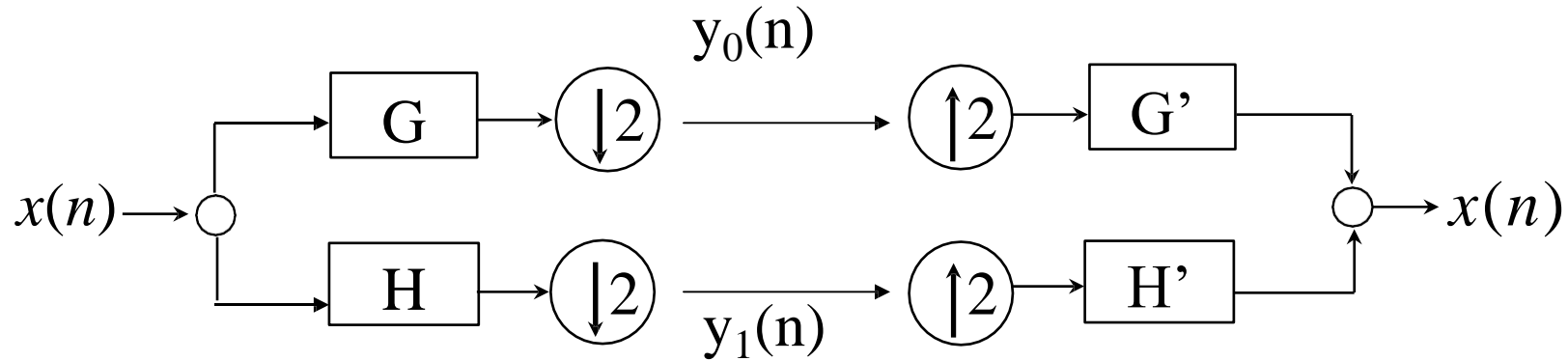
- **1976年Croisier**为实现语音信号压缩，将一个离散信号通过两个滤波器滤波并通过二选一的采样，发现滤波器如果满足一定条件，信号可以重建



- 该发现深刻影响了后面十年滤波器组的理论研究
- **1988年，S.Mallat**在构造正交小波基时提出了多分辨率分析（**Multi-Resolution Analysis**）概念，形象地说明了小波的多分辨率特性，并给出了正交小波变换的快速算法，即**Mallat**算法

■ 快速小波变换算法

- 对滤波器进行分解，形成一对共轭正交滤波器组，可使下述分解与重构后的信号与原始信号完全相同



- 两个滤波器必须满足条件

$$G^2(s) + H^2(s) = 1, \quad 0 \leq |s| \leq s_N$$

- 小波理论发现共轭镜像滤波器完全刻画了标准正交小波

■ 快速小波变换算法

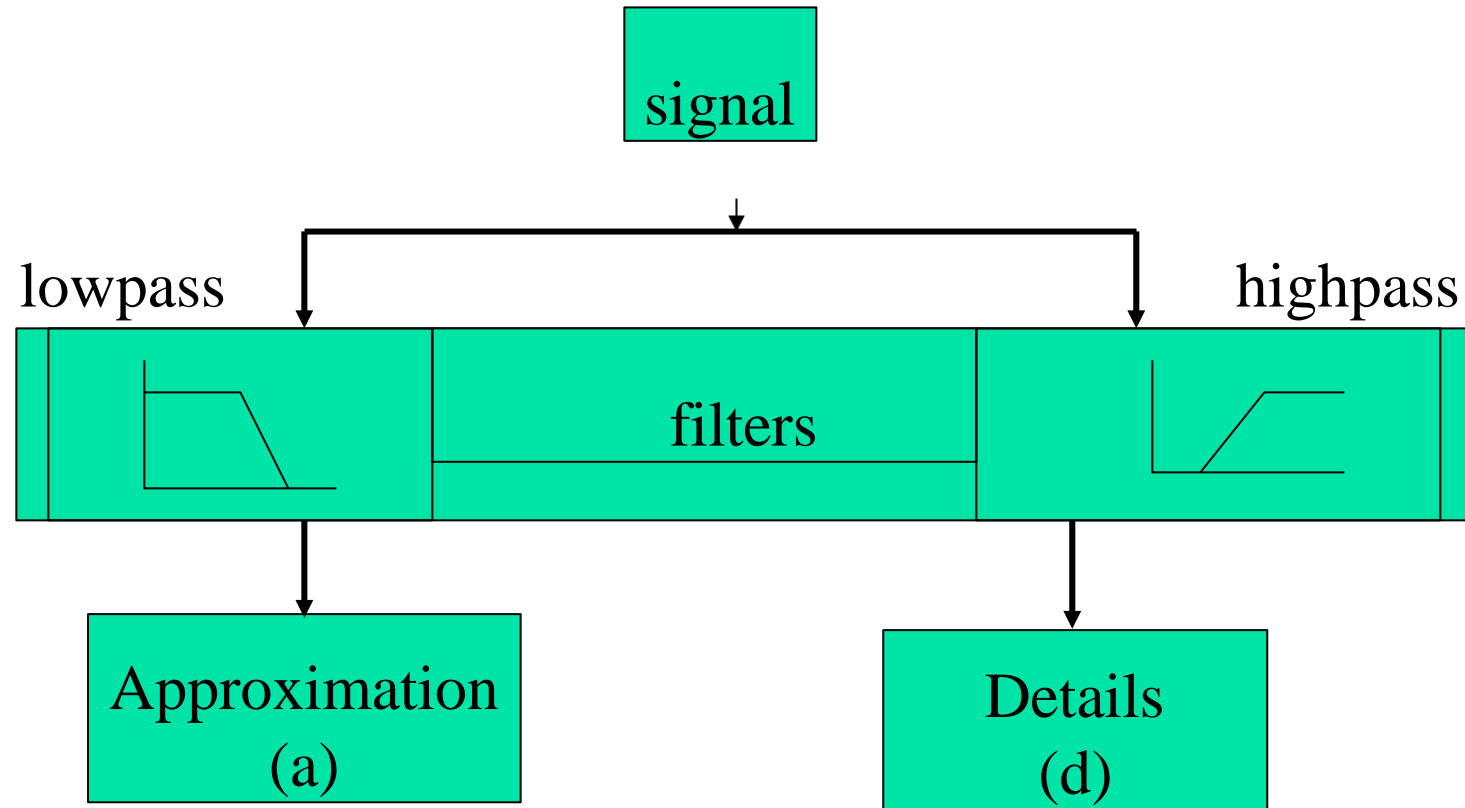
- 假定 $H(s)$ 为小波变换中使用的具有平滑边沿的低通滤波函数，则 $G(s)$ 相应为：

$$G^2(s) = 1 - H^2(s)$$

- 信号通过 $H(s)$ 和 $G(s)$ 后，相当于信号分解为低频部分 y_{low} （粗分量、平滑部分），与高频部分 y_{high} （细分量、细节部分）

设计一个离散小波变换的核心，就是如何选择低通滤波器 $H(s)$

■ 快速小波变换算法

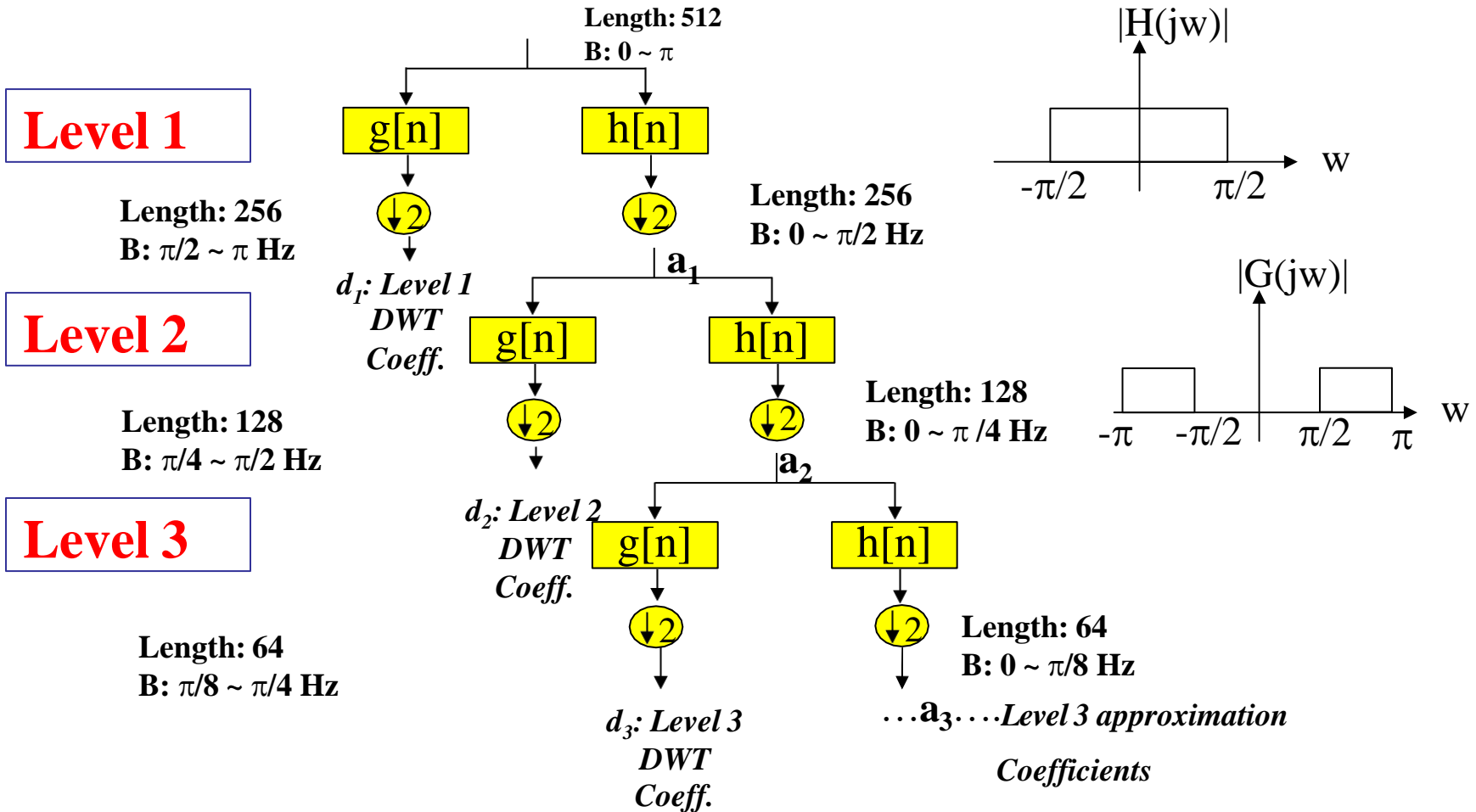




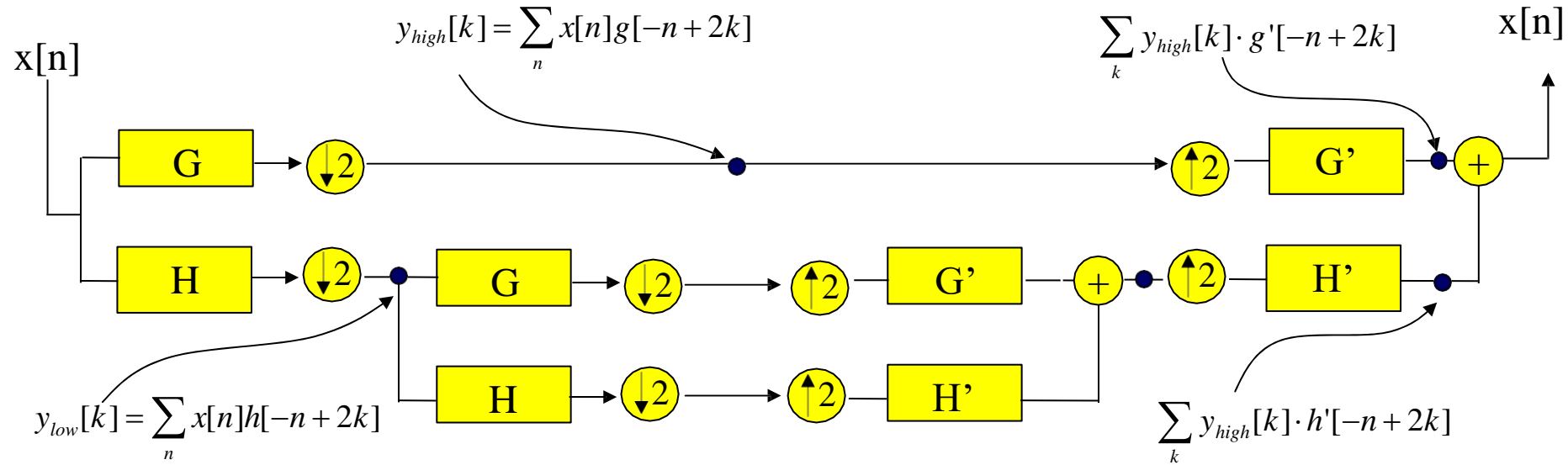
■ 快速小波变换算法

- 原始信号经过一对互补的滤波器组进行的分解称为一级分解；
- 理论上，可以用一对互补的滤波器组对前述分解后的信号继续进行分解
- 如果对信号的高频分量不再分解，而对低频分量进行连续分解，就可以得到信号不同分辨率下的高频分量和低频分量，这也称之为信号的多分辨率分析
- 连续的分解将形成一棵比较大的分解树，称其为信号的小波分解树（Wavelet Decomposition Tree）
- 实际中，分解的级数取决于要分析的信号数据特征及分析需求
- 形成FWT算法的基本思想

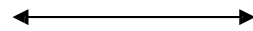
快速小波变换算法



快速小波变换算法



Decomposition

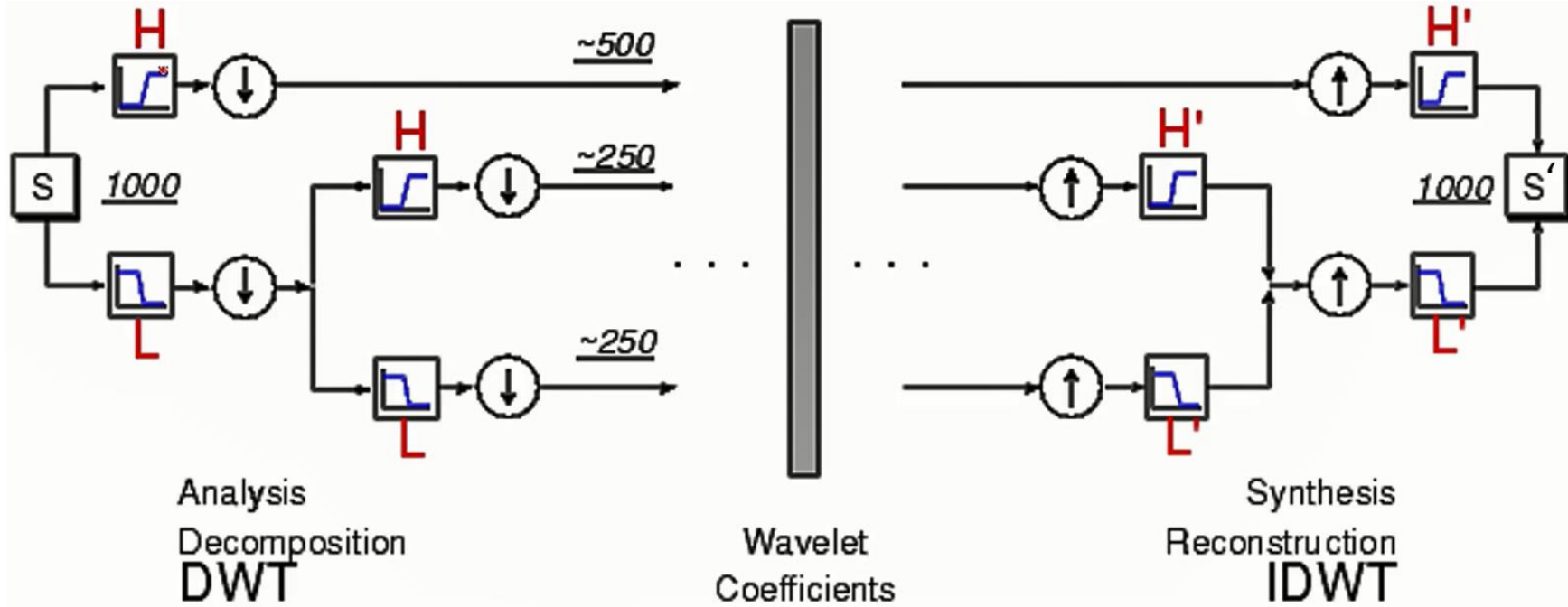


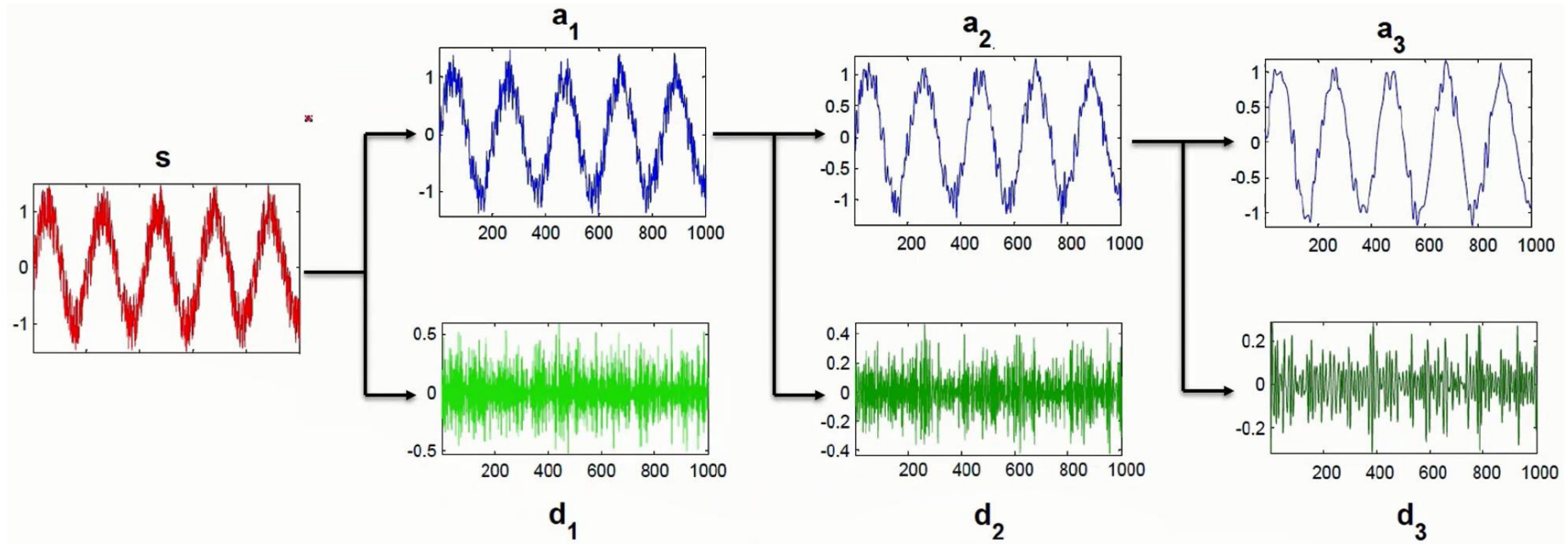
Reconstruction

G Half band high pass filter
 H Half band low pass filter

$\downarrow 2$ Down-sampling
 $\uparrow 2$ Up-sampling

Multistep Decomposition and Reconstruction





■ 二维小波变换

➤ 二维连续小波定义

$$\Psi_{a,b_x,b_y}(x,y) = \frac{1}{|a|} \Psi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right)$$

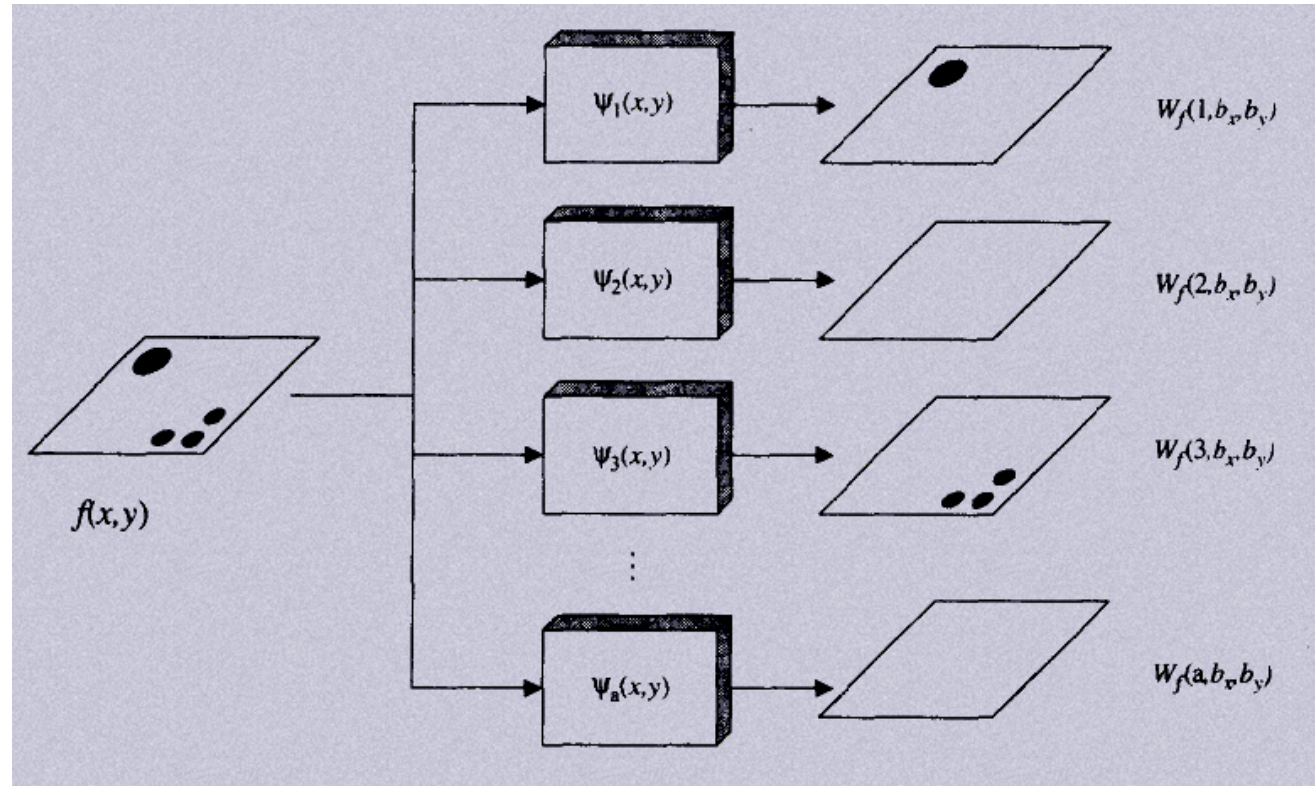
➤ 二维连续小波变换

$$W_f(a,b_x,b_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \Psi_{a,b_x,b_y}(x,y) dx dy$$

➤ 二维连续小波逆变换

$$f(x,y) = \frac{1}{C_\Psi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(a,b_x,b_y) \Psi_{a,b_x,b_y}(x,y) db_x db_y \frac{da}{a^3}$$

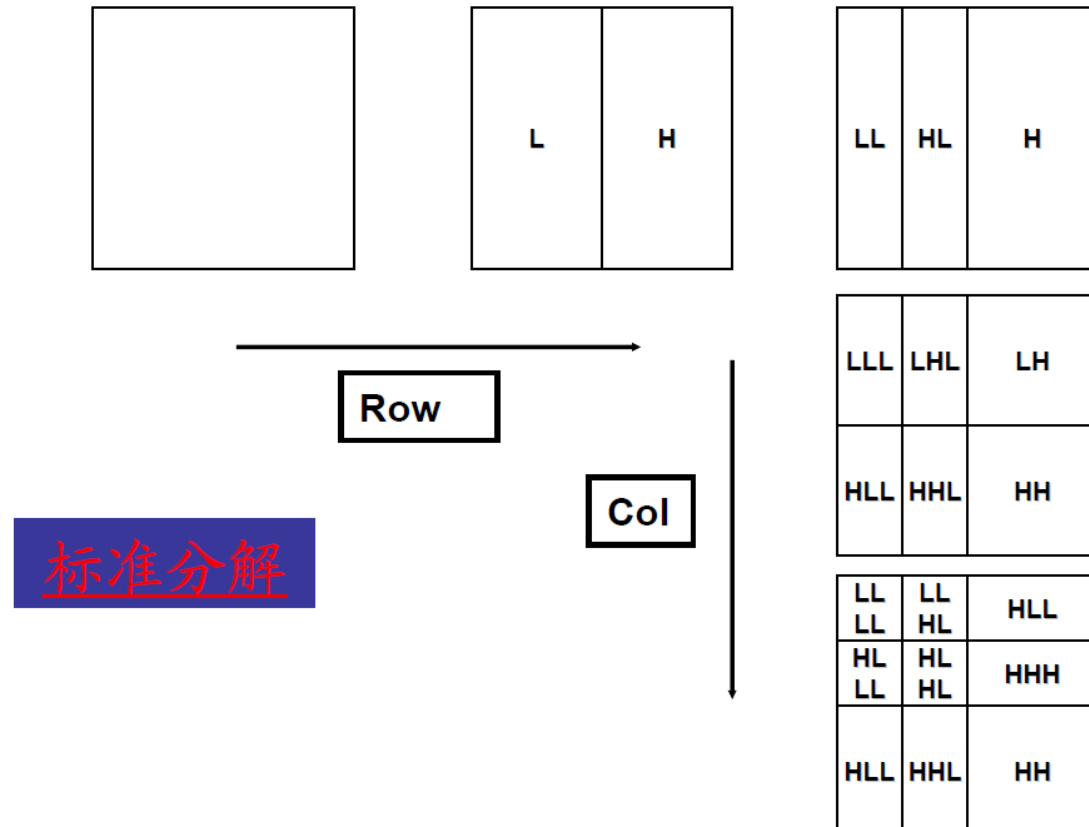
■ 二维小波变换的滤波器解释

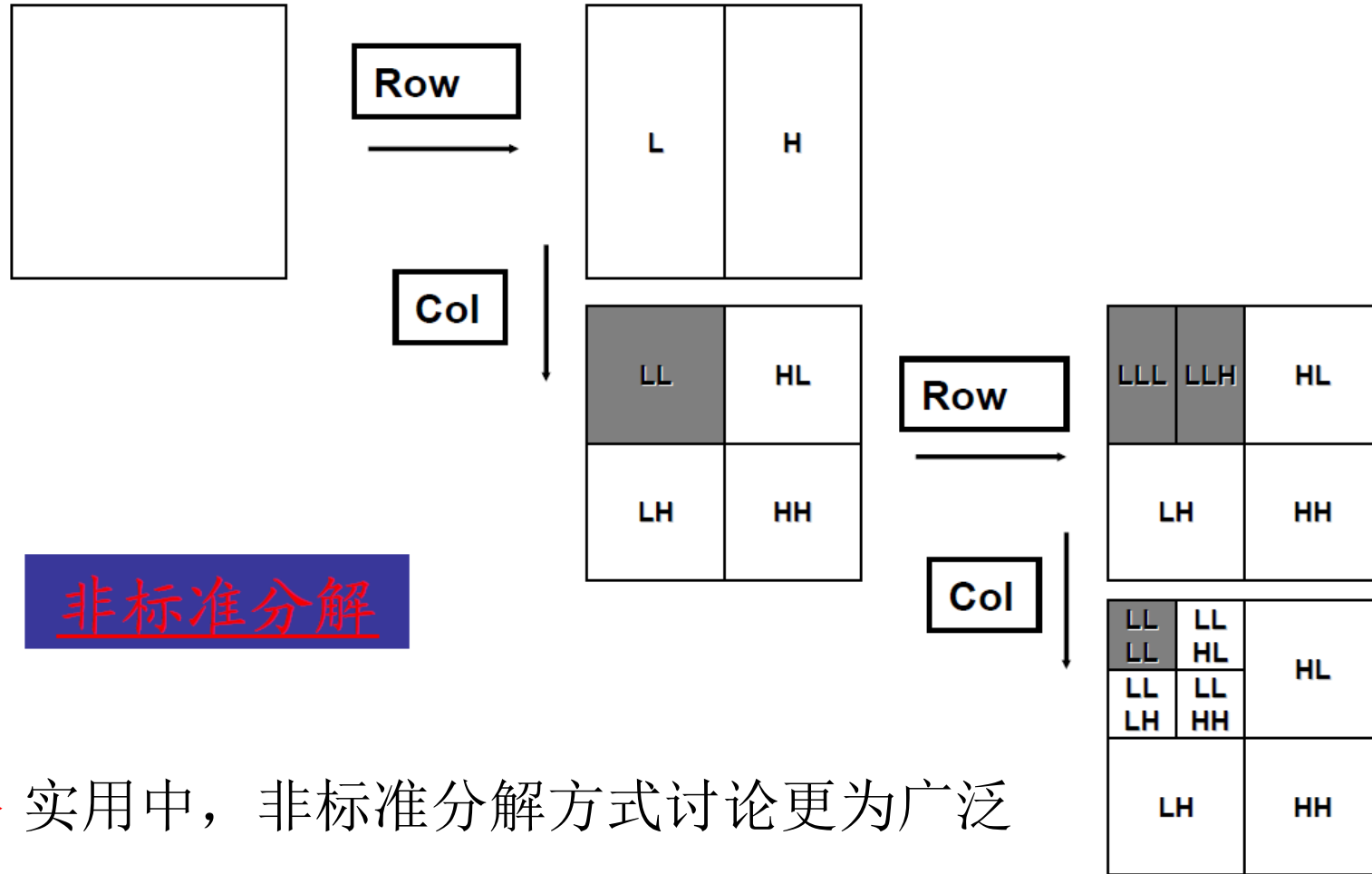


- 每一滤波器都是一个二维冲激响应，输入是图象上的带通滤波器，滤波后的图象的叠层组成了小波变换

■ 二维离散小波变换

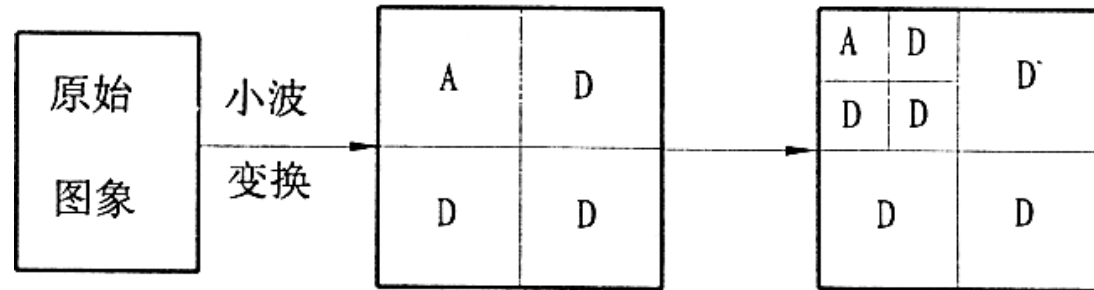
➤ 与一维分解类似，可以进行不同层次的分解



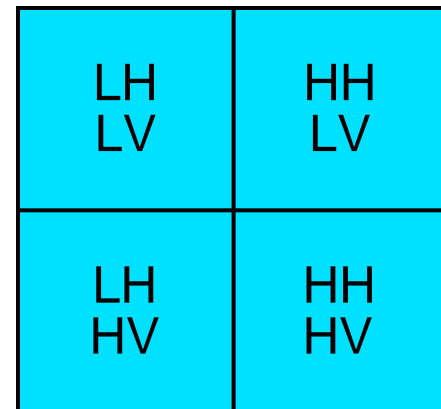


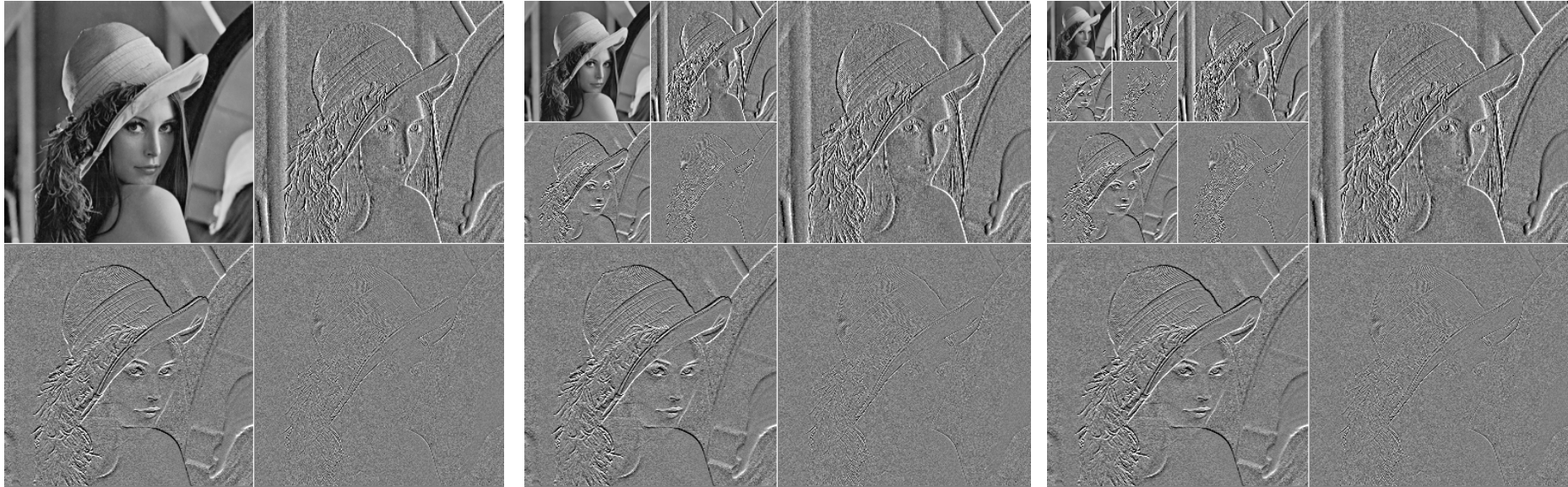
■ 二维离散小波变换

- 每一层次的分解，分别形成一个平滑子图（低频分量）和三个细节子图（高频分量）



- 深入考察二维小波基，细节子图又可进一步分解为垂直细节、水平细节的组合





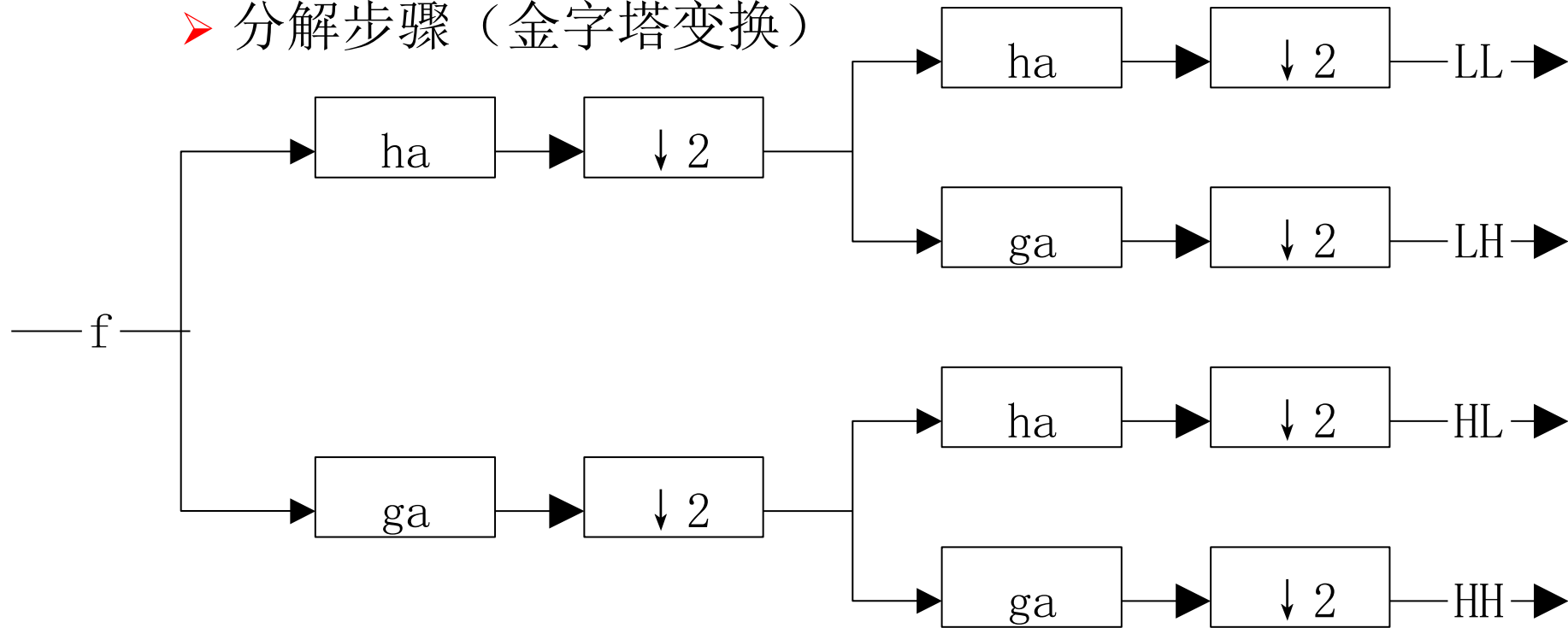
Level 1

Level 2

Level 3

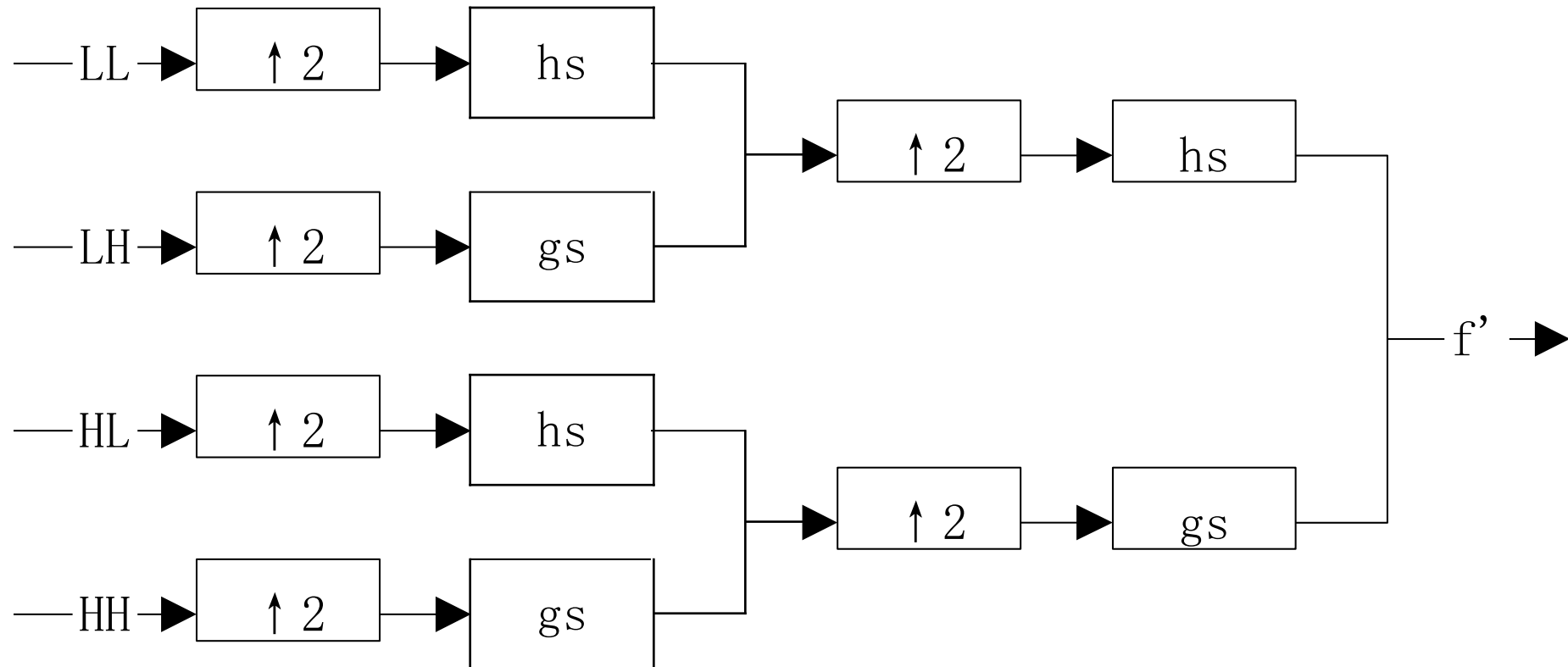
■ 二维离散小波变换

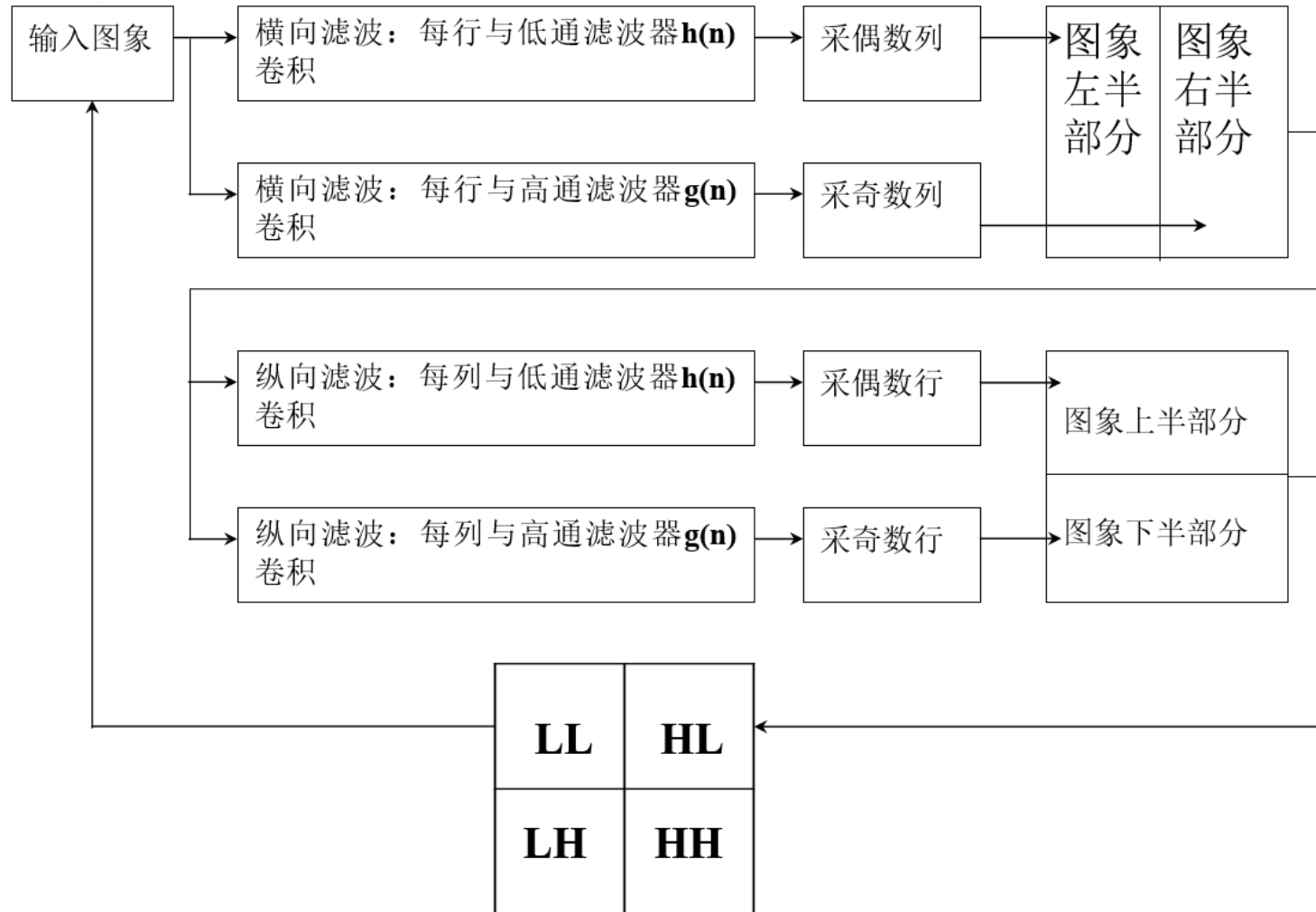
➤ 分解步骤（金字塔变换）



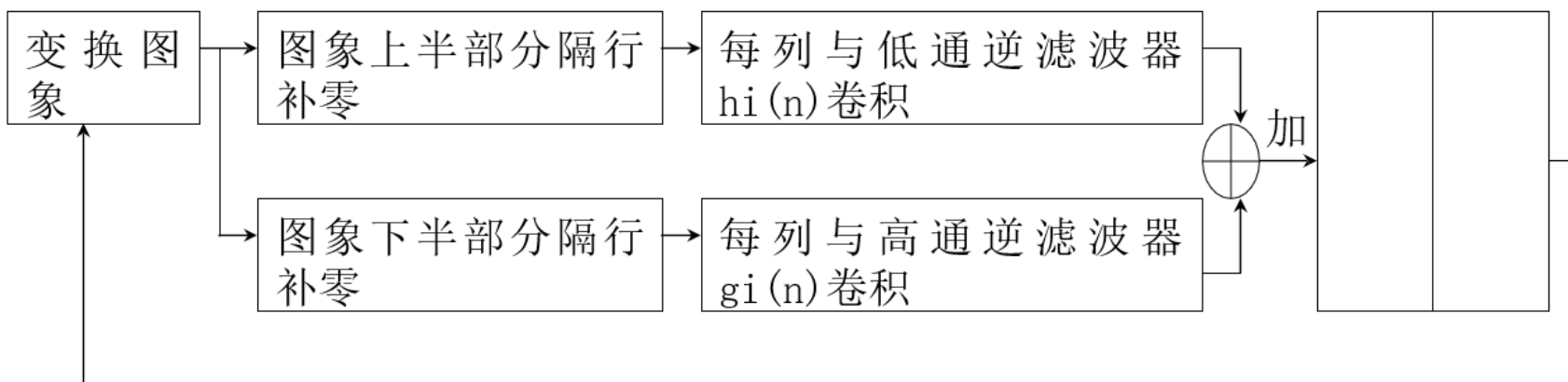
■ 二维离散小波变换

➤ 重构过程



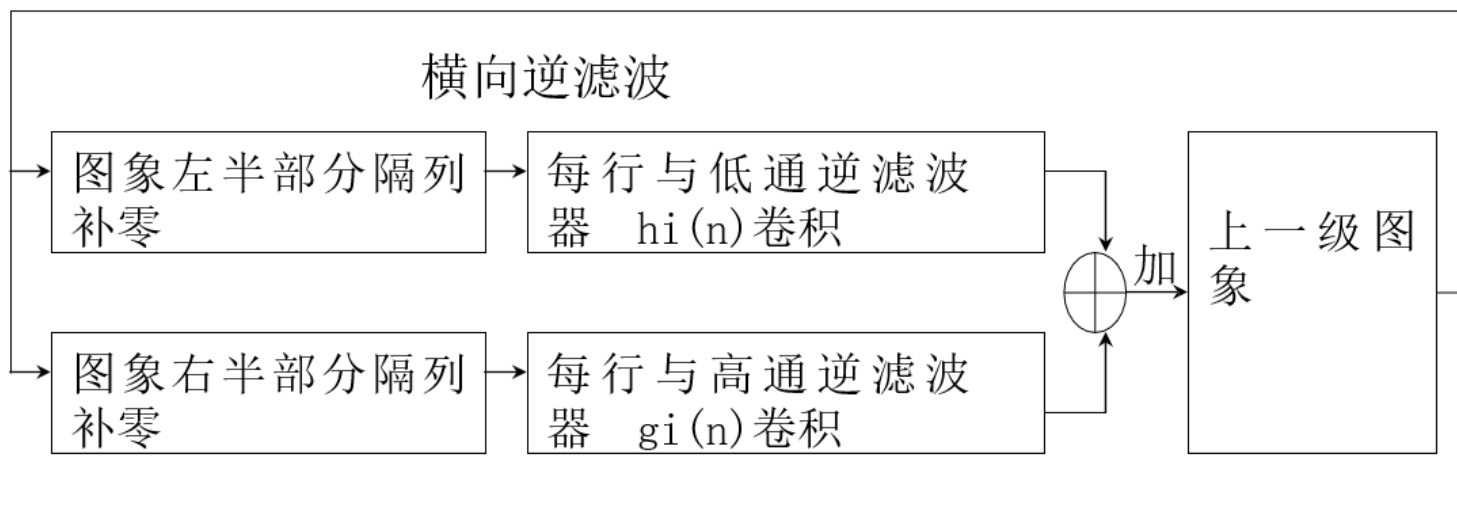


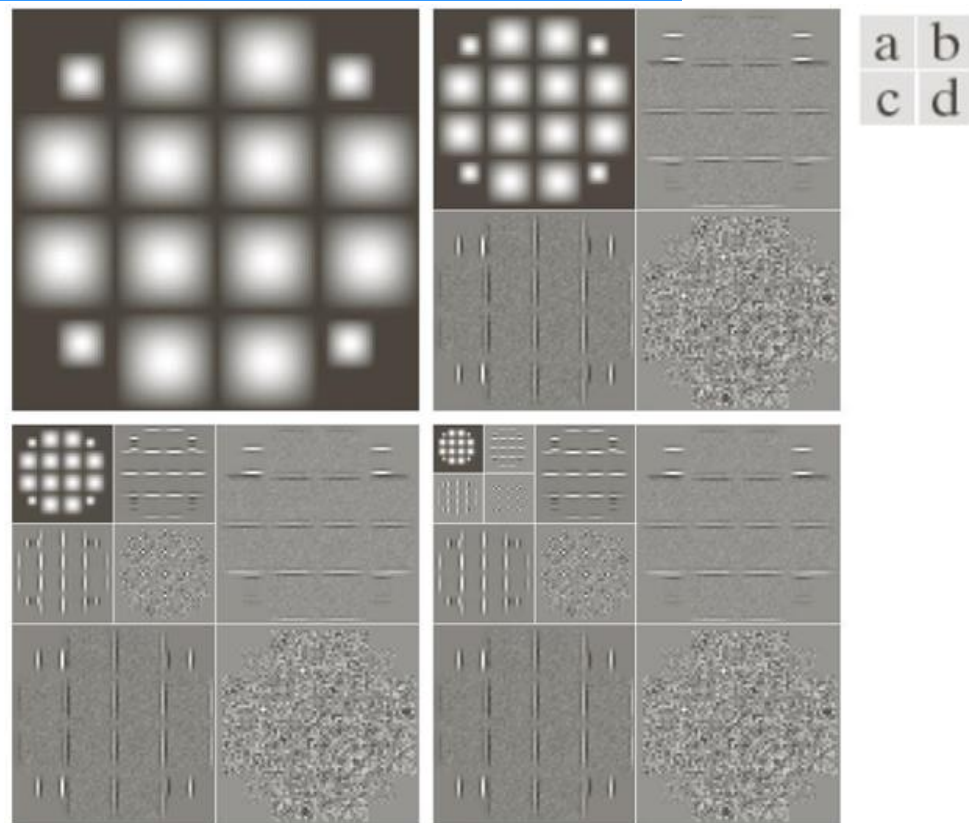
纵向逆滤波



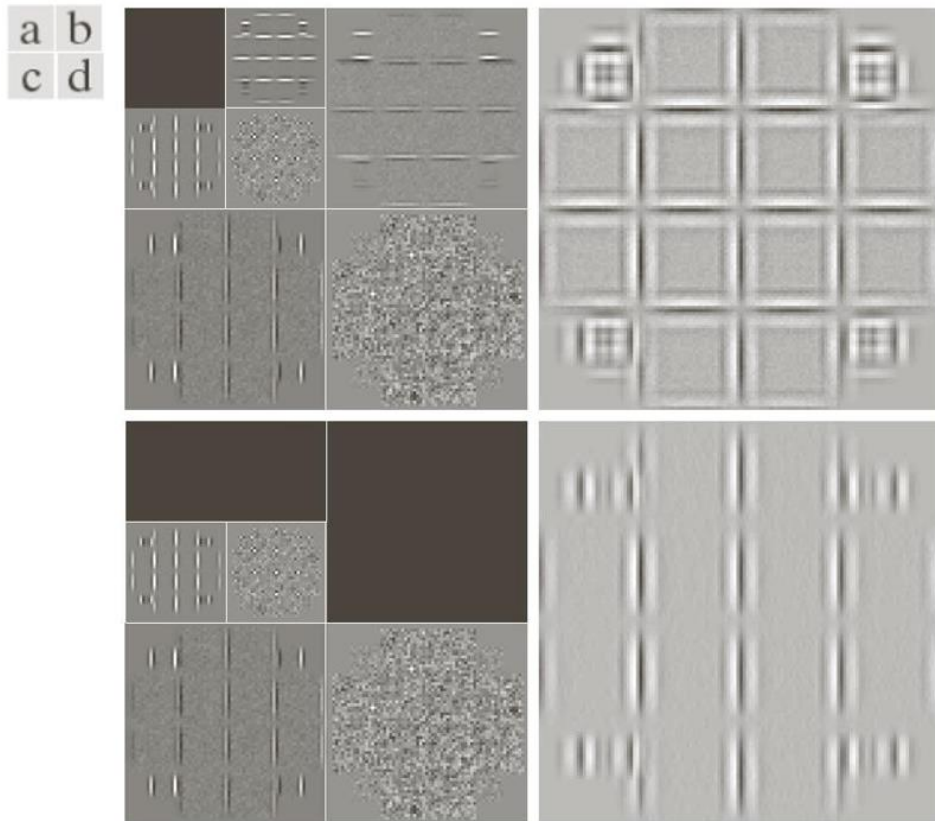
迭代

横向逆滤波

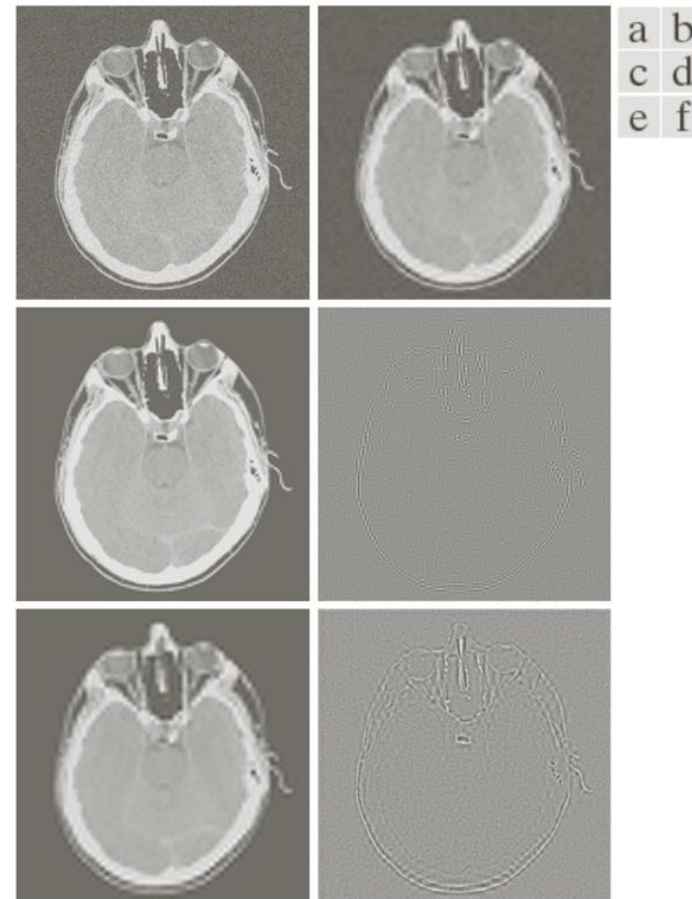




计算二维的三尺度FWT
(a) 原图像; (b) 一尺度FWT;
(c) 二尺度FWT; (d) 三尺度FWT



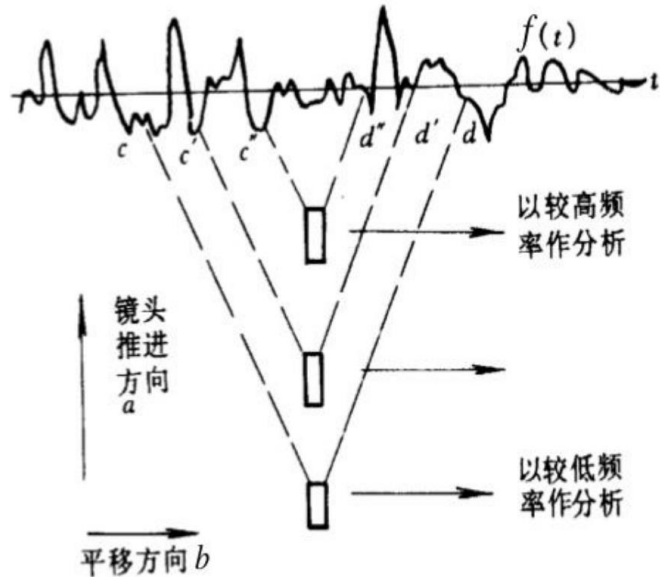
针对**边缘检测**改进的DWT
 (a) ~ (c) 删除所选系数的二尺度分解
 (b) ~ (d) 相应的重建



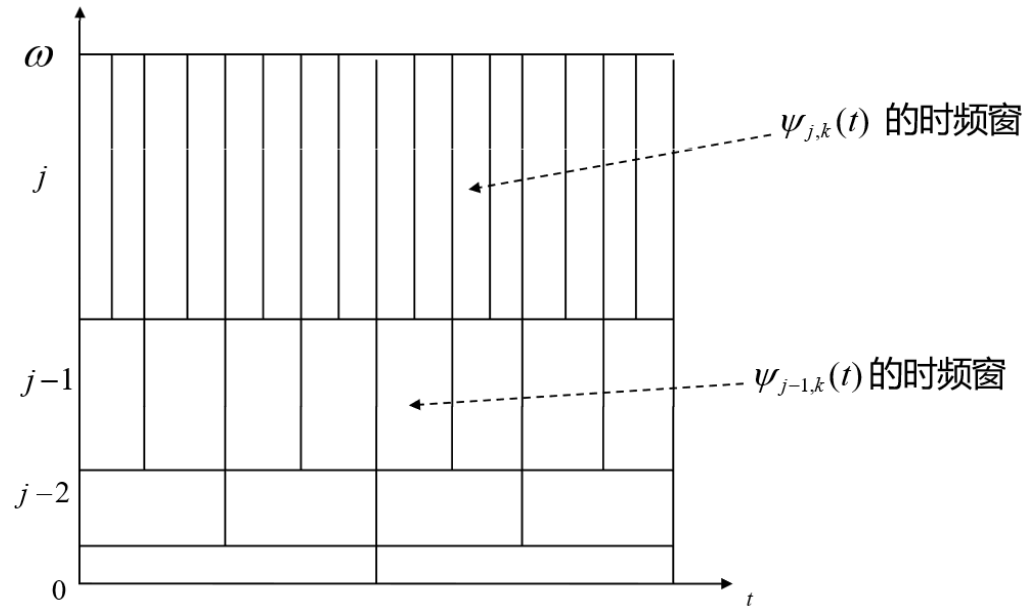
为**噪声去除**修改DWT: (a) 人的头部噪声CT图像;
 (b), (c)和(e)对细节系数进行阈值处理后的各种重建;
 (d)和(f)在(c)和(e)重建过程中所删除的信息

多分辨率分析

- 小波是多分辨率理论这种信号处理和分析方法的基础
 - 多分辨率理论涉及多个分辨率下的信号（或图像）表示与分析。
 - 这种方法的优势是：某种分辨率下无法检测的特性，在另一种分辨率下容易检测



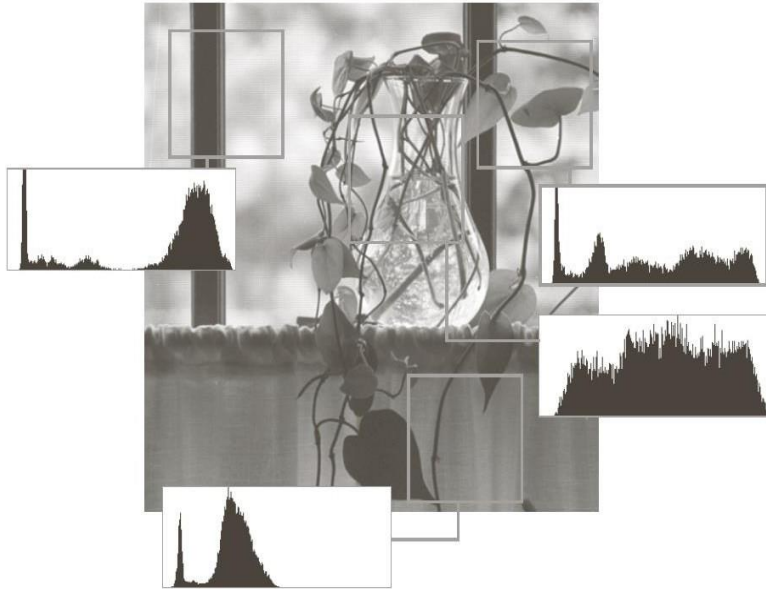
小波变换的粗略解释



离散小波分解自适应时-频窗示意图

□ 基本动机

- 图像中，区域由相似纹理和灰度级组成，不同区域结合形成物体
- 较小物体适合用较高分辨率分析
- 对于较大物体，用低分辨率分析即可
- 如果图像中同时存在较小物体和较大物体，则可用不同分辨率来研究他们



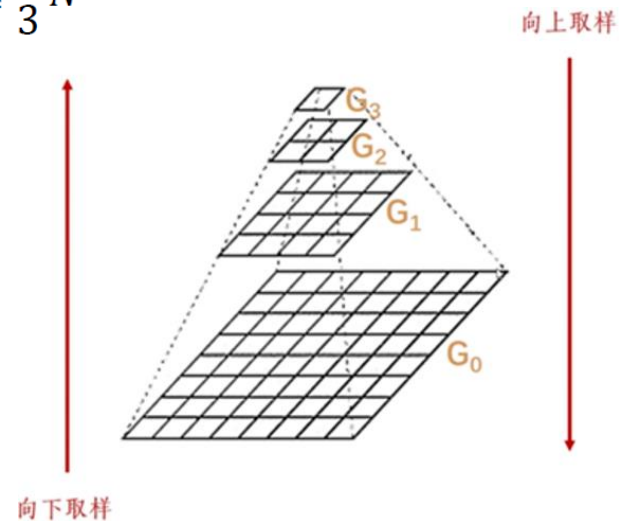
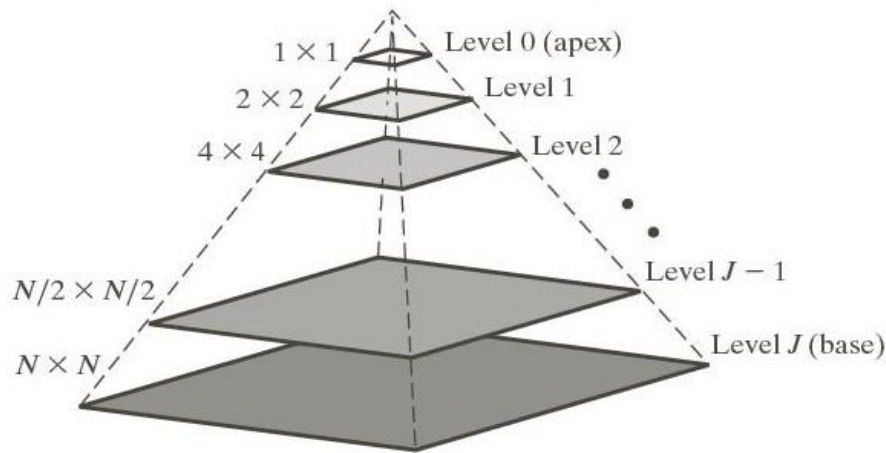
- 图像不同区域的局部直方图差异很大
- 难以对整图用统计模型进行分析

□ 图像金字塔:

- 一系列以金字塔形状排列的、分辨率逐渐降低的图像集合
- 底部: 高分辨率表示; 顶部: 低分辨率近似
- 最初用于机器视觉和图像压缩

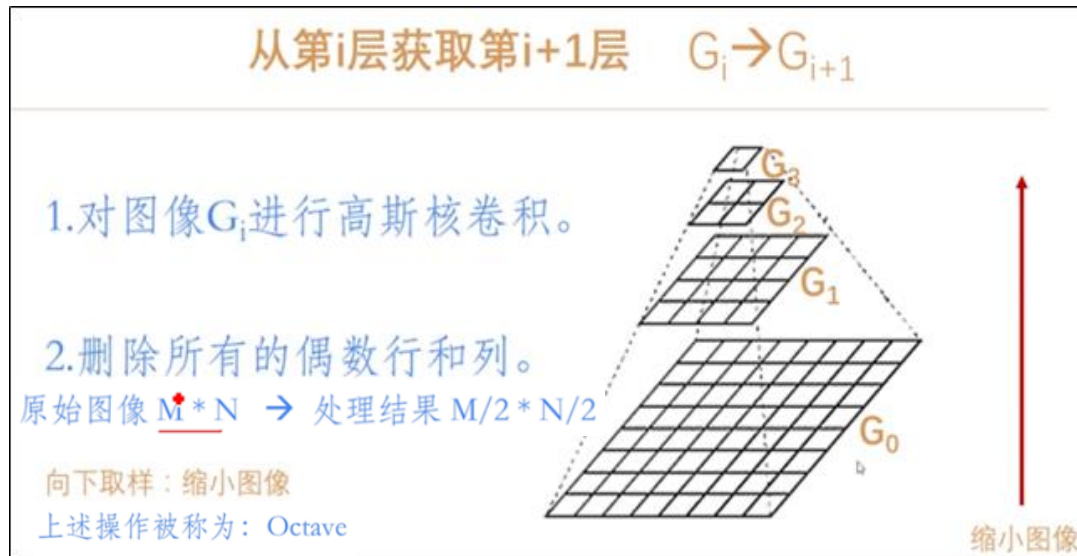
□ $P + 1$ 级图像金字塔像素总数是

$$N^2 \left(1 + \frac{1}{4^1} + \frac{1}{4^2} + \dots + \frac{1}{4^P} \right) \leq \frac{4}{3} N^2$$



□ 高斯金字塔

高斯金字塔是最基本的图像塔。首先将原图像作为最底层图像 G_0 （高斯金字塔的第0层），利用高斯核（ 5×5 ）对其进行卷积，然后对卷积后的图像进行下采样（去除偶数行和列）得到上一层图像 G_1 ，将此图像作为输入，重复卷积和下采样操作得到更上一层图像，反复迭代多次，形成一个金字塔形的图像数据结构，即高斯金字塔。

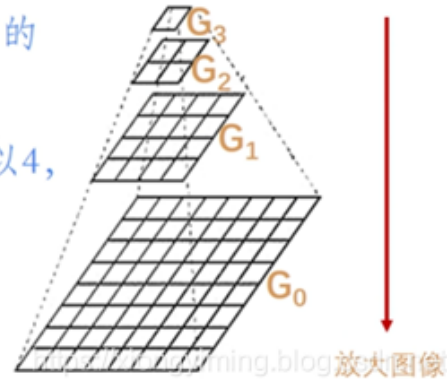


由于每次向下取样会删除偶数行和列，所以它会不停地丢失图像的信息



向上取样

- 在每个方向上扩大为原来的2倍，新增的行和列以0填充。
- 使用与“向下采用”同样的卷积核乘以4，获取“新增像素”的新值。



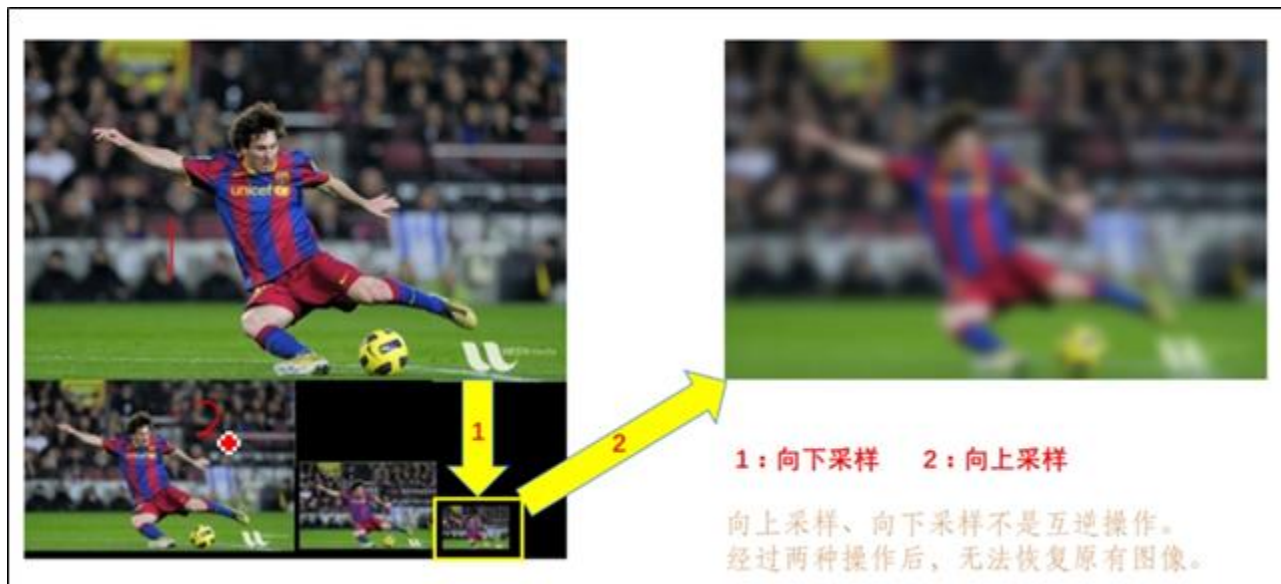
注意：放大后的图像比原始图像要模糊。

向上取样

- 在每个方向上扩大为原来的2倍，新增的行和列以0填充。
- 使用与“向下采用”同样的卷积核乘以4，获取“新增像素”的新值。
再与放大后的图像进行卷积运算

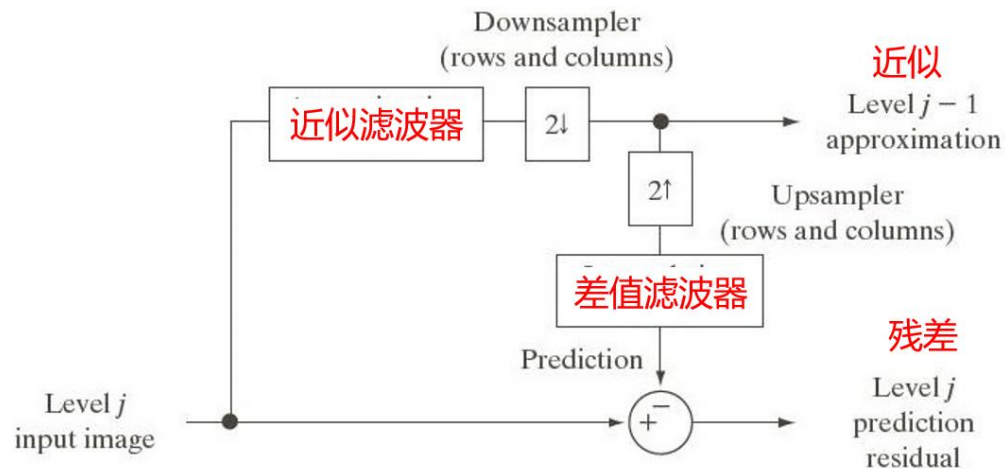
$$\begin{array}{|c|c|} \hline 45 & 123 \\ \hline 89 & 149 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{|c|c|c|c|} \hline 45 & 0 & 123 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 89 & 0 & 149 & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline \end{array}$$

所有元素都被规范化为4，而不是1 <https://xiongyiming.blog.csdn.net>



□ 创建近似金字塔和残差金字塔

- 步骤一：通过近似滤波器后，2倍下采样，得到第 $j-1$ 级近似
 - ✓ 近似滤波器：邻域平均、高斯低通滤波、无滤波
- 步骤二：由步骤一产生的分辨率降低的近似，2倍上采样，然后通过插值滤波，创建第 j 级输入图像的一个预测
 - ✓ 插值滤波：最近邻、双线性、双三次内插
- 步骤三：计算步骤二的预测图像和步骤一的输入之间的差，即预测残差



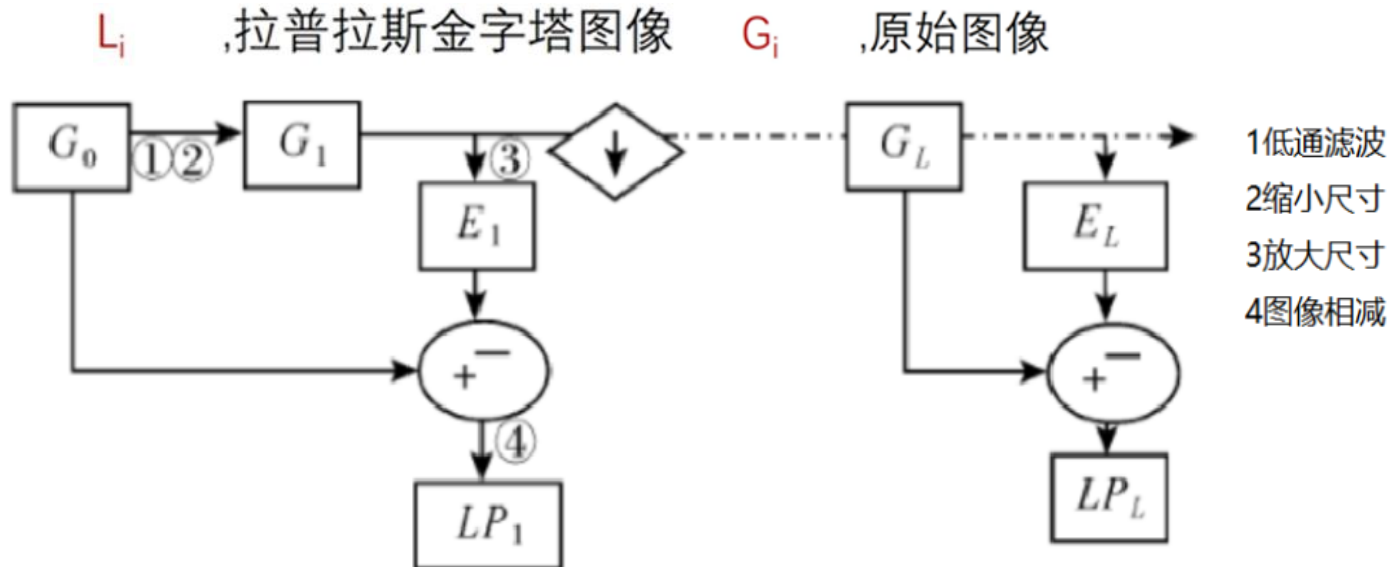
$$f_{2\uparrow}(n) = \begin{cases} f\left(\frac{n}{2}\right), & n \text{ 为偶数} \\ 0, & \text{其他} \end{cases}$$

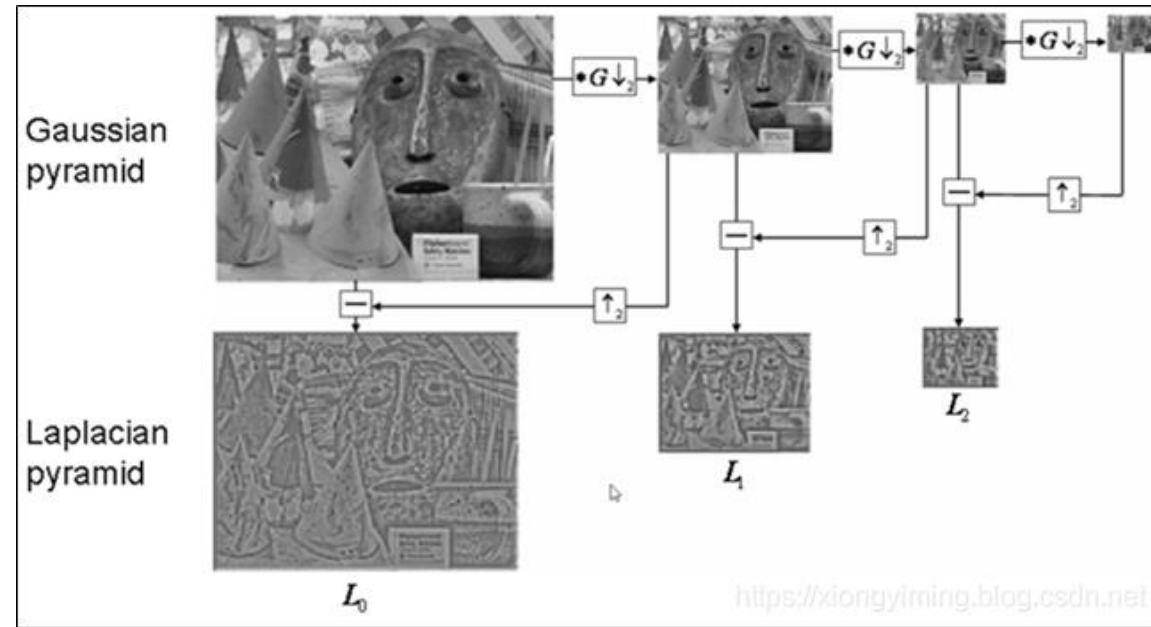
$$f_{2\downarrow}(n) = f(2n)$$

□ 拉普拉斯金字塔

在高斯金字塔的运算过程中，图像经过卷积和下采样操作会丢失部分高频细节信息。为描述这些高频信息，人们定义了拉普拉斯金字塔(Laplacian Pyramid, LP)。用高斯金字塔的每一层图像减去其上一层图像上采样并高斯卷积之后的预测图像，得到一系列的差值图像即为 LP 分解图像。

$$L_i = G_i - \text{PyrUp}(\text{PyrDown}(G_i))$$



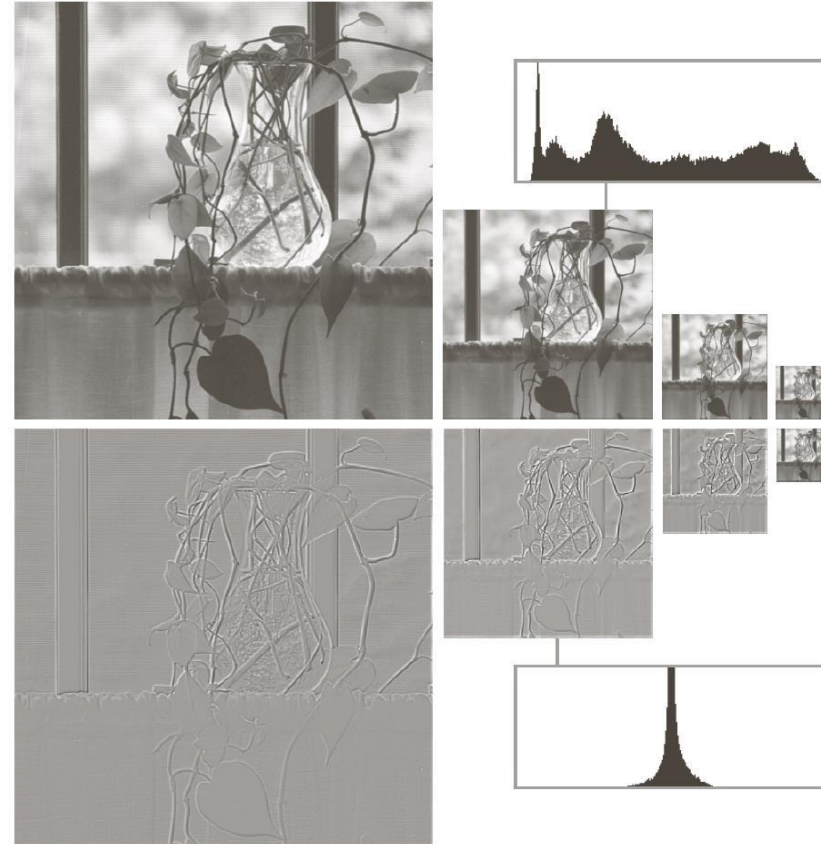


□ 上图：近似金字塔，称为**高斯**金字塔

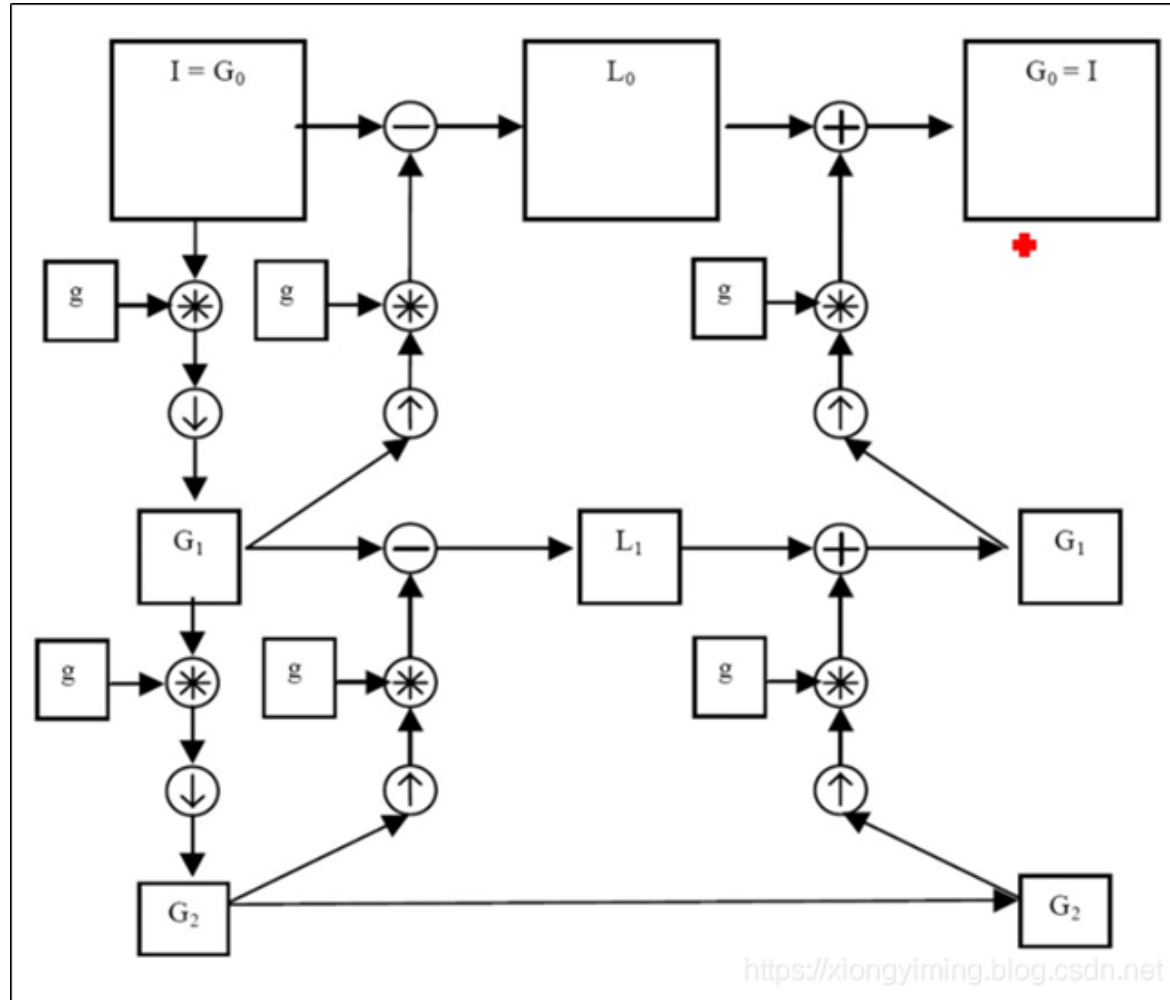
- 因为构建金字塔时使用了**高斯**滤波器
- 分辨率越低，细节越少
 - ✓ 低分辨率适合分析大结构或图像整体内容
 - ✓ 高分辨率适合分析单个物体特性

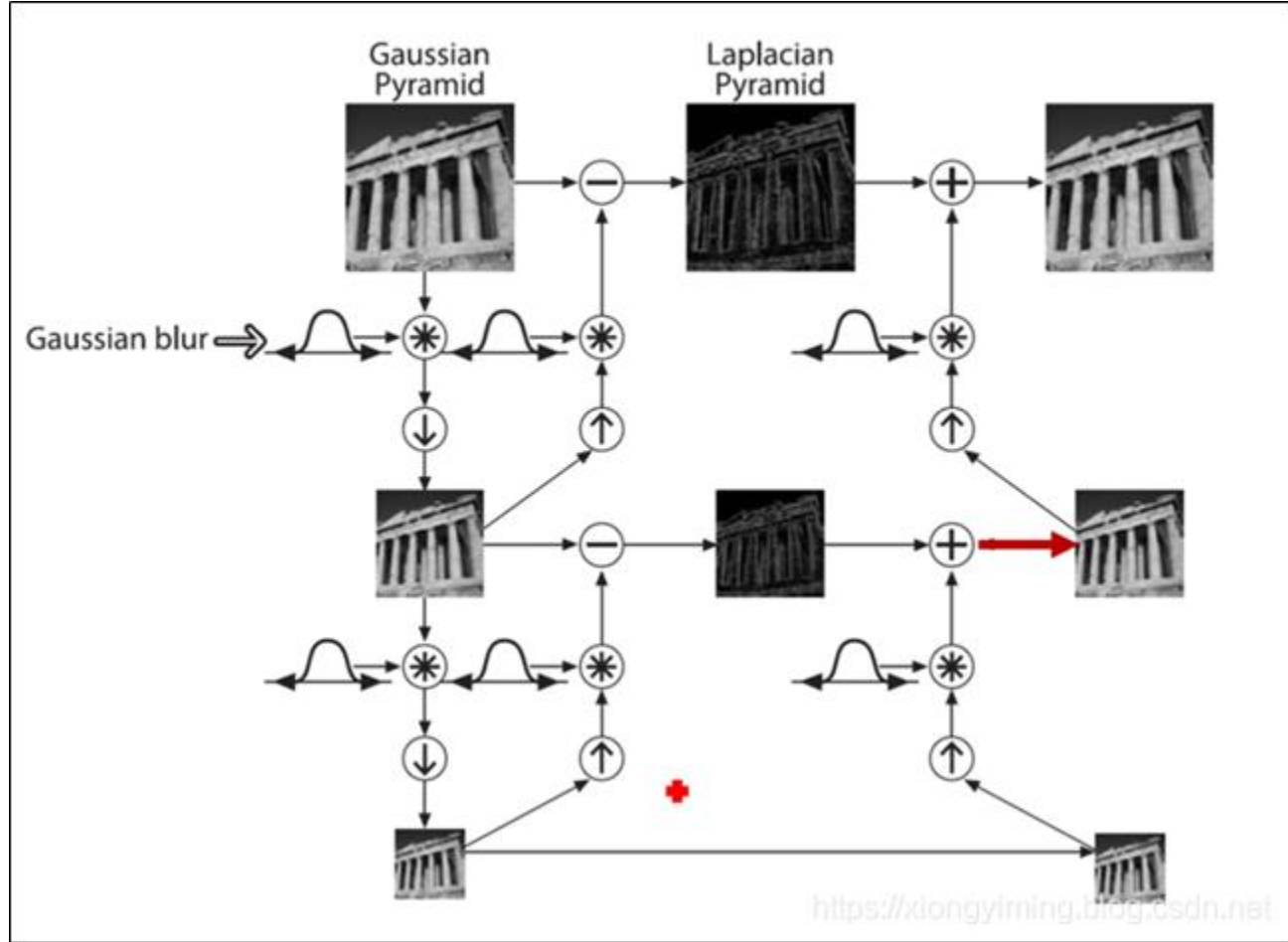
□ 下图：预测残差金字塔，通常称为**拉普拉斯**金字塔

- 直方图分布比较集中，适合用较少比特进行压缩



四级金字塔及它们的直方图









数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners





第5章 图像复原与重建

1. 图像退化/复原过程的模型
2. 噪声模型
3. 只存在噪声的复原——空间滤波
4. 用频率域消除周期噪声
5. 线性、位置不变的退化
6. 估计退化函数
7. 逆滤波
8. 最小均方误差（维纳）滤波
9. 约束最小二乘方滤波
10. 由投影重建图像

- 数字图象恢复与增强的目的类似，旨在**改善图象的质量**
- 但恢复是力求保持图象的本来面目，即以**保真**原则为其前提，这是区别于增强的基本不同点
- 因而恢复时要了解图象质量下降的物理过程，找出或估计其物理模型。恢复的过程就是沿着质量下降的逆过程来重现原始图象（利用退化现象的先验知识来复原已退化的图像）。





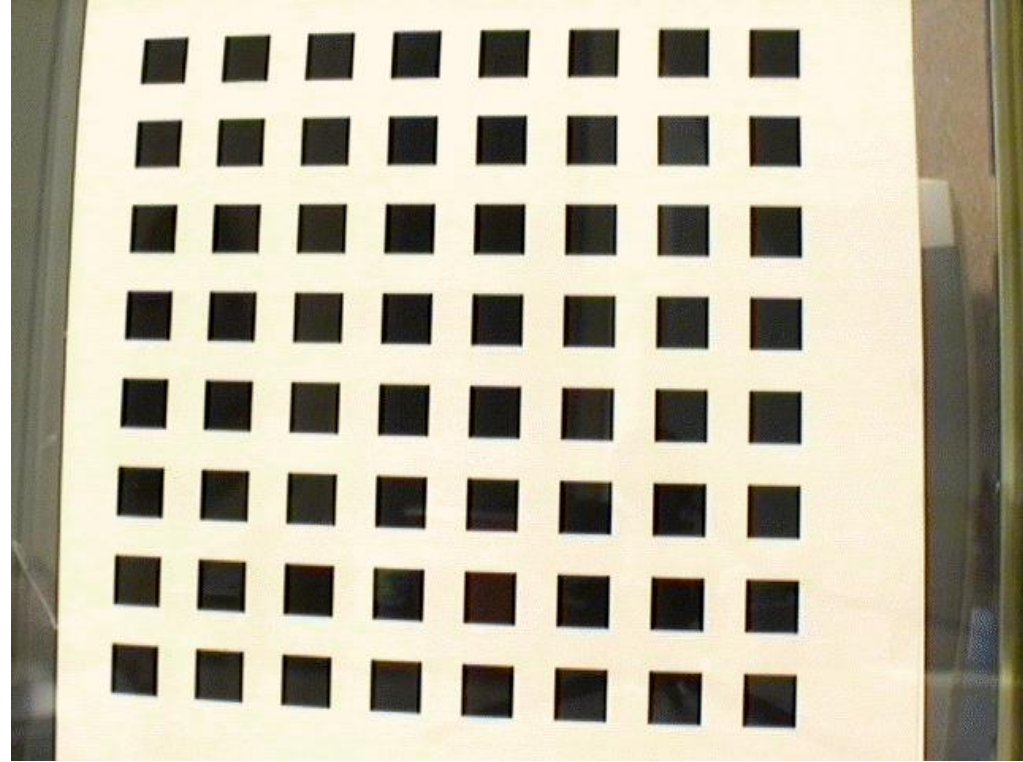
图像在形成、传输和记录过程中，由于成像系统、传输介质和设备的不完善，使图像的质量变坏，这一过程称为**图像的退化**。

图像的复原就是要尽可能恢复退化图像的本来面目，它是沿图像降质的逆向过程进行。典型的图像复原是根据图像退化的先验知识建立一个退化模型，以此模型为基础，采用各种逆退化处理方法进行恢复，使图像质量得到改善。

找图像退化的原因 -> 建立退化模型 -> 反向推演 -> 恢复图像



由于镜头聚焦不好引起的模糊



由于镜头畸变引起图像的几何失真



由于运动产生的模糊

5.1 图像退化/复原过程的模型

□ 线性位移不变系统，加性噪声

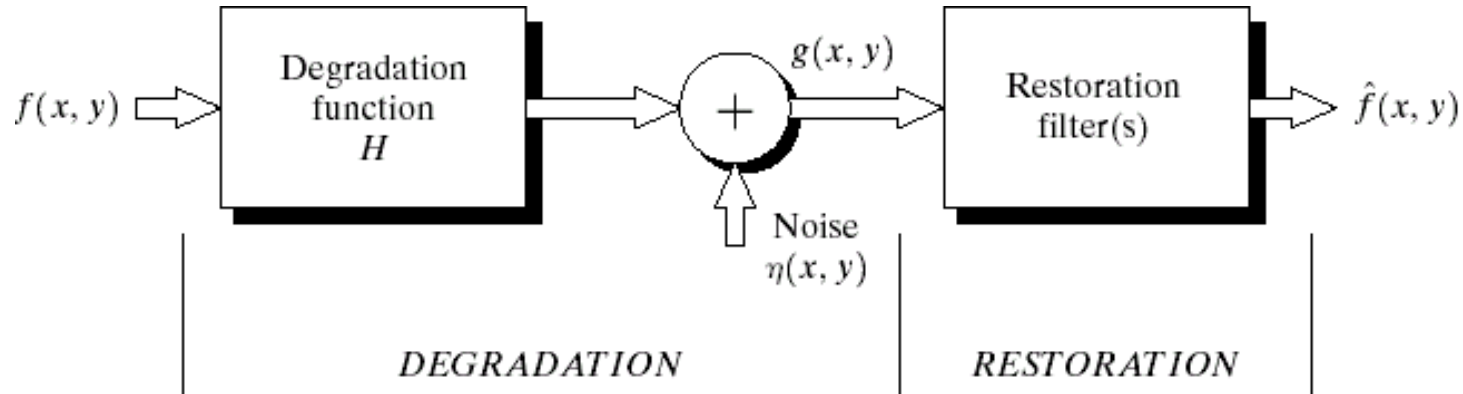


FIGURE 5.1 A model of the image degradation/restoration process.

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

A Mathematical Theory of Communication - Shannon - 1948 - Bell ...

<https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x> ▼

作者: CE Shannon - 1948 - 被引用次数: 109271 - 相关文章

[A Mathematical Theory of Communication](#). C. E. Shannon · Search for more papers by this author · C.

E. Shannon · Search for more papers by this author.

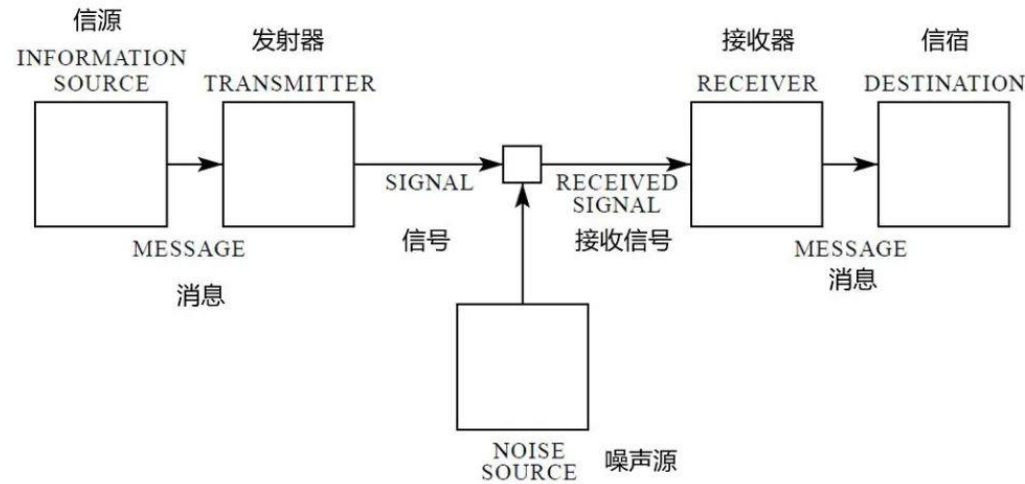
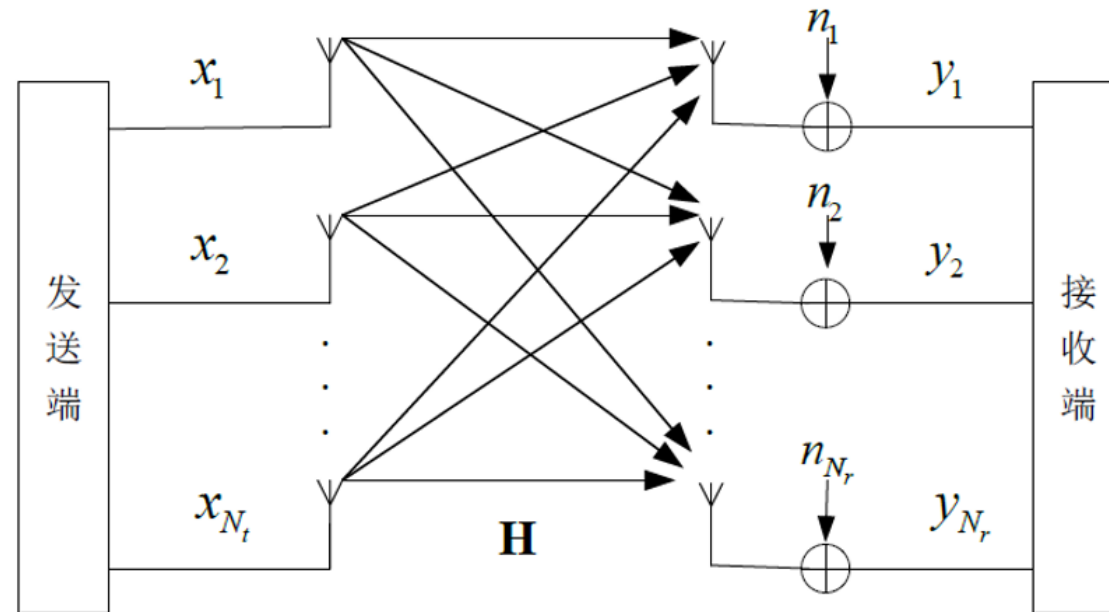


Fig. 1—Schematic diagram of a general communication system.

图1 —— 一般通信系统示意图

- 图像复原过程可以类比于MIMO系统的信号检测过程



$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$



5.2 噪声模型

- 噪声的产生
 - 图像获取: CCD相机
 - 图像传输: 无线传输
- 几种典型的噪声模型
 - 高斯噪声 (Gaussian noise)
 - 瑞利噪声(Rayleigh noise)
 - 厄兰噪声(Erlang noise)
 - 指数噪声(Exponential noise)
 - 均匀噪声(Uniform noise)
 - 椒盐噪声(salt-and-pepper)

概率密度函数 (PDF)

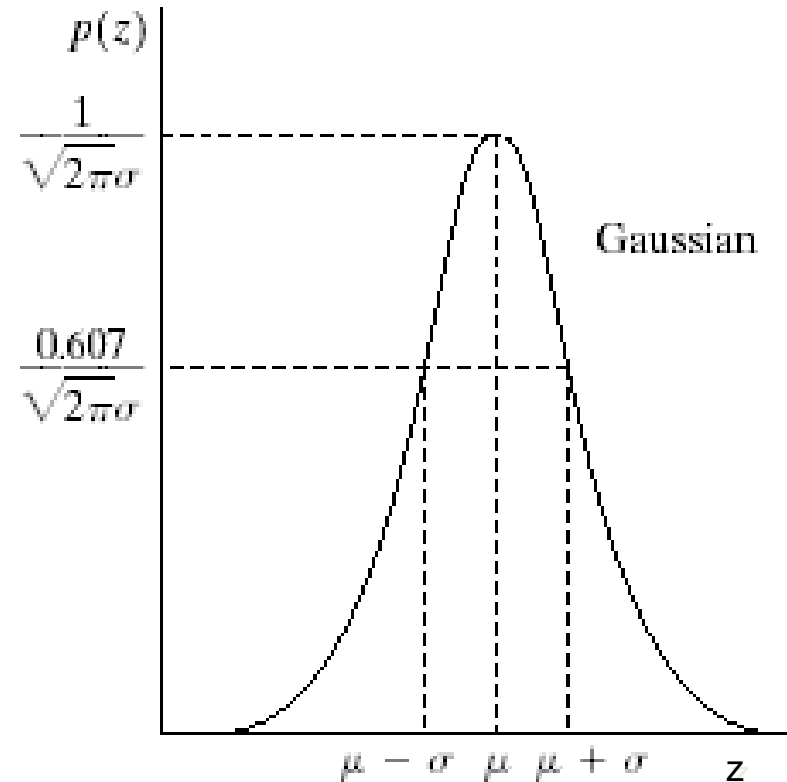
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$$

当 z 服从上式分布时,

其值有**70%**落在 $[(\mu - \sigma), (\mu + \sigma)]$

有**95%**落在 $[(\mu - 2\sigma), (\mu + 2\sigma)]$

高斯噪声的产生源于电子电路噪声和由低照明度或高温带来的传感器噪声。

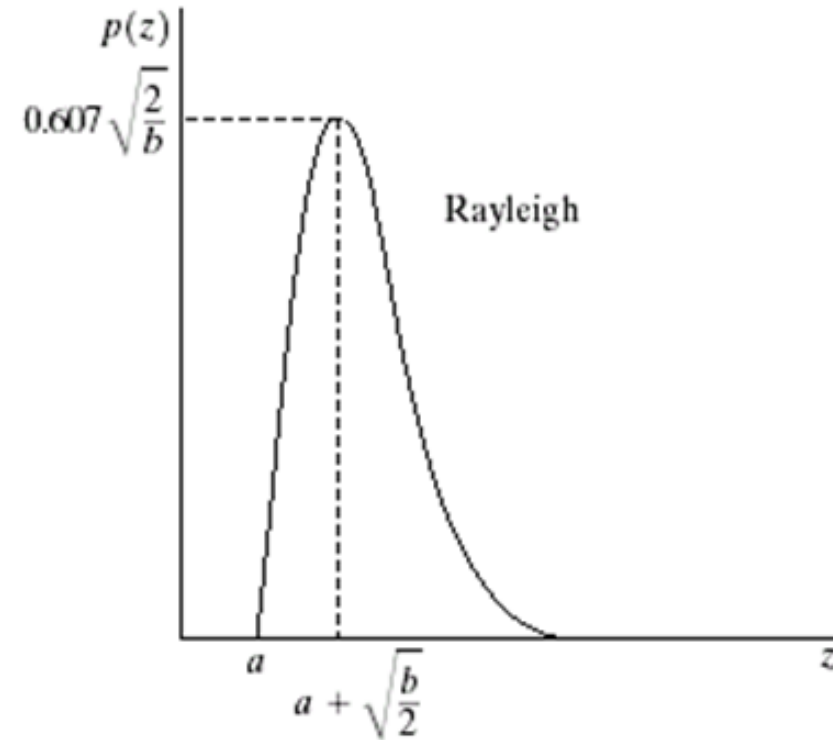


概率密度函数 (PDF)

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{2b}} & z \geq a \\ 0 & z < a \end{cases}$$

均值: $\mu = a + \sqrt{\pi b}/4$

方差: $\sigma^2 = b(4 - \pi)/4$



瑞利密度曲线距原点的位移和其密度向右变形瑞利密度对于近似歪斜的直方图，十分适用深度成像中

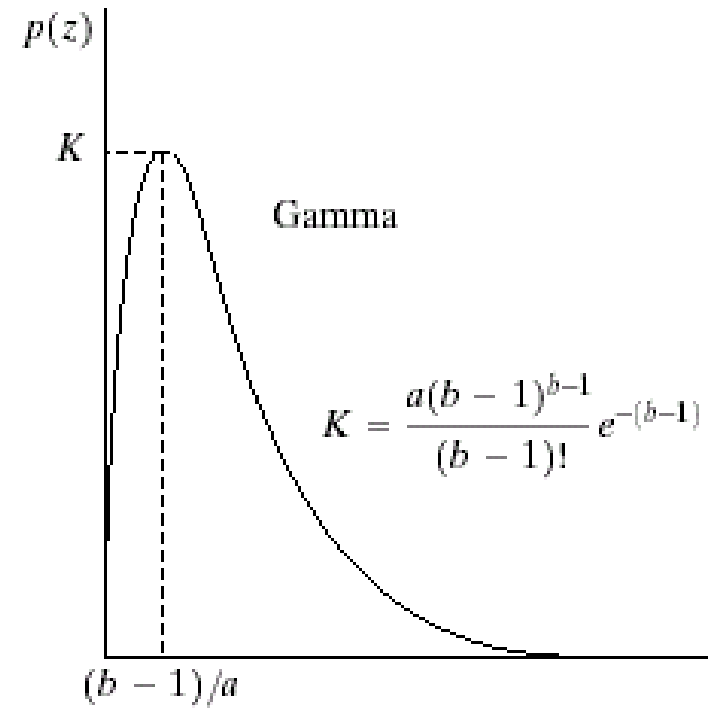
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$a > 0, b$ 为正整数

均值: $\mu = b/a$

方差: $\sigma^2 = b/a^2$

激光成像中

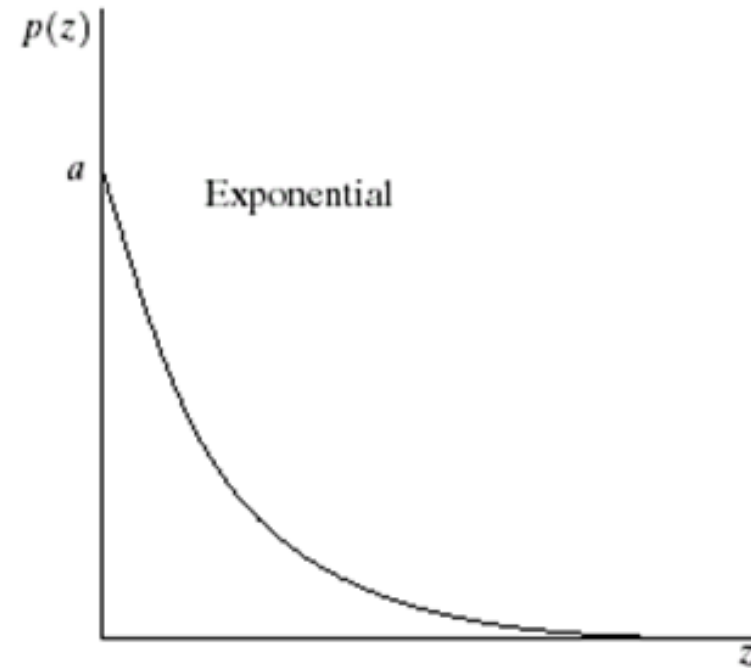


$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$a > 0$$

$$\text{均值: } \mu = 1/a$$

$$\text{方差: } \sigma^2 = 1/a^2$$

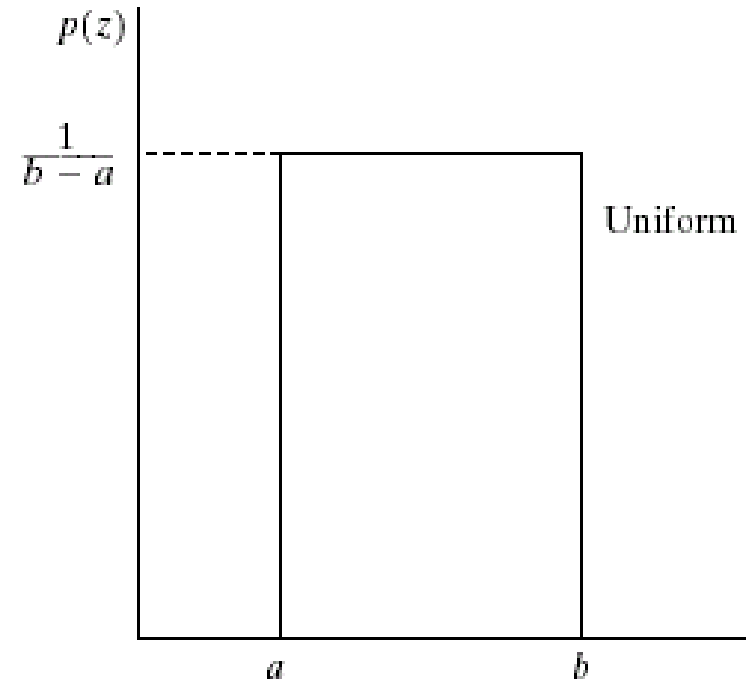


- 指数分布是 **$b=1$** 时爱尔兰概率分布的特殊情况。
- 指数分布噪声在激光成像中有些应用。

$$p(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{其它} \end{cases}$$

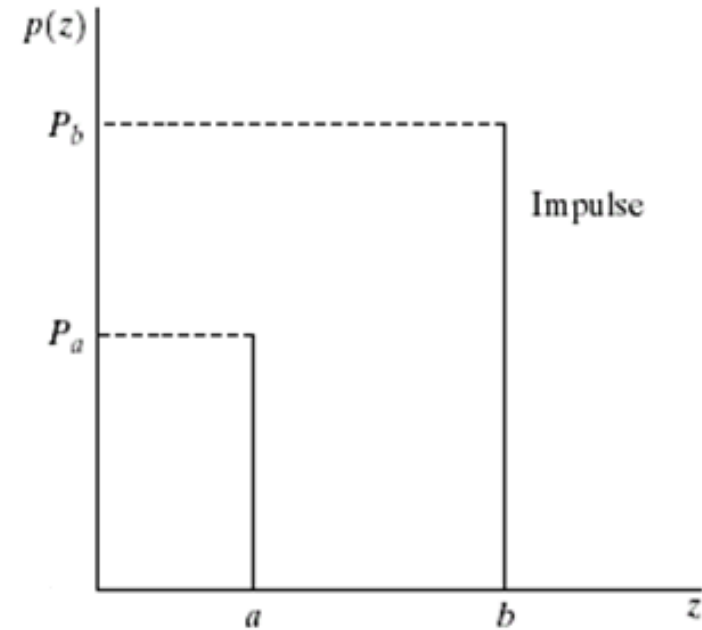
均值: $\mu = (a+b)/2$

方差: $\sigma^2 = (b-a)^2 / 12$



- 均匀分布噪声在实践中描述较少，但均匀密度分布作为模拟随机数产生器的基础非常有用。

$$p(z) = \begin{cases} p_a & z = a \\ p_b & z = b \\ 1 - p_a - p_b & \text{其它} \end{cases}$$



- 双极脉冲噪声也叫**椒盐噪声**，在图像上表现为孤立的亮点或暗点。
- 脉冲噪声表现在成像中的快速过度中，例如，错误的开关操作。
- 由于脉冲干扰通常与图像信号的强度相比较较大，因此，脉冲噪声总是被数字化为最大值或最小值。

噪声模型

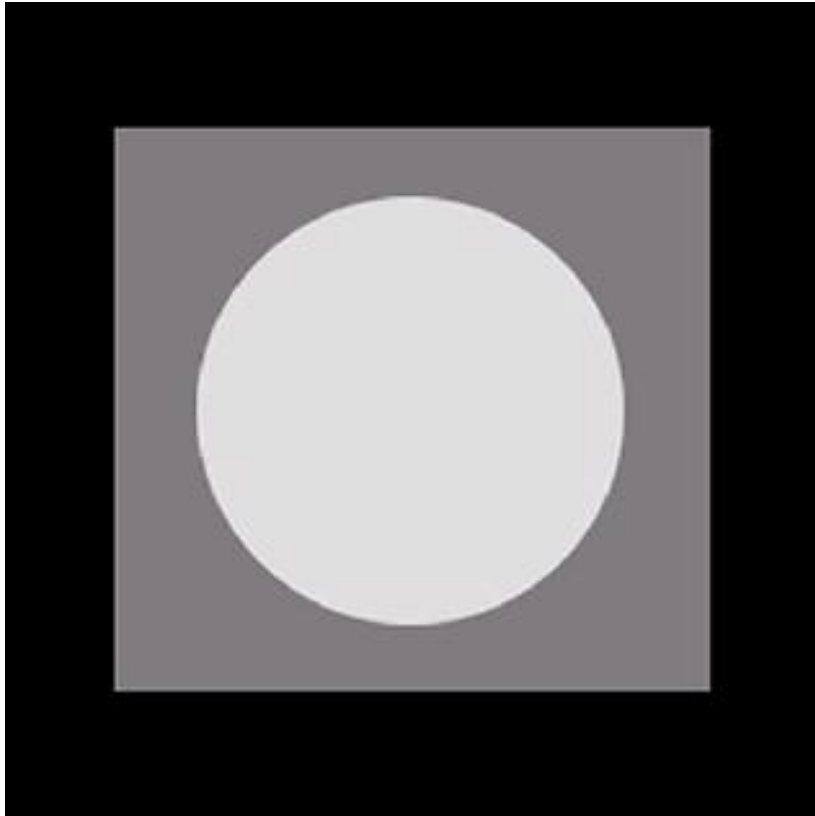
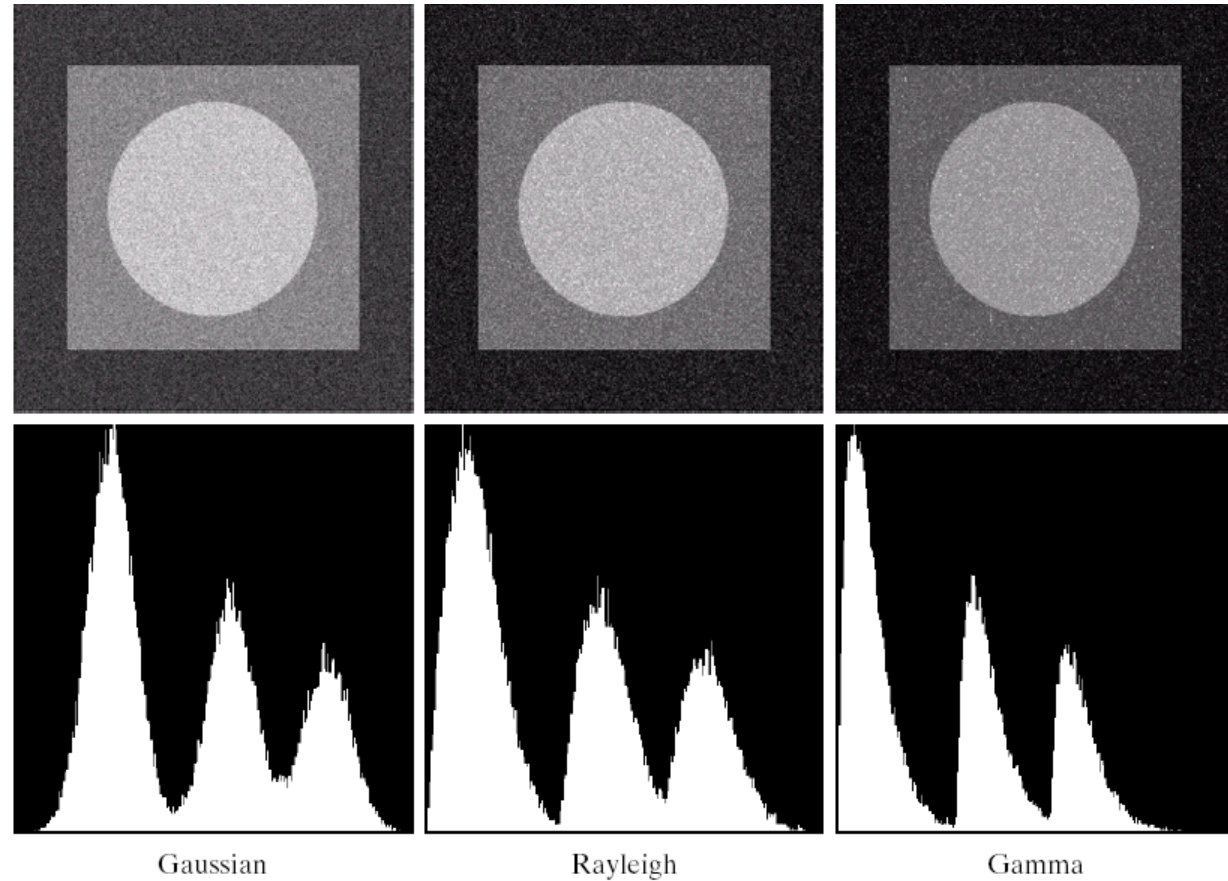
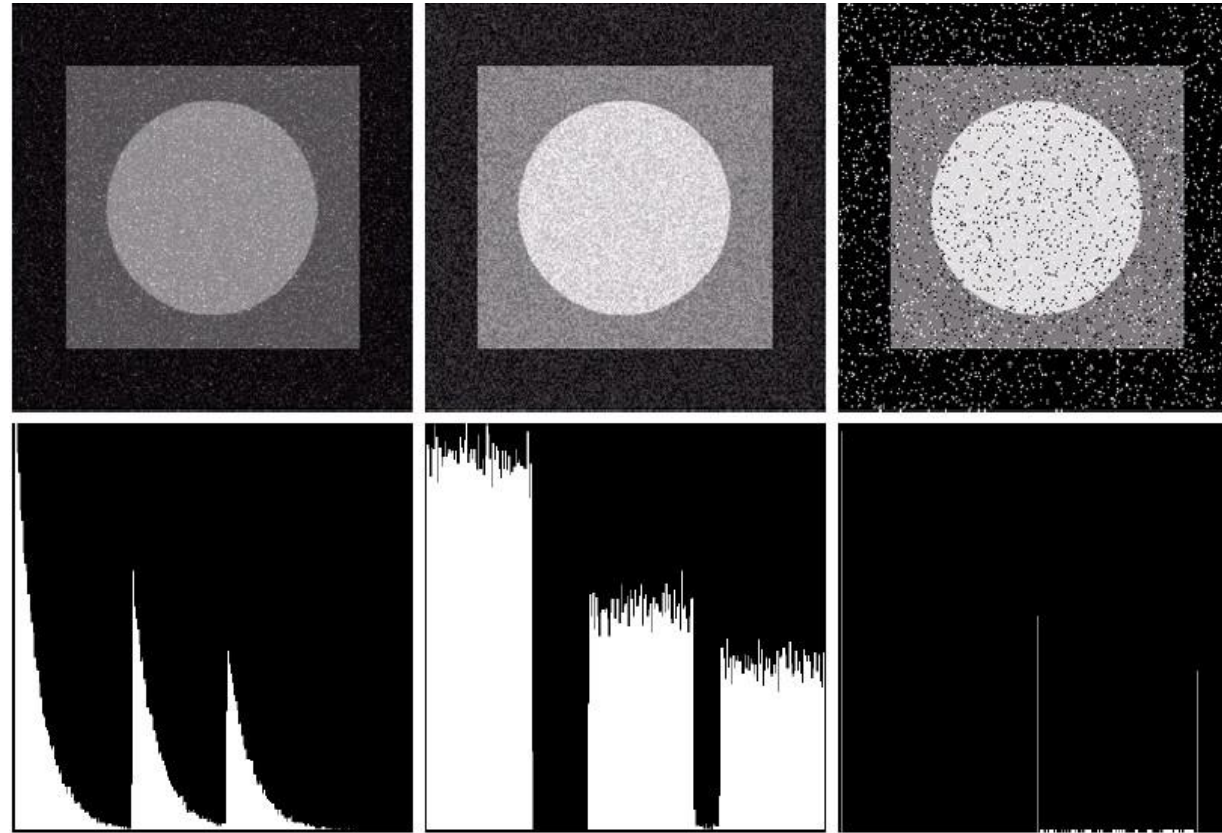


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.



a b c
d e f

FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.



Exponential

Uniform

Salt & Pepper

g h i
j k l

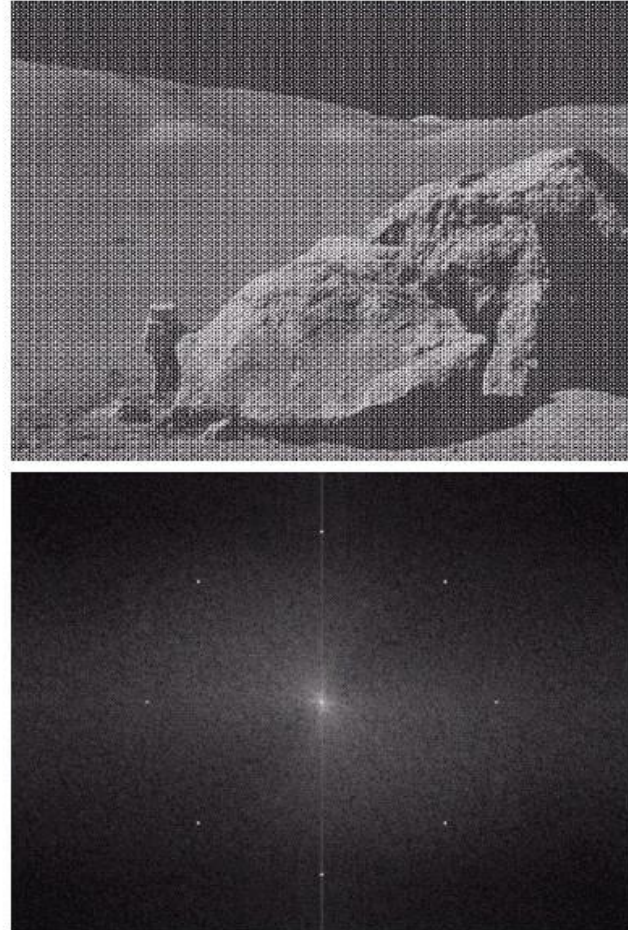
FIGURE 5.4 (Continued) Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

周期噪声(Periodic Noise)

- 图像周期噪声通常是在获取图像期间由电气或机电干扰产生的

a
b

FIGURE 5.5
(a) Image corrupted by sinusoidal noise.
(b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)





5.3 只存在噪声的复原——空间滤波

- 基于空域滤波的方法仅针对加性噪声

$$g(x, y) = f(x, y) + \eta(x, y)$$

$$G(u, v) = F(u, v) + N(u, v)$$

- 与空域增强原理相同
- 几种常用滤波器
 - 均值滤波器 (Mean Filters)
 - 次序统计滤波器 (Order Statistic Filters)
 - 自适应滤波器 (Adaptive Filters)

几种常用滤波器

- 算术平均滤波器(Arithmetic mean filters)

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} g(r, c)$$

降低图像的噪声，
但也会模糊图像

- 几何平均滤波器(Geometric mean filter)

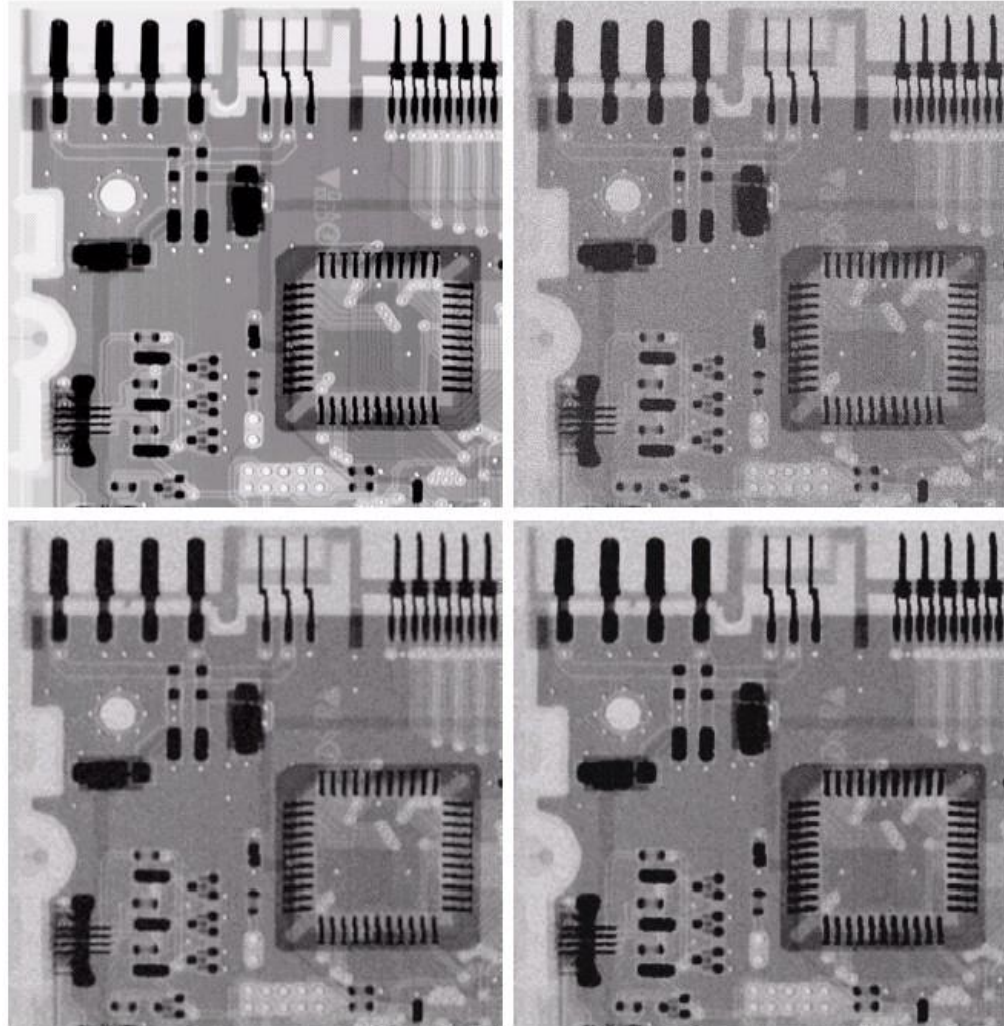
$$\hat{f}(x, y) = \left[\prod_{(r,c) \in S_{xy}} g(r, c) \right]^{\frac{1}{mn}}$$

相比于算数平均，
损失的细节更少

- 调和平均滤波器(Harmonic mean filter)

$$\hat{f}(x, y) = \frac{mn}{\sum_{(r,c) \in S_{xy}} \frac{1}{g(r, c)}}$$

既能处理盐粒噪声，又
能处理类似高斯噪声的
其它噪声，但不能处理
胡椒噪声



a	b
c	d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3 . (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

次序统计滤波器

■ 中值滤波器(Median filter)

$$\hat{f}(x, y) = \underset{(s,t) \in S_{xy}}{\text{median}} \{g(s, t)\}$$

适于处理椒盐噪声，通过多次使用小模板，可以获得很好的去噪效果

■ 最大最小滤波器(Max and min filters)

$$\begin{cases} \hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \\ \hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \end{cases}$$

这种滤波器对发现图像中的最亮点非常有用，可以用来消除胡椒噪声

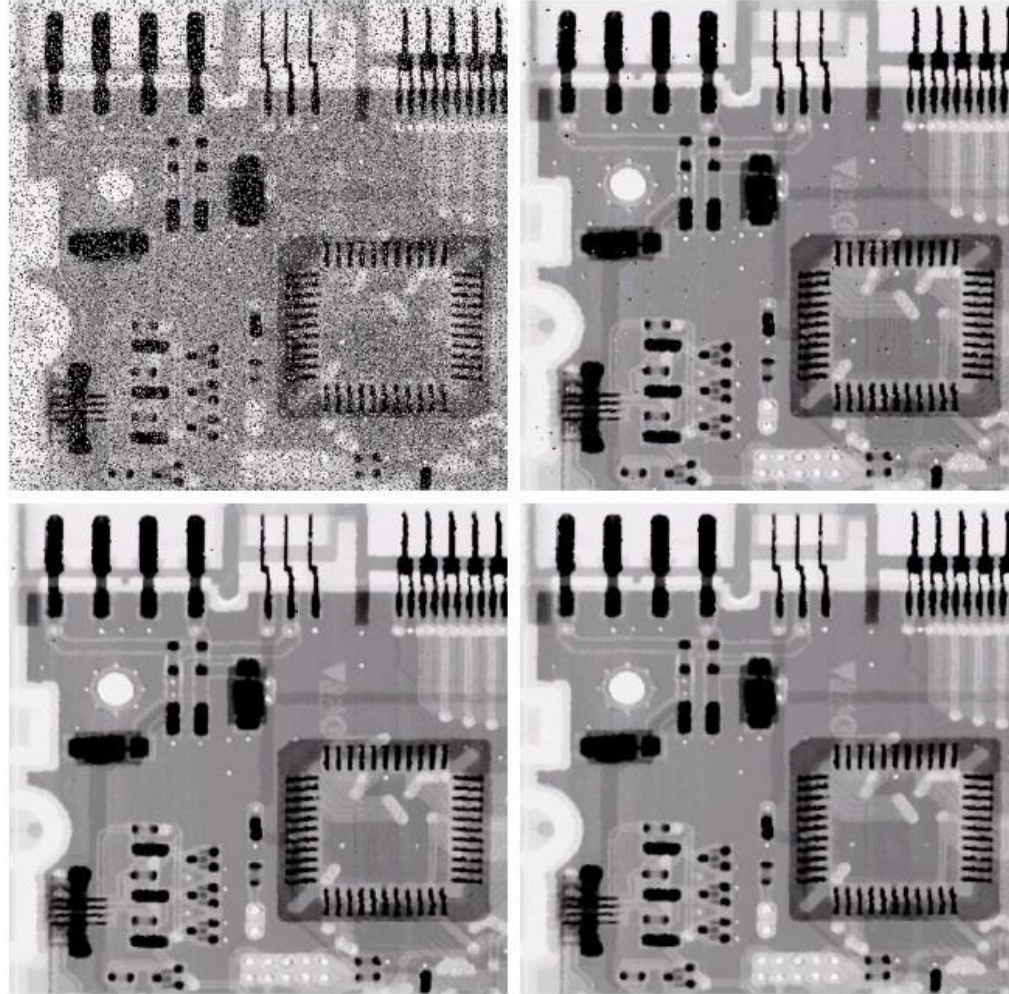
■ 中点滤波器(Midpoint filter)

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right]$$

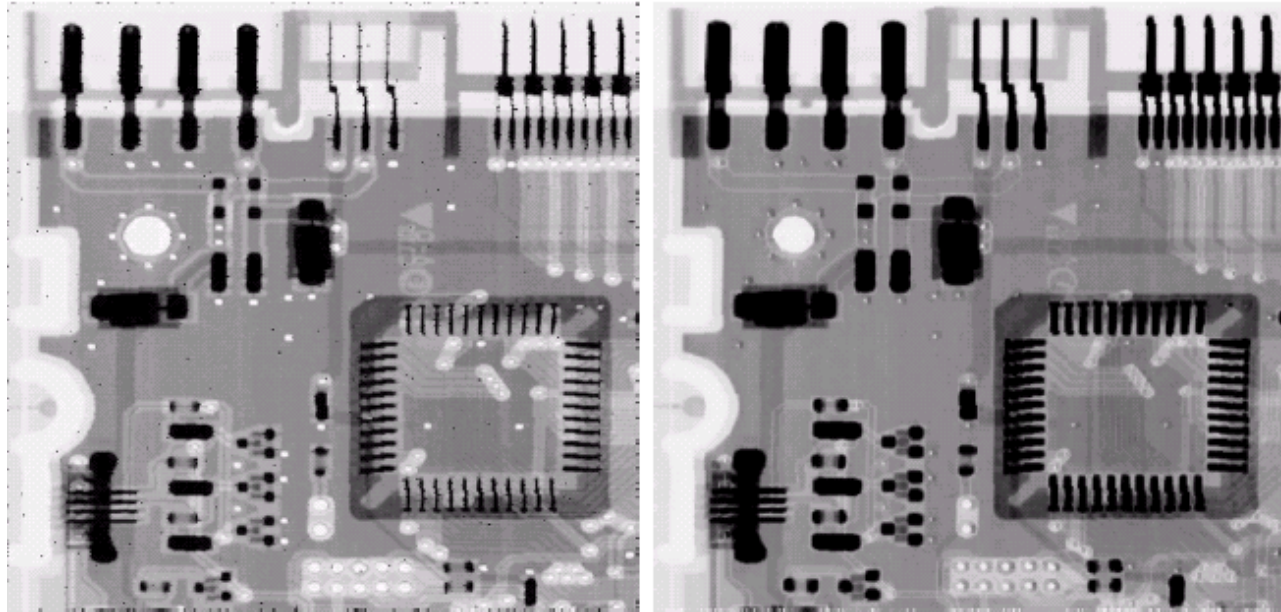
这种滤波器对发现图像中的最暗点非常有用，可以用来消除盐噪声

a b
c d

FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$.
(b) Result of one pass with a median filter of size 3×3 .
(c) Result of processing (b) with this filter.
(d) Result of processing (c) with the same filter.



最大最小滤波器滤波

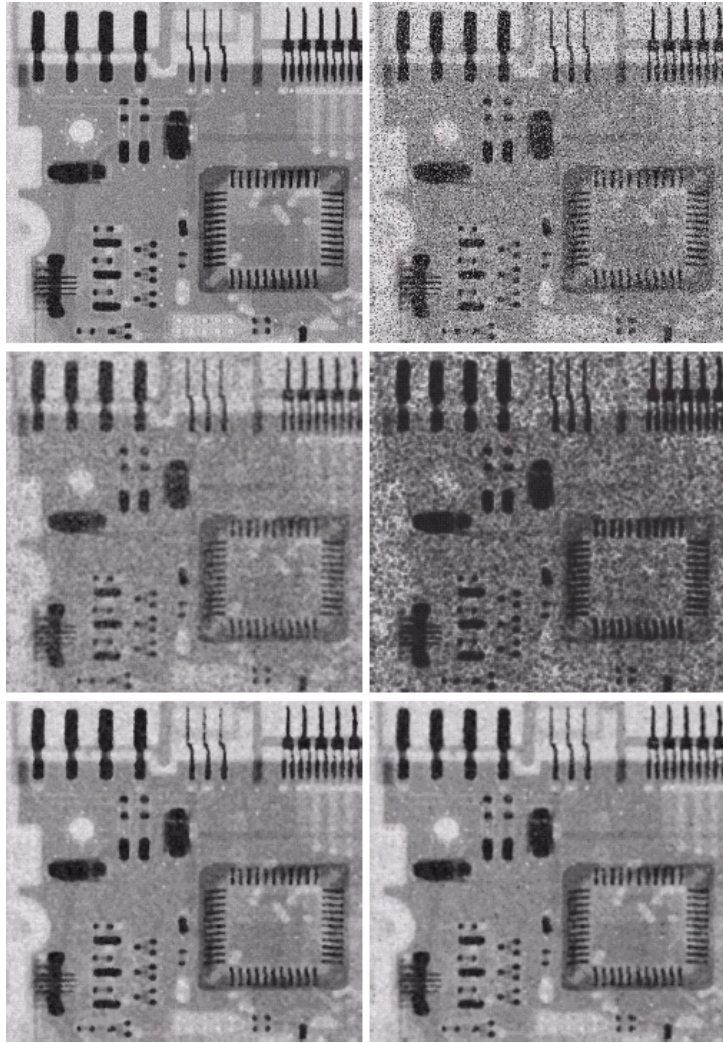


a b

FIGURE 5.11

(a) Result of filtering

Fig. 5.8(a) with a max filter of size 3×3 . (b) Result of filtering 5.8(b) with a min filter of the same size.



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5 : (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

原图



原图+椒盐噪声



中值滤波



最大值滤波



最小值滤波



<https://blog.csdn.net/dianglavender>



5.4 频域滤波消除周期噪声

Periodic Noise Reduction by Frequency Domain Filtering

- 带阻滤波器(Bandreject Filters)
 - 带通滤波器(Bandpass Filters)
 - 槽口滤波器(Notch Filters)
-



带阻滤波器(Bandreject Filters)

理想带阻滤波器(Ideal Bandreject Filters)

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1 & \text{if } D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

Butterworth带阻滤波器(Butterworth Bandreject Filters)

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)W}{D^2(u, v) + D_0^2} \right]^{2n}}$$

高斯带阻滤波器(Gaussian Bandreject Filters)

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[\frac{D^2(u, v) + D_0^2}{D(u, v)W} \right]^2}$$

带阻滤波器(Bandreject Filters)

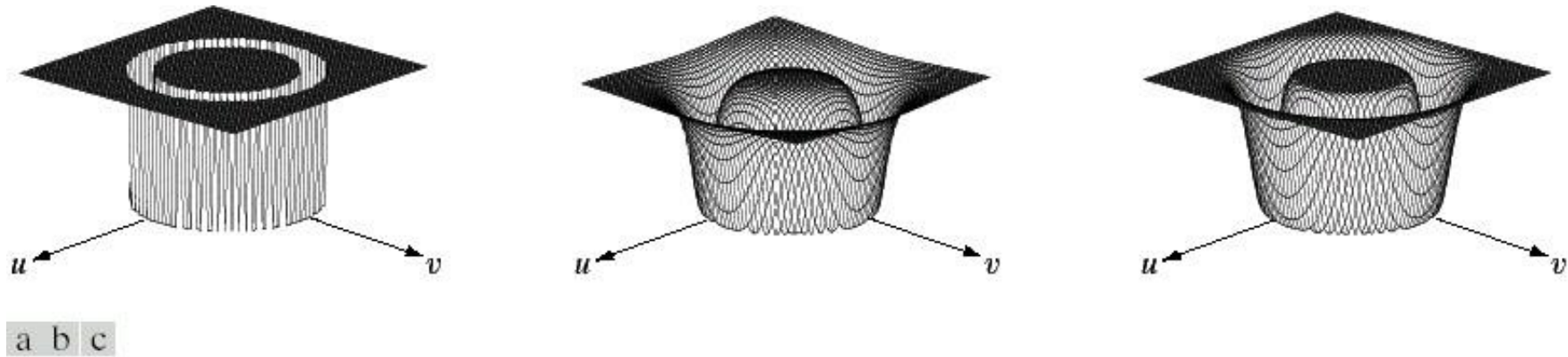
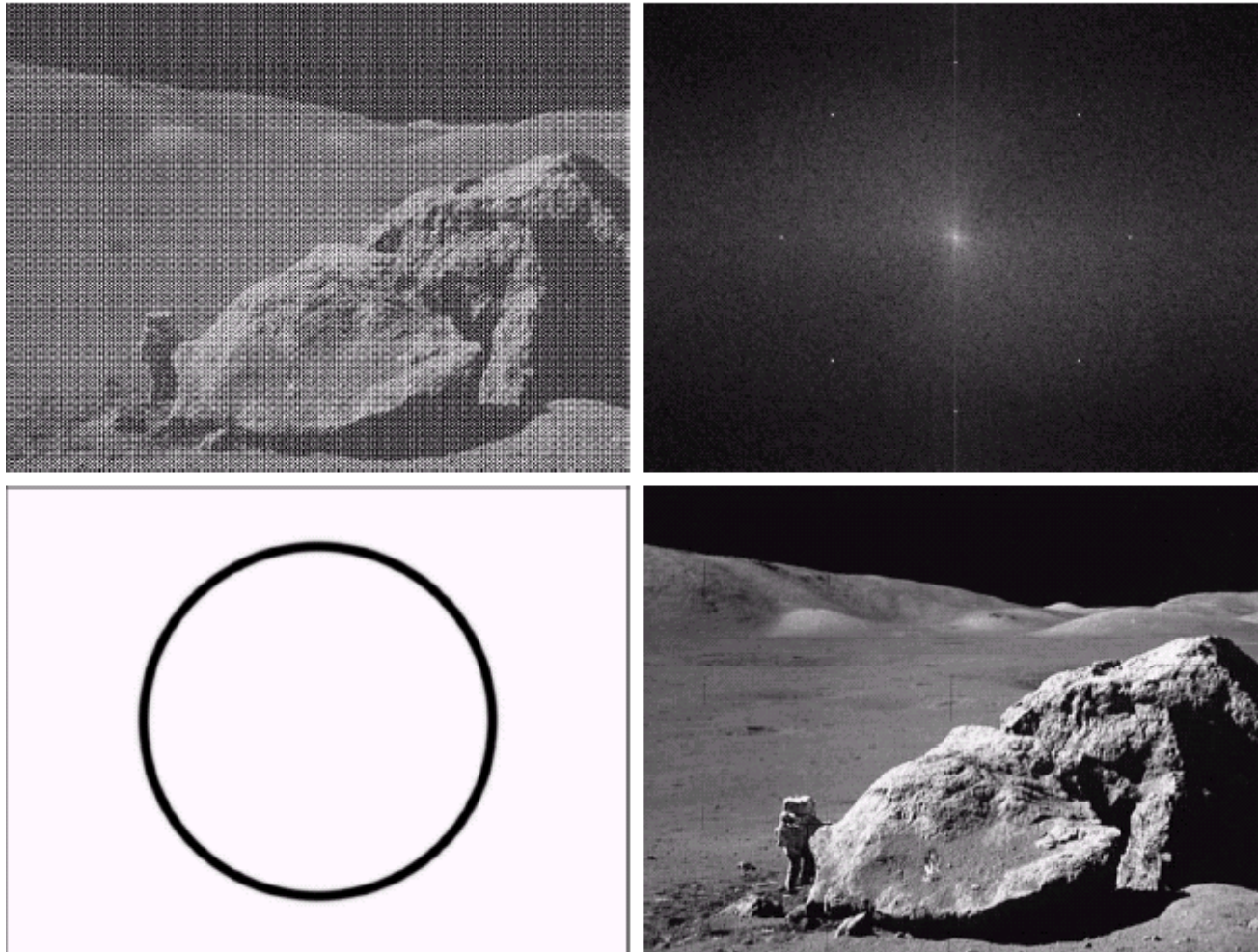


FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters

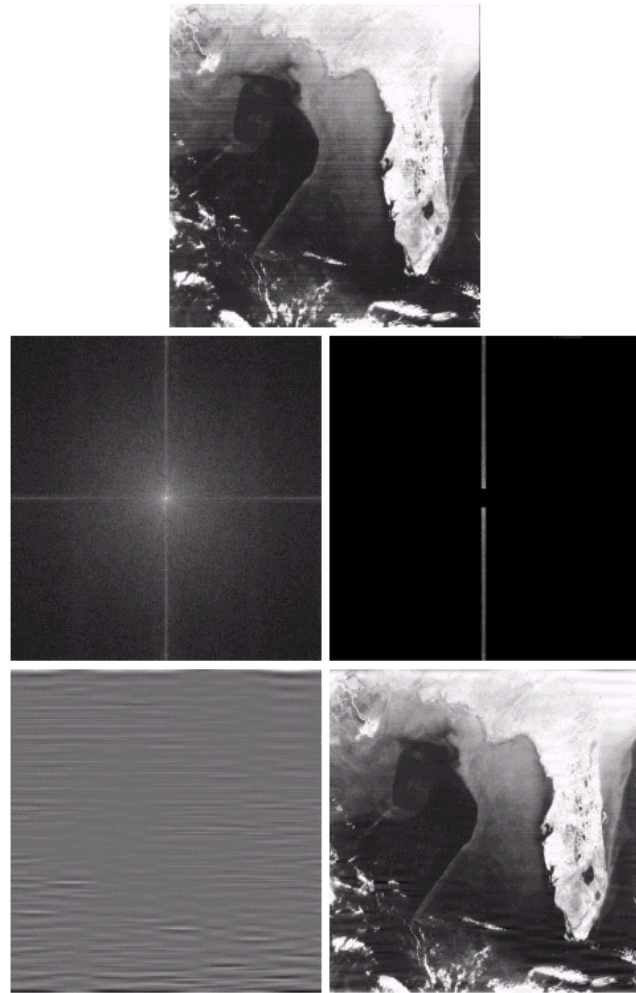
带阻滤波器(Bandreject Filters)



a b
c d

FIGURE 5.16

(a) Image corrupted by sinusoidal noise.
(b) Spectrum of (a).
(c) Butterworth bandreject filter (white represents 1).
(d) Result of filtering. (Original image courtesy of NASA.)



a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)



5.5 线性、位置不变的退化模型

降级过程:

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$

线性系统:

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$$

位置不变系统:

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta)$$

线性位置不变降级模型

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$



5.6 估计退化函数

- Estimating the Degradation Function
 - 通过观测图像估计(Estimation by Image Observation)
 - 通过实验估计(Estimation by Experimentation)
 - 通过数学建模估计(Estimation by Modeling)

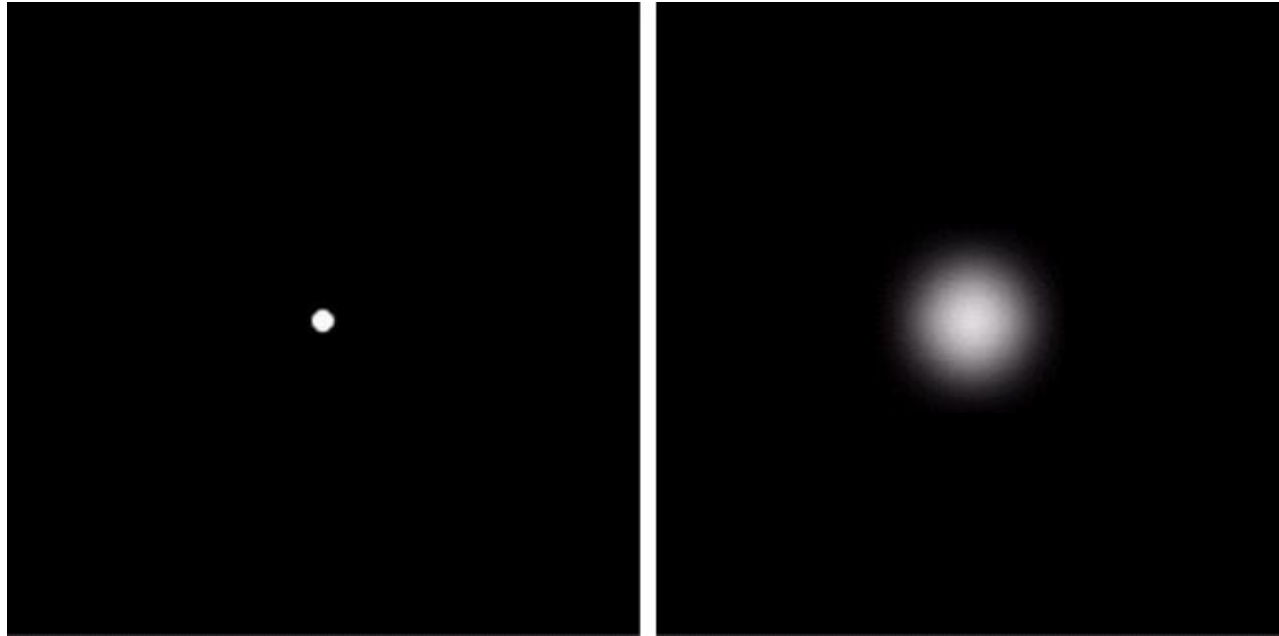
(1) 图像观测估计

- 假设：退化过程为线性、位置不变的
- 通过图像本身收集信息来估计退化函数 H
 - 从图像中选取有很强信号内容的区域（高对比度）
 - 对该图像进行处理，得到尽可能不模糊的结果
 - ✓ 如锐化处理、甚至手工方法处理
 - 基于处理之后的图像频谱，估计退化函数频谱

$$H(u, v) = \frac{G(u, v)}{F(u, v)}$$

(2) 试验估计

- 假设退化过程可用设备近似，重复成像



a b

FIGURE 5.24

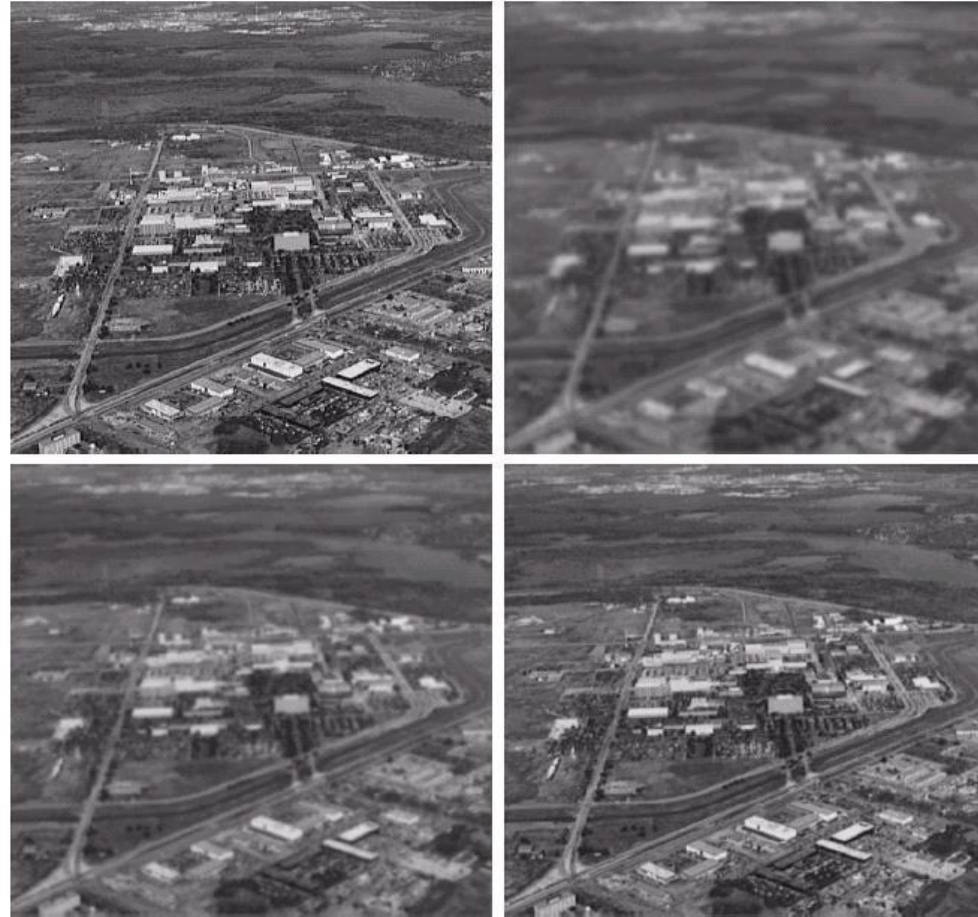
Degradation estimation by impulse characterization.
(a) An impulse of light (shown magnified).
(b) Imaged (degraded) impulse.

(3) 建模估计-I

□ 对退化过程背后的物理特性进行建模

a b
c d

FIGURE 5.25
Illustration of the
atmospheric
turbulence model.
(a) Negligible
turbulence.
(b) Severe
turbulence,
 $k = 0.0025$.
(c) Mild
turbulence,
 $k = 0.001$.
(d) Low
turbulence,
 $k = 0.00025$.
(Original image
courtesy of
NASA.)



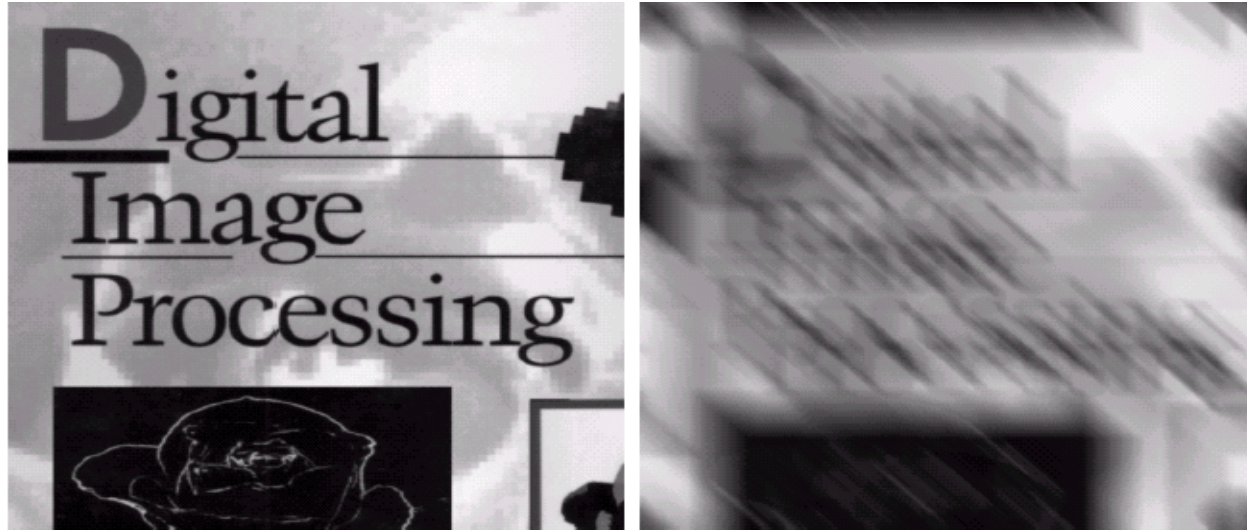
$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$

(3) 建模估计-II

- 从基本原理推导数学模型

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt$$

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua + vb)} \quad x(t) = at / T; \quad y(t) = bt / T$$



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

5.7 逆滤波

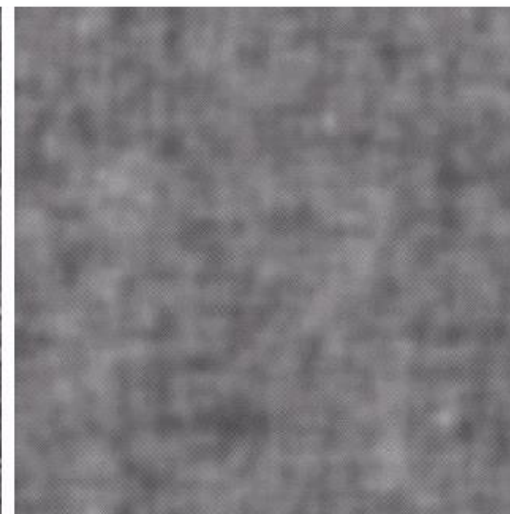
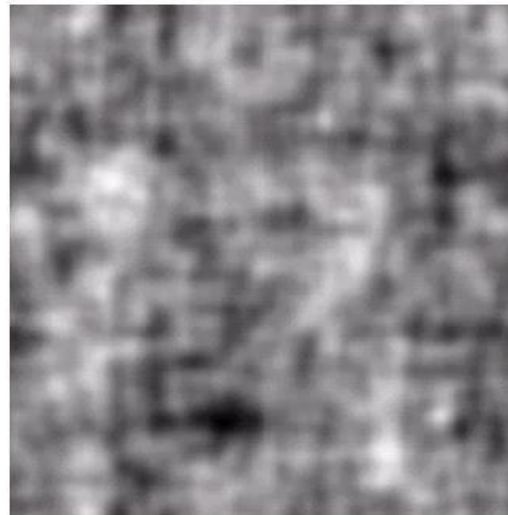
- 最简单且粗糙的恢复方法
- 降质图像频谱直接除以降质函数：

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \quad \Rightarrow \quad \hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

- 由于 $N(u, v)$ 未知，即使得到降质函数 $H(u, v)$ ，也难以精确重建
- $H(u, v)$ 在某些位置为0或者是非常小的值， $F(u, v)$ 被淹没
 - 解决方法：限制滤波的频率，使其接近原点

a b
c d

FIGURE 5.27
Restoring
Fig. 5.25(b) with
Eq. (5.7-1).
(a) Result of
using the full
filter. (b) Result
with H cut off
outside a radius of
40; (c) outside a
radius of 70; and
(d) outside a
radius of 85.



5.8 最小均方误差 (维纳) 滤波

- 确定未污染图像 f 的一个估计 \hat{f} , 使得他们之间的均方误差最小

$$e^2 = E\{(f - \hat{f})^2\}$$

- 最小化上述均方误差, 可得频域解:

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v)$$

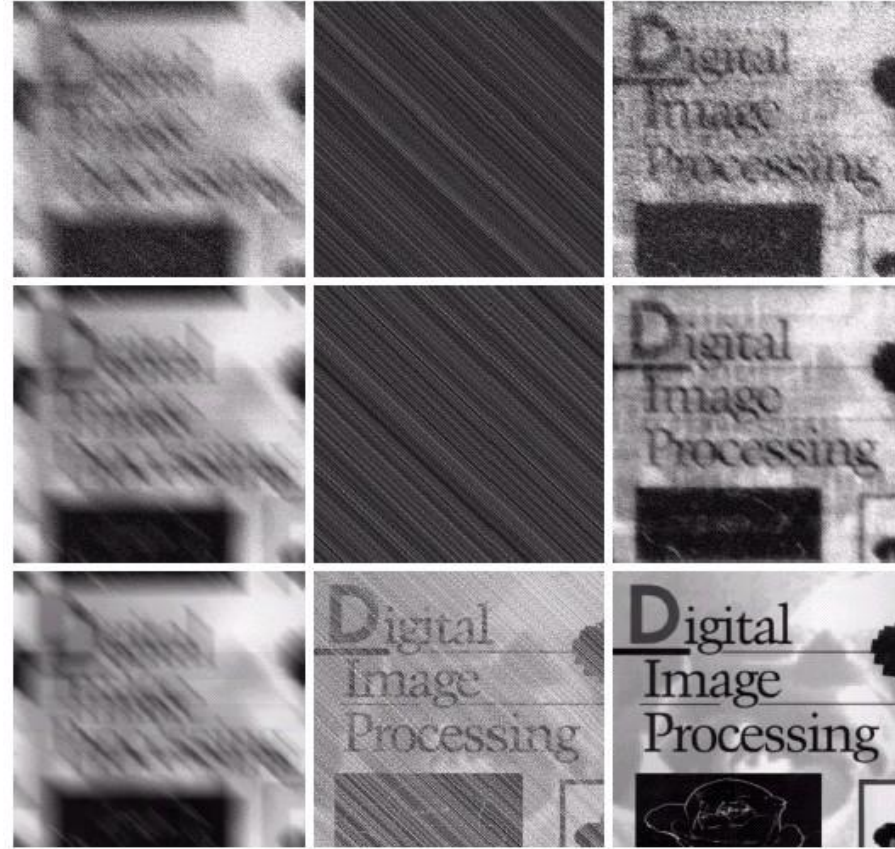
逆滤波与维纳滤波比较



a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

进一步比较

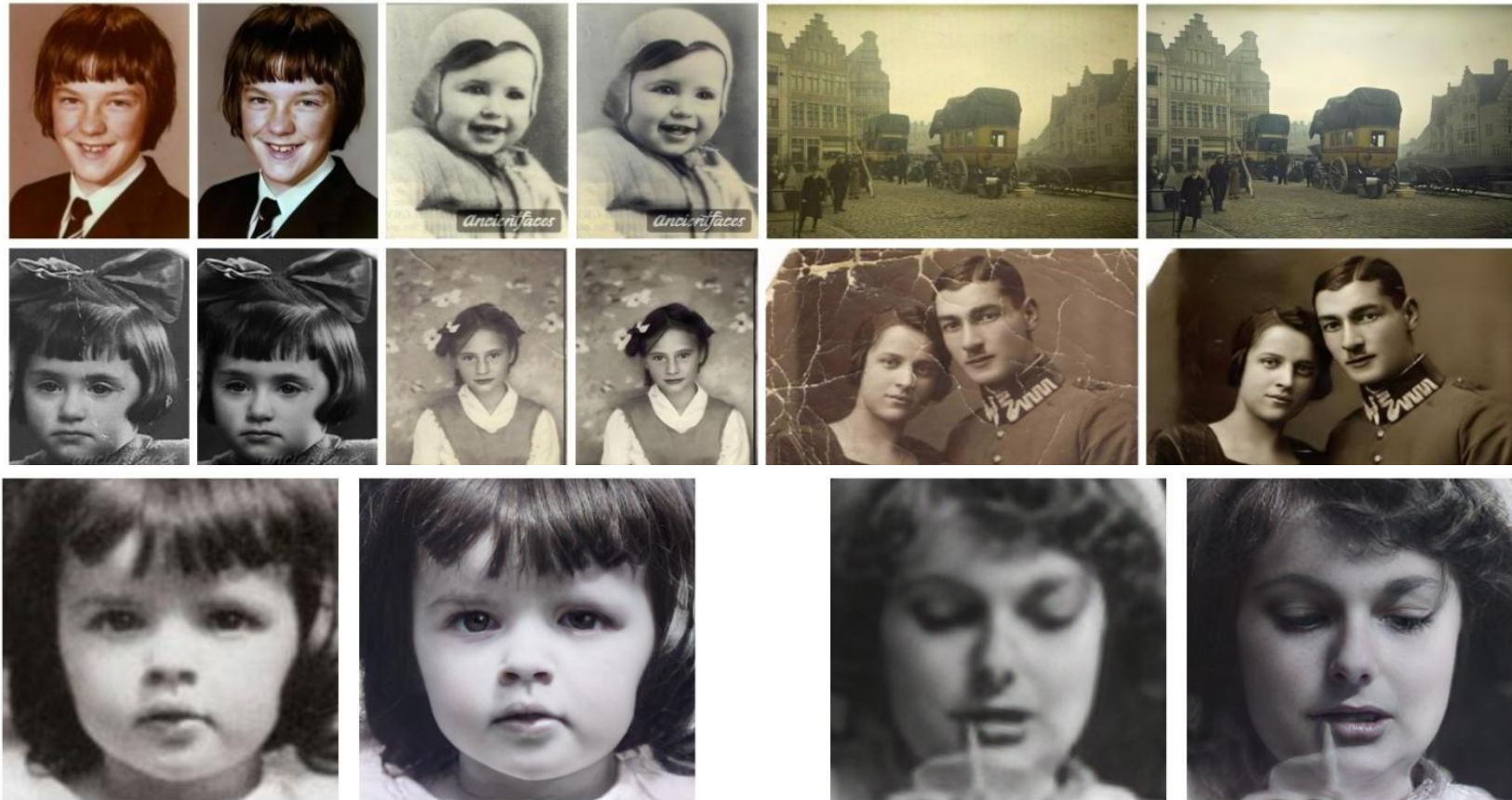


a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a “curtain” of noise.



Old Photo Restoration (Official PyTorch Implementation)

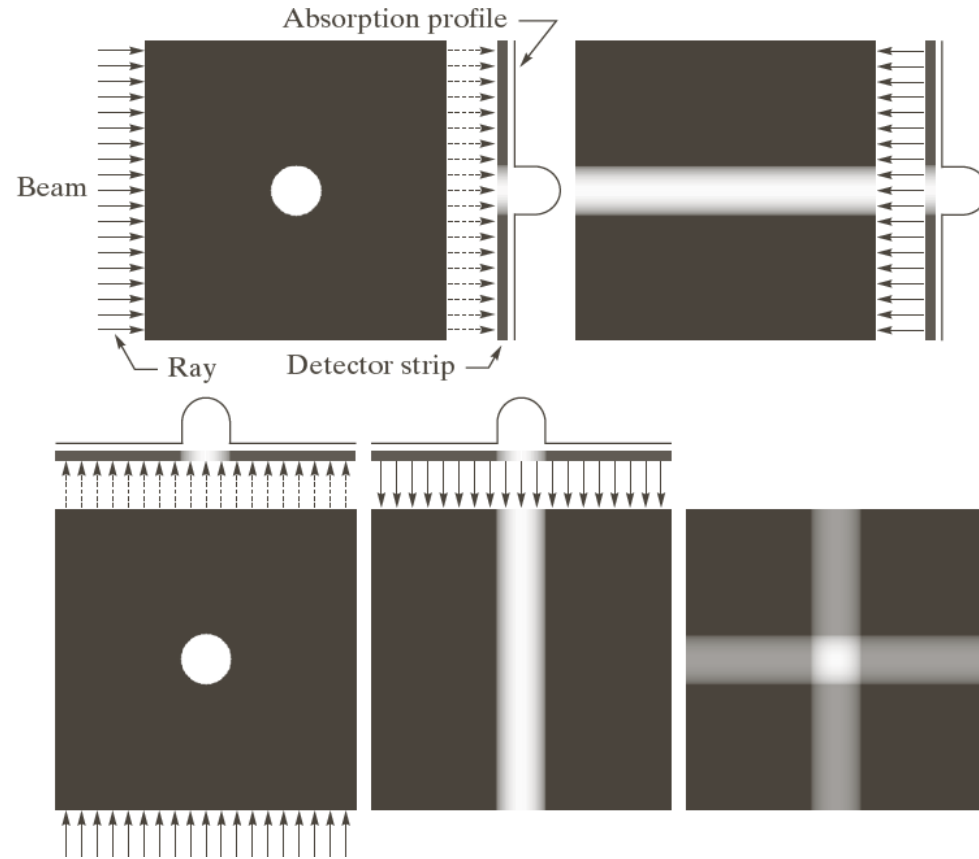


5.10 由投影重建图像

计算机断层(CT)原理

a b
c d e

FIGURE 5.32
 (a) Flat region showing a simple object, an input parallel beam, and a detector strip.
 (b) Result of back-projecting the sensed strip data (i.e., the 1-D absorption profile).
 (c) The beam and detectors rotated by 90°.
 (d) Back-projection.
 (e) The sum of (b) and (d). The intensity where the back-projections intersect is twice the intensity of the individual back-projections.



5.10 由投影重建图像

计算机断层(CT)原理

a	b	c
d	e	f

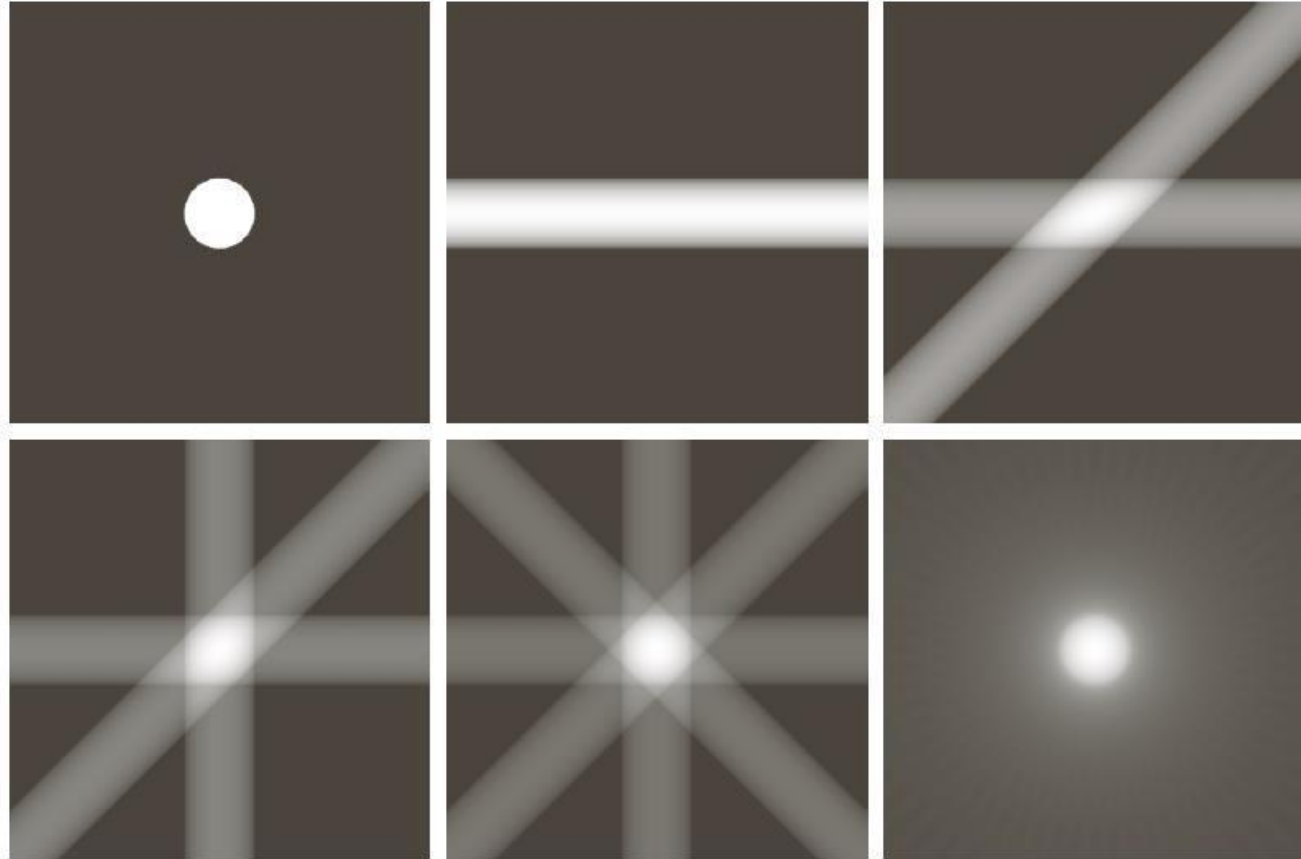
FIGURE 5.33

(a) Same as Fig. 5.32(a).

(b)–(e)

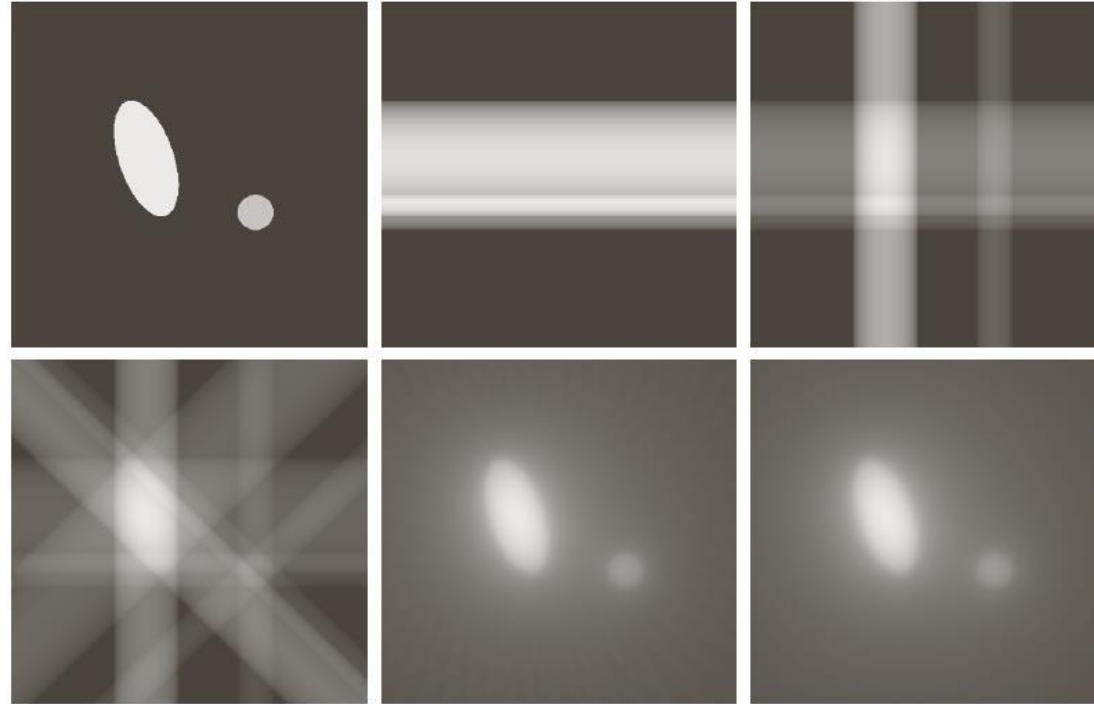
Reconstruction using 1, 2, 3, and 4 backprojections 45° apart.

(f) Reconstruction with 32 backprojections 5.625° apart (note the blurring).



5.10 由投影重建图像

计算机断层(CT)原理



a b c
d e f

FIGURE 5.34 (a) A region with two objects. (b)–(d) Reconstruction using 1, 2, and 4 backprojections 45° apart. (e) Reconstruction with 32 backprojections 5.625° apart. (f) Reconstruction with 64 backprojections 2.8125° apart.

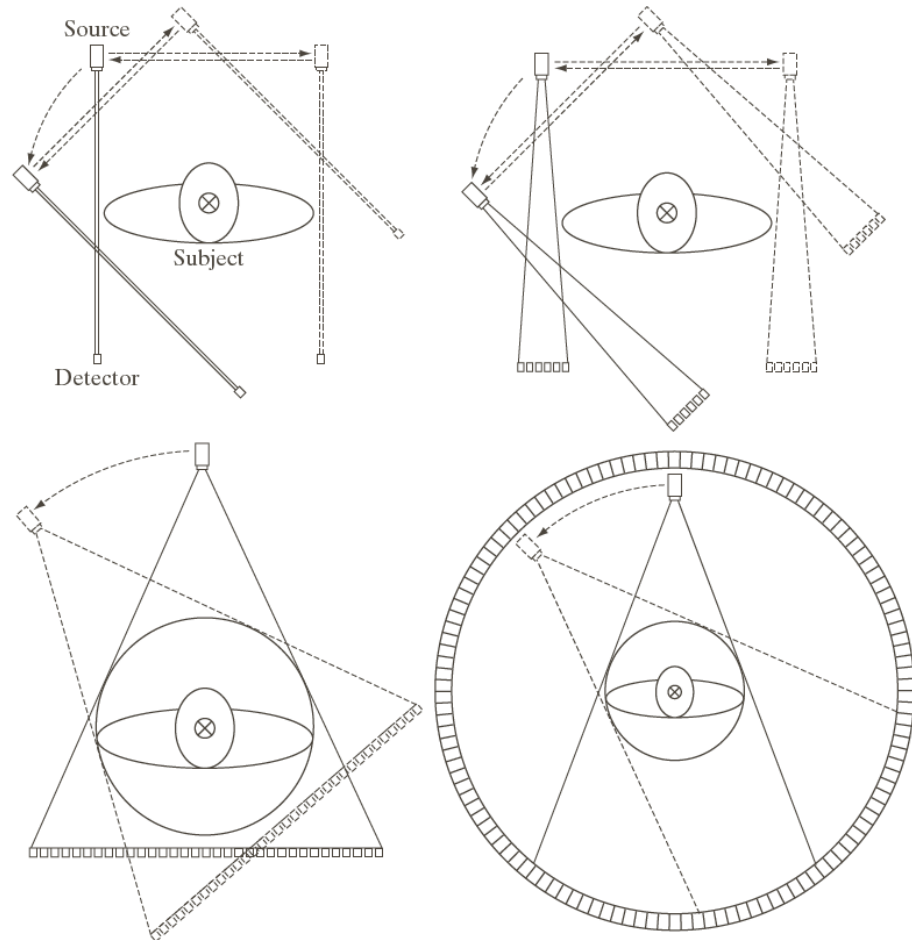
5.10 由投影重建图像

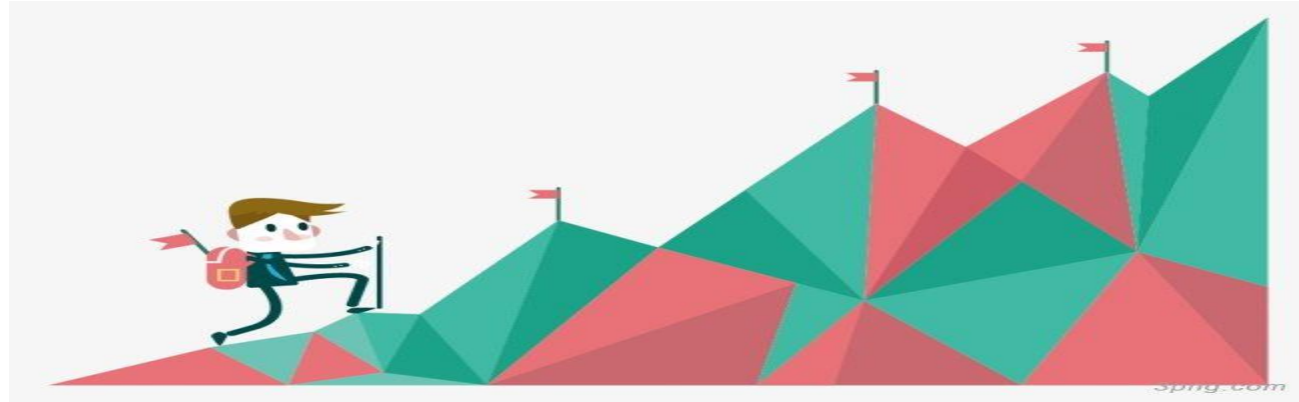
计算机断层(CT)原理

- 1919 Johann Rodon提出 **Radon变换**
- 1962年 Allan M. Cormack, Tufts Univ. 设计CT原型
- G. N. Hounsfield, 英国伦敦EMI公司工程师, 同时设计第一台医用CT
- A. M. Cormack和G. N. Hounsfield获1979年诺贝尔医学奖

a b
c d

FIGURE 5.35 Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.







数字图像处理 (Digital Image Processing)

王正

Email: zheng_wang@seu.edu.cn
办公室: 无线谷A5415

**Image
Processing
Project Ideas**
Beginners



图像压缩

- 图像压缩是一种减少描绘一幅图像所需数据量的技术和科学，它是数字图像处理领域最有用、商业上最成功的技术之一
 - 压缩的本质：用尽可能少的数据表达尽可能多的信息
 - 图像压缩必要性实例
 - ✓ 一部两小时的标清电影，分辨率 720×480 ，全彩色（24bit/pixel）
 - ✓ 如果不压缩，存储需要224GB！



原始图像
分辨率： 3000×2000
大小： $3000 \times 2000 \times 3\text{Byte} = 18\text{MB}$



压缩后的JPEG图像
大小： $< 1\text{MB}$



2010年10月发射的“嫦娥二号”探月卫星的天地通信采用基于JPEG 2000的图像压缩算法，速率约达每秒6M bits，像素分辨率是1米/像素。月球表面积约为3800万平方公里，如采用上述方法，传递整个月球表面的影像全时工作约需两年左右。

◆ 图像压缩目的

图像数据的显著特点：数据量非常大，视频影像由于连续播放，数据量更加庞大，对计算机的**存储**以及**网络传输**都造成了极大的负担。

解决办法：就是**进行图像压缩（编码）**，压缩后再进行存储和传输，到需要时再解压、还原。

出发点：像素与像素之间在行方向和列方向都具有很大的相关性，我们说整体数据的**冗余度很大**，需要对图像数据进行很大程度的压缩。

◆ 图像编码分类

1. 信息保持编码（无损编码）

它降低了压缩图像的比特数而不丢失任何信息。

2. 保真度编码（有损编码）

若接收端是人，因人眼的生理特性不需过高的空间和灰度分辨率，压缩中可丢失一些人感觉不到的信息，多媒体中常用这种压缩。

3. 特征编码（感兴趣区域编码）

目的是用于计算机的识别、分析、控制，并不需要全部细节，只要能保留相关特征信息。

原始图像

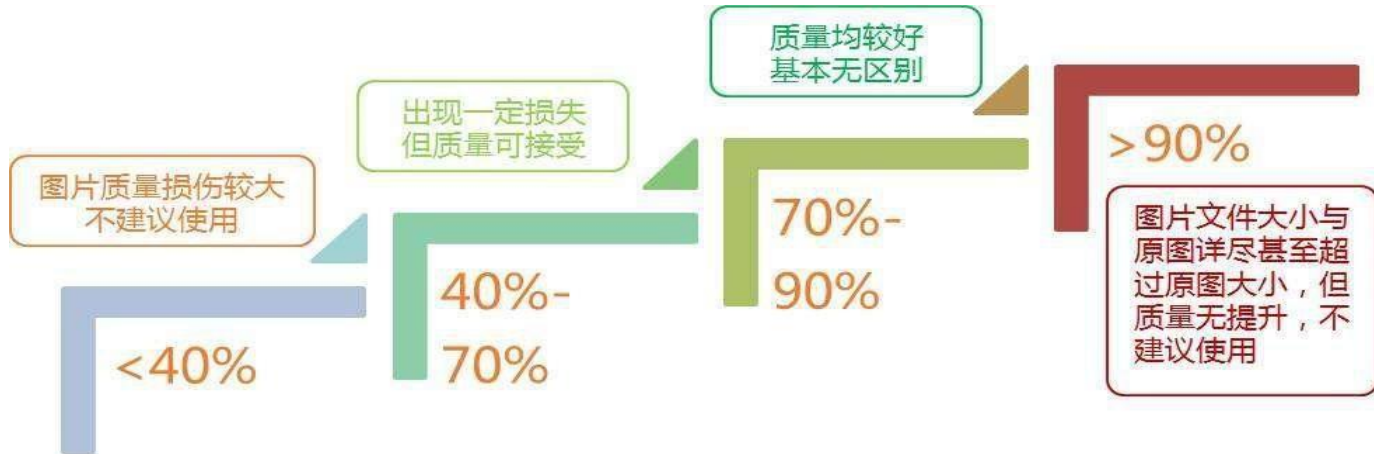


编码结果

```

10101010110101000001001
00100010111110010101111
11111111000000000000001
00000111101010000000000
1111111110101011011011
00000000000011011000000
00100000101010000000000
    
```

解码图像



□ 基础知识

- 编码冗余、空间冗余和时间冗余、不相关的信息量、图像信息的度量、保真度准则、图像压缩模型、图像格式容器和压缩标准

□ 一些基本的压缩方法

- 霍夫曼编码
- 算术编码
- LZW编码
- 行程编码
- 基于符号的编码
- 比特平面编码
- 块变换编码
- 预测编码
- 小波编码

□ 数字图像水印

基础知识

□ 相对数据冗余 R

$$R = 1 - \frac{1}{C} = \frac{b - b'}{b}$$

压缩率 $C = b/b'$

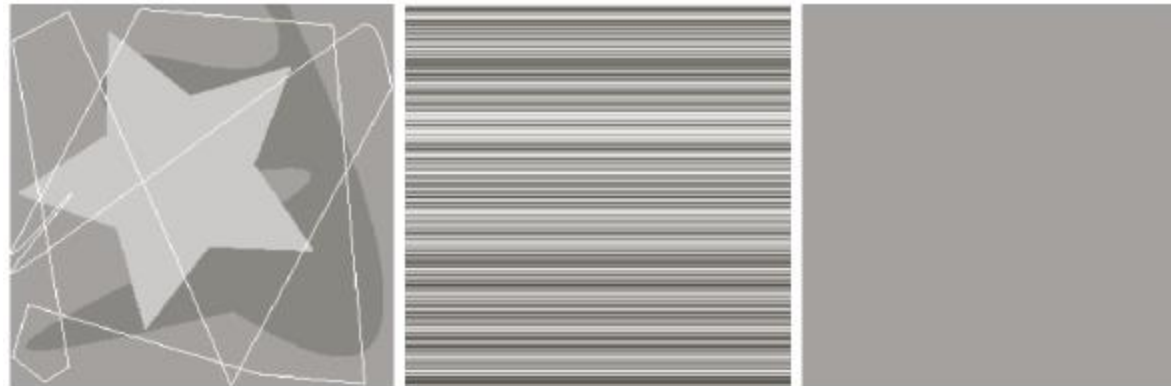
□ 三种主要类型的数据冗余:

- 编码冗余: 统计冗余
- 空间和时间冗余
- 不相关的信息: 感知冗余

编码冗余

空间冗余

无关信息



计算机产生的 $256 \times 256 \times 8$ 比特图像

编码冗余

- r_k 表示一副 $M \times N$ 图像的灰度, r_k 出现概率 $p_r(r_k)$

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, 2, \dots, L - 1$$

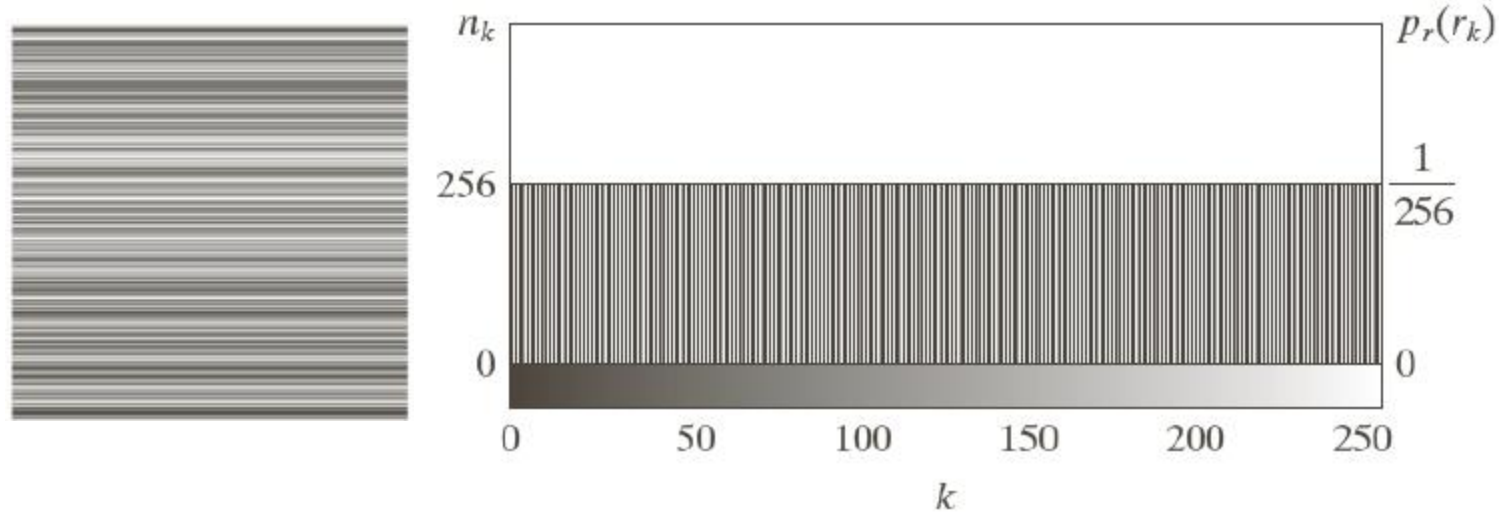
$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

- 变长编码: $L_{avg}=1.81$ bit
- 定长编码: $L_{avg}=8$ bit
- $C=8/1.81=4.42$ $R=1-1/C=0.774$

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

TABLE 8.1
Example of variable-length coding.

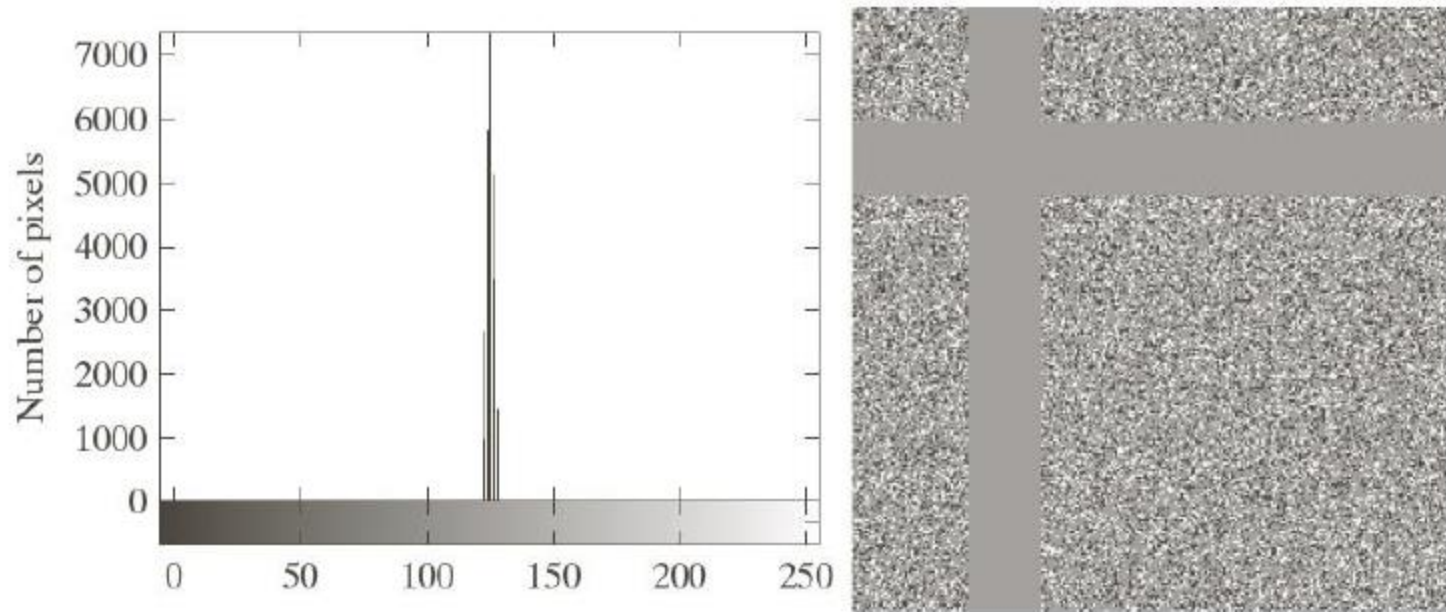
空间冗余和时间冗余



- 空间相关（x和y方向）：多数像素可根据相邻像素灰度进行合理预测
 - 行程编码：指定一个新的灰度开始点，以及该灰度持续像素的数量
- 时间相关：当该图像是视频序列的一部分时

不相关信息

- 数字图像压缩中，被人类视觉系统忽略的信息或与图像与其的应用无关的信息需要被删除



a b

FIGURE 8.3
(a) Histogram of the image in Fig. 8.1(c) and (b) a histogram equalized version of the image.

图像信息的度量

- 概率为 $P(E)$ 的随机事件包含的信息

$$I(E) = -\log P(E)$$

- 信源的平均信息称为该信源的熵

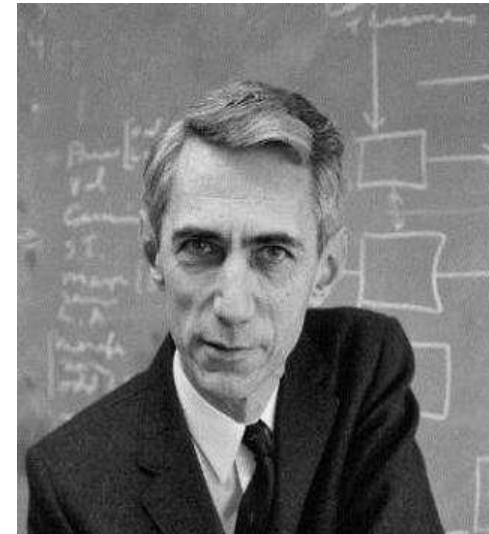
$$H = -\sum_{j=1}^J P(a_j) \log P(a_j)$$

- 灰度信源的熵

$$\tilde{H} = -\sum_{k=0}^{L-1} P_r(r_k) \log_2 p_r(r_k)$$

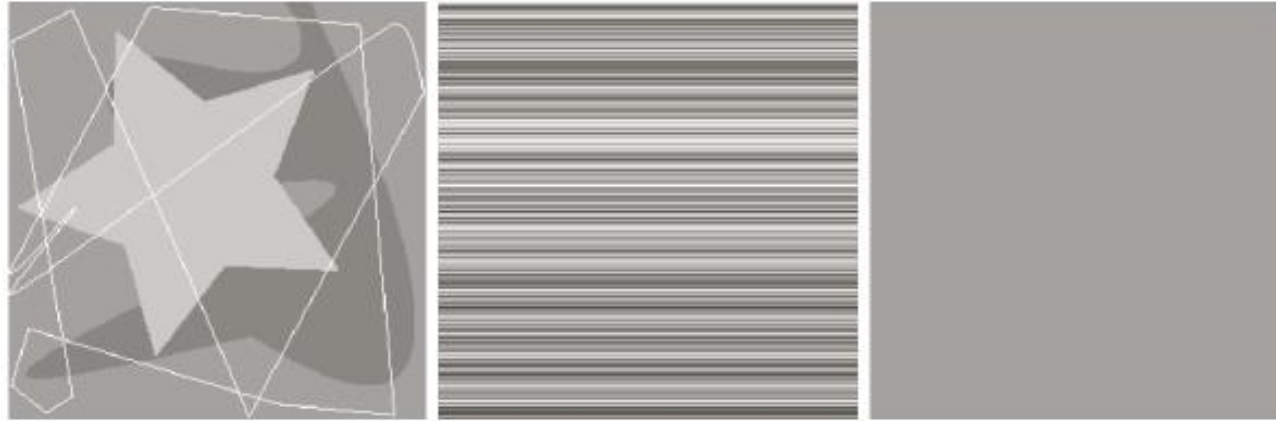
- 香农第一定理（无噪声编码定理）

$$\lim_{n \rightarrow \infty} \left[\frac{L_{avg,n}}{n} \right] = H$$



L_{avg} 是表示所有 n 个符号组
所需的编码符号的平均数量

- 一幅图像中的熵量和信息量与直觉相差很远



H=1.66像素/比特

H=8像素/比特

H=1.56像素/比特

$$\begin{aligned}\tilde{H} &= -[0.25 \log_2 0.25 + 0.47 \log_2 0.47 + 0.25 \log_2 0.25 + 0.03 \log_2 0.03] \\ &= -[0.25(-2) + 0.47(-1.09) + 0.25(-2) + 0.03(-5.06)] \\ &\approx 1.6614 \text{ bits/pixel}\end{aligned}$$

保真度准则：度量压缩质量

- $f(x, y)$ 和 $\hat{f}(x, y)$ 之间误差 $e(x, y)$

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- 平均误差的平方根

$$e_{rms} = \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{\frac{1}{2}}$$

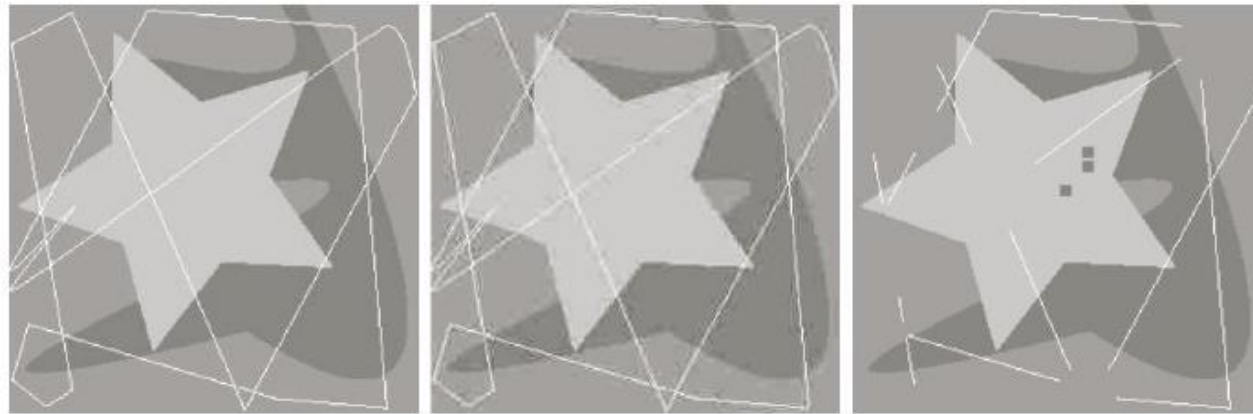
- 均方信噪比

$$\text{SNR}_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$

保真度准则

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

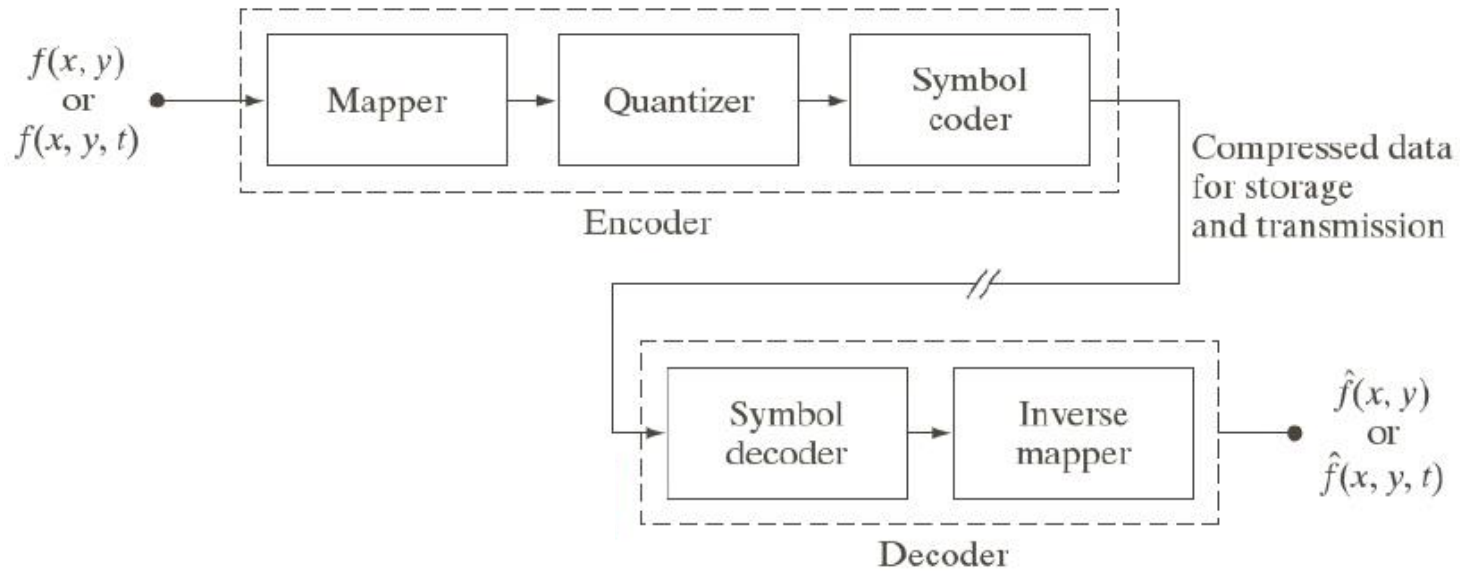
电视配置研究组织的等级尺度



EXAMPLE 8.3: Image quality comparisons.

Figure 8.4 shows three different approximations of the image in Fig. 8.1(a). Using Eq. (8-10) with Fig. 8.1(a) as $f(x,y)$ and Figs. 8.4(a) through (c) as $\hat{f}(x,y)$, the computed rms errors are 5.17, 15.67, and 14.17 intensity levels, respectively. In terms of rms error (an objective fidelity criterion), the images

图像压缩模型



□ 编码或压缩过程

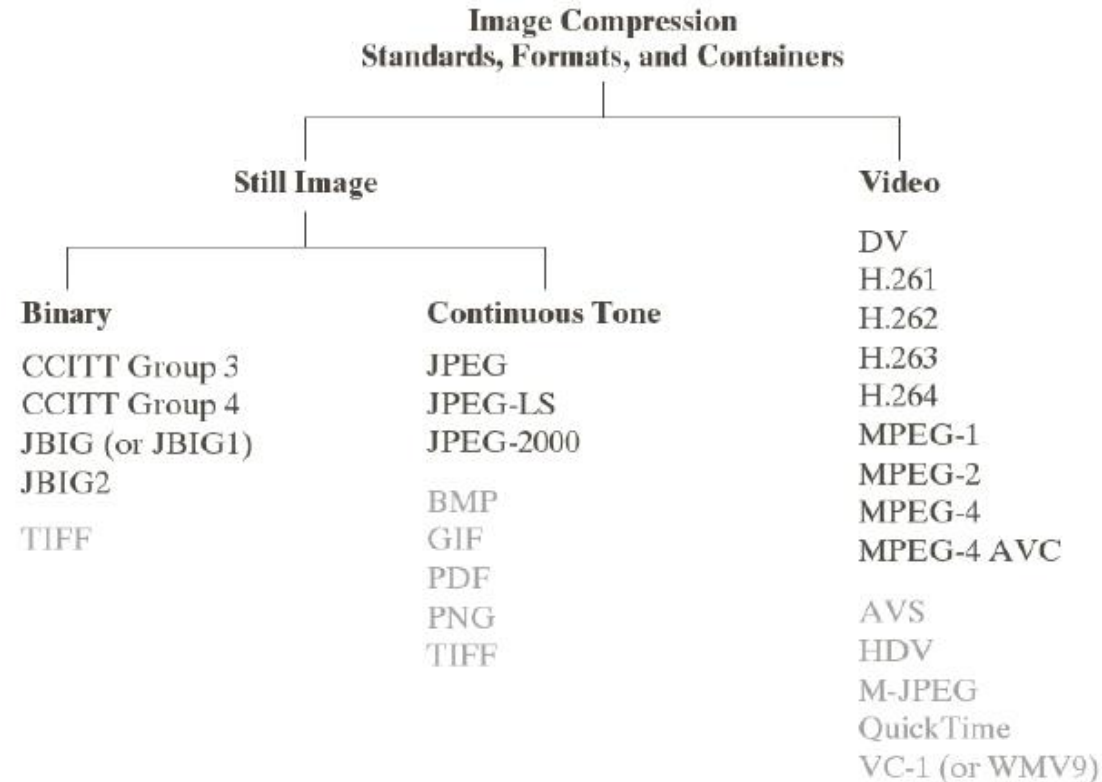
- 映射器：降低空间和时间冗余 可逆
- 量化器：根据预设保真度准则，排除压缩表示的无关信息
- 符号编码器：生成定长编码或变长编码，最小化编码冗余 可逆

□ 解码或解压缩过程

- 符号解码器和反映射器

图像格式、容器和压缩标准

- 文件格式：相当于高层的信息，定义了数据类型、压缩方法等
- 容器：相当于一个存放压缩以后的数据的东西
- 压缩方法：处在更底层，规定了数据的具体压缩的过程





8.2 一些基本的压缩方法

- 霍夫曼编码
- 算数编码
- LZW编码
- 行程编码
- 基于符号的编码
- 比特平面编码
- 块变换编码
- 预测编码
- 小波编码

霍夫曼编码

霍夫曼编码步骤:

(1) 概率排序, 缩减和简化信源符号数量

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	
a_1	0.1	0.1	0.2	0.3	0.4
a_4	0.1	0.1			
a_3	0.06	0.1	0.1		
a_5	0.04				

在信源数据中出现概率越大的符号, 编码以后相应的码长越短; 出现概率越小的符号, 其码长越长, 从而达到用尽可能少的码符表示信源数据。它在无损变长编码方法中是最佳的

霍夫曼编码

霍夫曼编码步骤:

(2) 对每个简化后的信源进行编码, 从最小信源开始, 直到遍历原始信源

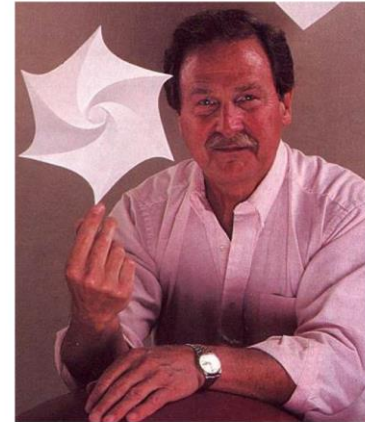
Original source		Source reduction								
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

平均长度

$$L_{avg} = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5 = 2.2$$

- 哈夫曼编码(Huffman Coding), 又称霍夫曼编码
 - 1951年, MIT的哈夫曼同学正要面对令人头疼的期末, 老师告诉他, 如果他能完成一个让老师满意的学期报告, 就可以不用参加考试。
 - 哈夫曼同学果断选择了学期报告! 报告的题目是“寻找最有效的二进制编码”。
 - 不久后哈弗曼同学交上了一篇名为“可能是最好的二进制编码”的报告, 老师看完后皱了皱眉: “你这名字起的不好, 得把“可能是”去掉。”

David A. Huffman (1925-1999)



<http://www.huffmancoding.com/my-uncle/scientific-american>
(Photo by Matthew Mulbry, originally for *Scientific American*, Sep. 1991)

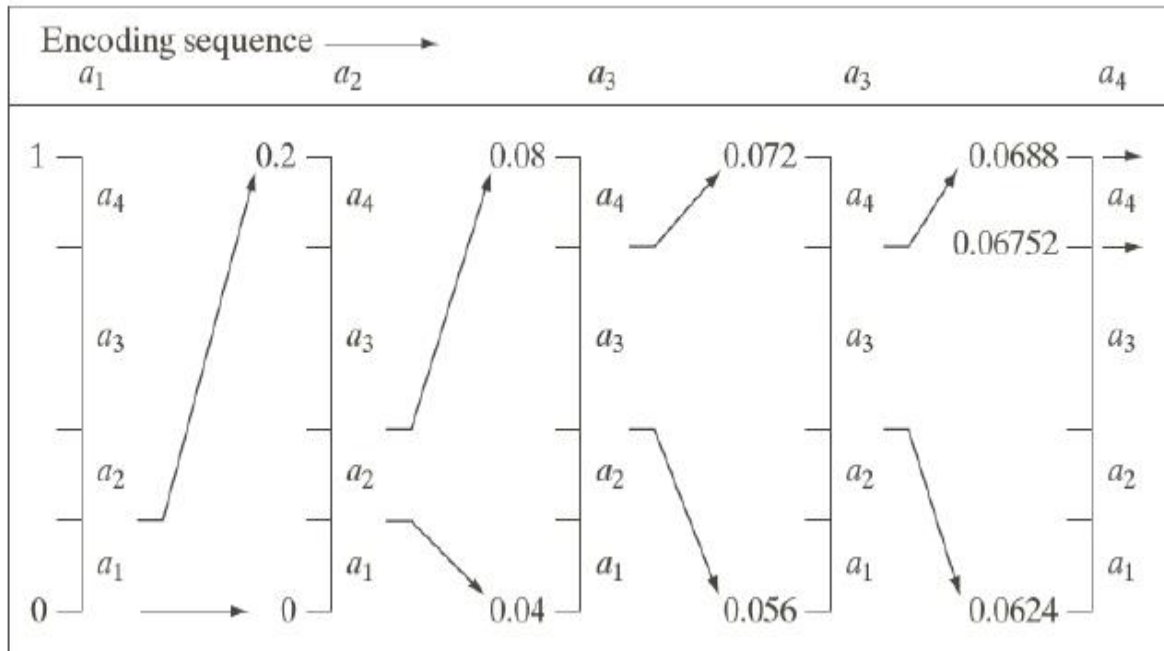
算术编码

- 将一个信源符号序列编码为一个介于0到1之间的实数
- 示例：对来自一个4符号信源的5符号序列进行编码

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

信源

算术编码过程



随着编码序列长度增大，得到的算术编码会接近香农第一定理的极限，然而存在两个实际问题使得编码性能无法达到这个极限

1. 需要增加一个消息结束指示器来分隔不同的消息
2. 算法精度有限



LZW编码

- 无误差压缩图像中空间冗余的方法
- 将定长码字分配各变长信源符号序列

LZW编码 (Encoding) 的核心思想其实比较简单，就是把出现过的字符串映射到记号上，这样就可能用较短的编码来表示长的字符串，实现压缩，例如对于字符串：

```
ABABAB
```

可以看到子串AB在后面重复出现了，这样我们可以用一个特殊记号表示AB，例如数字2，这样原来的字符串就可以表示为：

```
AB22
```

这里我们称2是字符串AB的记号(Symbol)。那么A和B有没有记号来表示？当然有，例如我们规定数字0表示A，数字1表示B。实际上最后得到的压缩后的数据应该是一个记号流 (Symbol Stream)：

```
0122
```




字典的生成

这里有一个问题：为什么第一个AB不用2表示？即表示为222，这样不又节省了一个记号？这个问题实际上引出的是LZW的一个核心思想，即压缩后的编码是**自解释** (self-explaining) 的。什么意思？即字典是**不会被写进压缩文件的**，在解压缩的时候，一开始字典里除了默认的0->A和1->B之外并没有其它映射，2->AB是在解压缩的过程中一边加入的。这就要求压缩后的数据自己能告诉解码器，完整的字典，例如2->AB是如何生成的，在解码的过程中**还原**出编码时用的字典。

用上面的例子来说明，我们可以想象ABABAB编码的过程：

1. 遇到A，用0表示，编码为0。
2. 遇到B，用1表示，编码为1。
3. 发现了一个子串AB，添加映射2->AB到字典里。
4. 后面又出现了AB子串，都用2来编码。

以上过程只是一个概述，并非真正LZW编码过程，只是为了表示它的思想。可以看出最前面的A和B是用来生成表项2->AB的，所以它们必须被保留在压缩编码里，作为表项2->AB生成的“第一现场”。这样在解码0122的时候，解码器首先通过01直接解析出最前面A和B，并且生成表项2->AB，这样才能将后面出现的2都解析为AB。实际上解码器是自己还原出了编码时2->AB生成的过程。

编码和解码都是从前往后步步推进的，同时生成字典，所以解码的过程也是一个不断还原编码字典的过程。解码器一边解码，向后推进，一边在之前已经解出的原始数据上重现编码的过程，构建出编码时用的字典。