

# 第3章

# 第3章 离散时间信号的傅里叶变换

3.1 CTFS, CTFT

3.2 DTFT

3.3 CT信号的抽样

3.4 DTFS, DFS

3.5 DFT 重点内容

3.6 用DFT计算线性卷积

3.7 与DFT有关的几个问题

3.8 二维傅里叶变换

3.9 Hilbert 变换

为了  
引出  
DFT

傅立叶变换是信号分析  
与处理的基本工具

## 3.8 二维傅立叶变换(2)

多用于图像处理

通过采样和  
量化，得到  
数字图像，  
矩阵表示

图像平面的原点：(0,0)

$$x(n_1, n_2) = \begin{bmatrix} x(0,0) & x(0,1) & \cdots & x(0, N_2 - 1) \\ x(1,0) & x(1,1) & \cdots & x(1, N_2 - 1) \\ \vdots & \vdots & \vdots & \vdots \\ x(N_1 - 1, 0) & x(N_1 - 1, 1) & \cdots & x(N_1 - 1, N_2 - 1) \end{bmatrix}$$

阵列的每个元素，称为图像元素，图画元素，或像素

MATLAB程序设计里， $x(0,0) \rightarrow x(1,1)$ ，或 $f(1,1)$

$M$ 行 $\times$  $N$ 列

$$X(z_1, z_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) z_1^{-n_1} z_2^{-n_2}$$

$$X(e^{j\omega_1}, e^{j\omega_2}) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}$$

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2}$$



先对行作DFT，作 $N_1$ 次，对其中间结果，  
再对列作变换，作 $N_2$ 次。或反之。

# 图像增强

图像处理最基本的目的之一是改善图像，改善图像最常用的技术就是图像增强

## 造成图像质量的损伤的途径

- 图像传输等过程中由于信道带宽的限制所造成的图像模糊；
- 与图像无关的加性噪声，如信道、摄象机的热噪声；
- 与图像有关的噪声干扰，如乘性噪声；
- 量化噪声、背景干扰噪声；
- 信号减弱、对比度下降；
- 成像条件不好。

图像增强有两大类应用：改善图像的视觉效果；突出图像的特征以便于计算机处理

按作用域分为两类：空域处理（包括图像灰度变换、空间滤波）和频域处理（包括频域滤波、同态滤波、小波变换滤波，等）

空域处理是直接对图像像素进行处理；频域处理则是在图像的某个变换域内，对图像的变换系数进行运算，然后通过逆变换获得图像增强效果



**(a)** 原始图像



**(b)** 3%椒盐噪声  
干扰的噪声图像



**(c)** 3%随机值脉冲噪  
声干扰的噪声图像

归纳起来，**图像质量退化的原因**

对比度问题，对比度局部或全部偏低，影响图像视觉；

噪声干扰问题，使图像蒙受干扰和破坏；

清晰度下降问题，使图像模糊不清，甚至严重失真。

针对图像的退化和不足，改善图像的质量以较好地**满足实际需要**，  
是**图像增强的目的**

**图像增强**并不去估计图像实际退化的过程和因素而加以矫正，而只是**考虑图像退化的一般性质，加以修正，以求得一般的或平均地图像质量的改进**

例如，边缘锐化去模糊、平滑去噪声、直方图修正对比度增强等

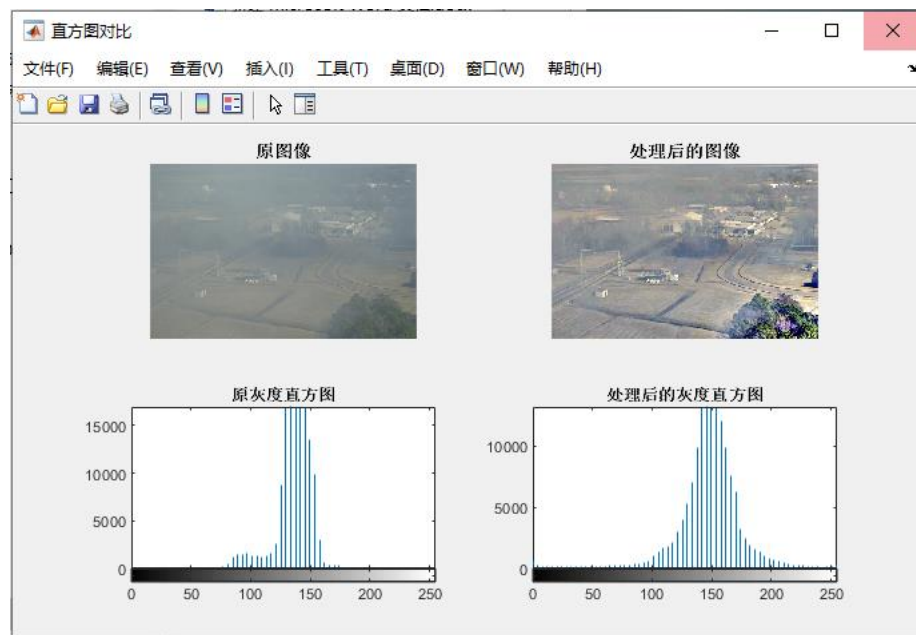
图像增强还可能为了人类视觉需要，使图像的内容更突出，更容易被获取，并不关心和原始图像是否一致，甚至人为地畸变原始图像，以达到视觉增强效果

例如，伪彩色图像增强：将不同灰度的图像赋以不同的彩色，以增强人类的视觉感知，在医学图像处理中经常采用；

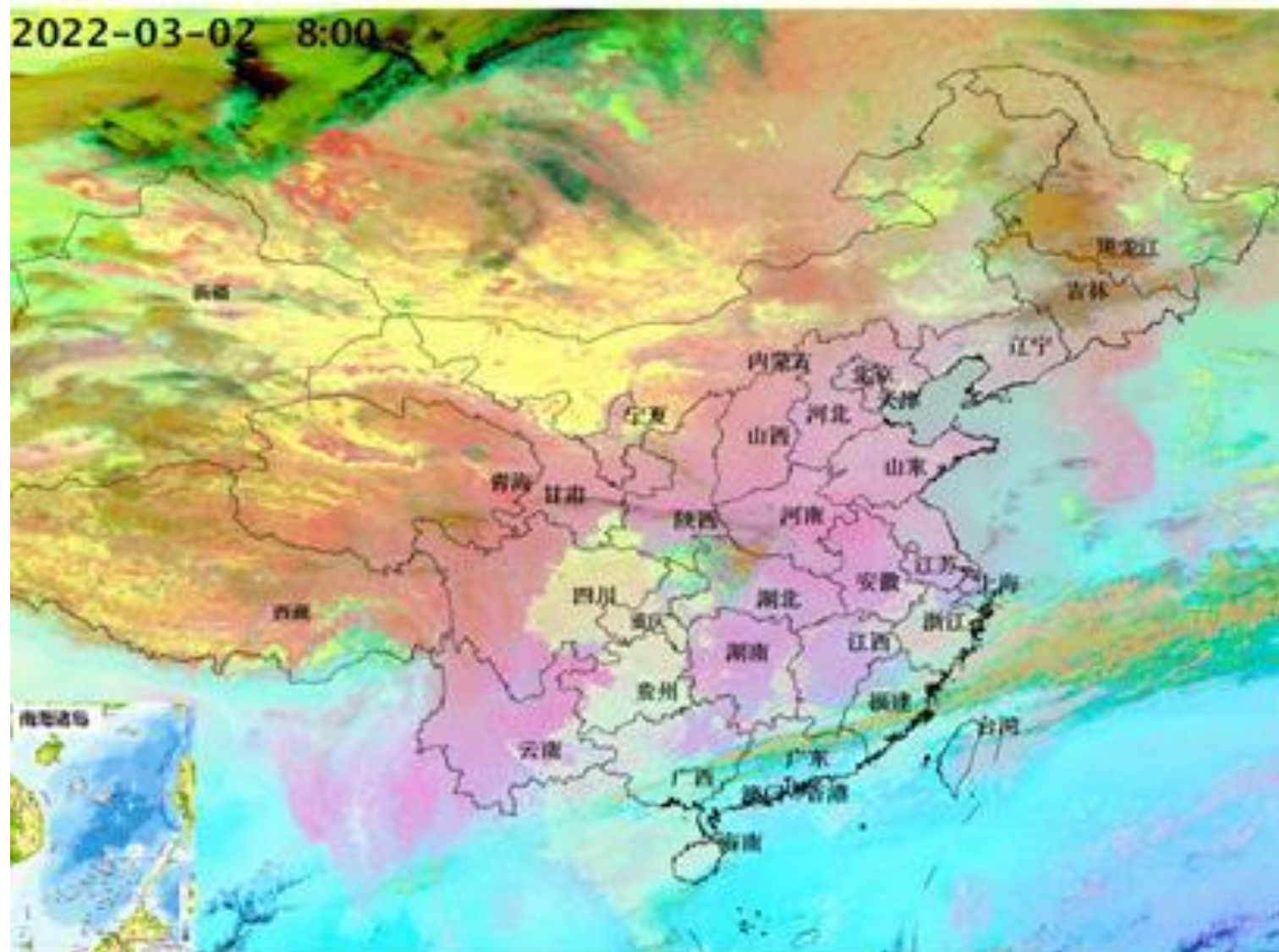
又如，假彩色增强：不同波段获取的图像赋以不同的彩色，用在多波段遥感图像的假彩色显示等

图像增强是综合和一般性地改善图像质量，解决图像由于噪声、模糊退化和对比度降低等三类问题，获得最好的视觉效果





2022-03-02 8:00



### 伪色彩图沙尘卫星监测

数据源: Himawari-8 AHI

时间分辨率: 10min

图例:

沙尘浓度<0.1	0.1-0.2	0.2-0.3	0.3-0.4
沙尘浓度>0.4	沙尘浓度>0.5	沙尘浓度>0.6	沙尘浓度>0.7



中国科学技术大学

环境光学遥感中心



# 图像灰度变换、空间滤波、频域滤波

## 空间滤波

空间滤波，又称为邻域处理或空间卷积

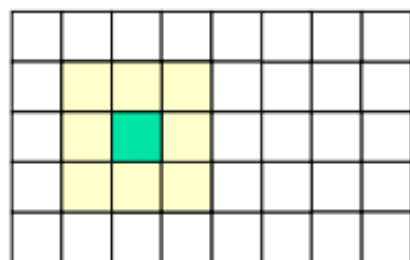
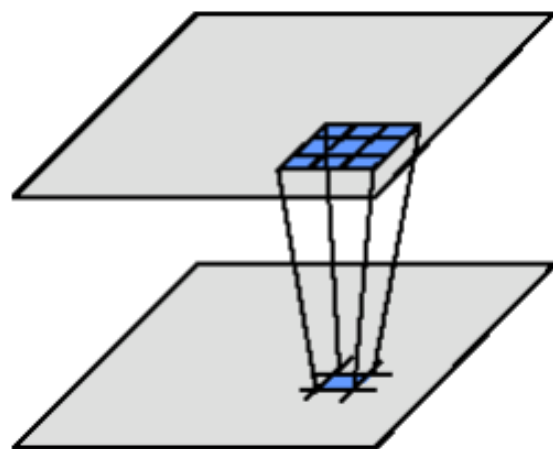
$$g(x, y) = T[f(x, y)]$$

$f(x, y)$ 为输入图像， $g(x, y)$ 为输出图像， $T$ 是在点 $(x, y)$ 的一个指定邻域上定义的对图像 $f$ 进行处理的算子；邻域大小表示为 $m \times n$ ，有意义的最小模板尺寸为 $3 \times 3$ ，涉及上下左右9个像素（参见下一个页面）；邻域内像素参与计算 $(x, y)$ 点的 $g$ 值。

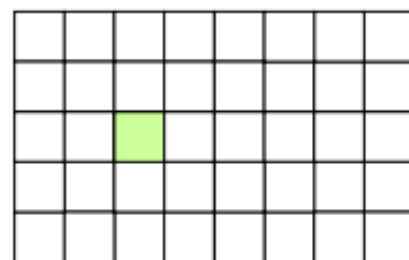
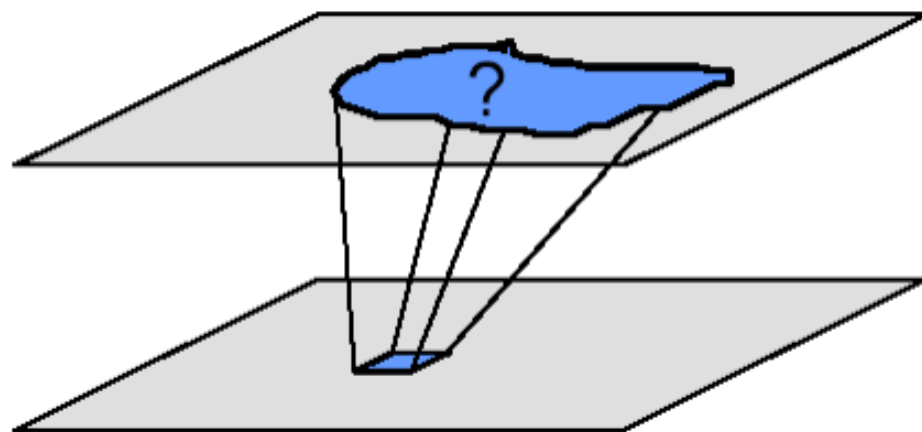
若对邻域 $m \times n$ 中像素进行线性计算，则该操作为线性空间滤波，或称空间卷积。对邻域中每个像素乘以相应的系数，再求和，以得到 $(x, y)$ 点处的响应，需要 $mn$ 个系数。

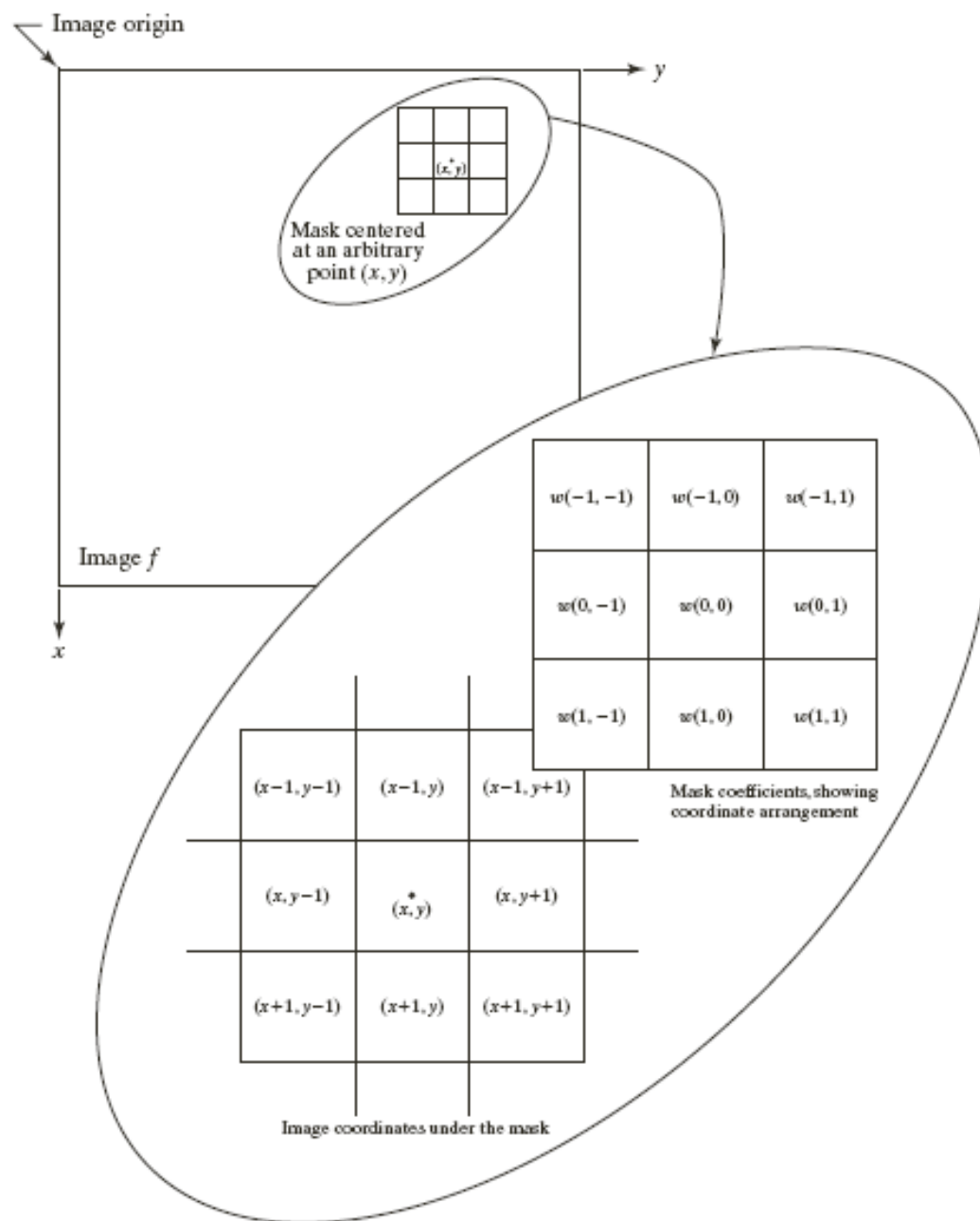
系数排列成一个矩阵，称之为滤波器、模板、滤波模板、核、掩膜或窗口，其中常见的是前三个术语。为更加明显些，也用卷积滤波、卷积模板、卷积核等术语。

规则映射



非规则映射





**FIGURE 2.13**

The mechanics of linear spatial filtering. The magnified drawing shows a  $3 \times 3$  filter mask and the corresponding image neighborhood directly under it. The image neighborhood is shown displaced out from under the mask for ease of readability.



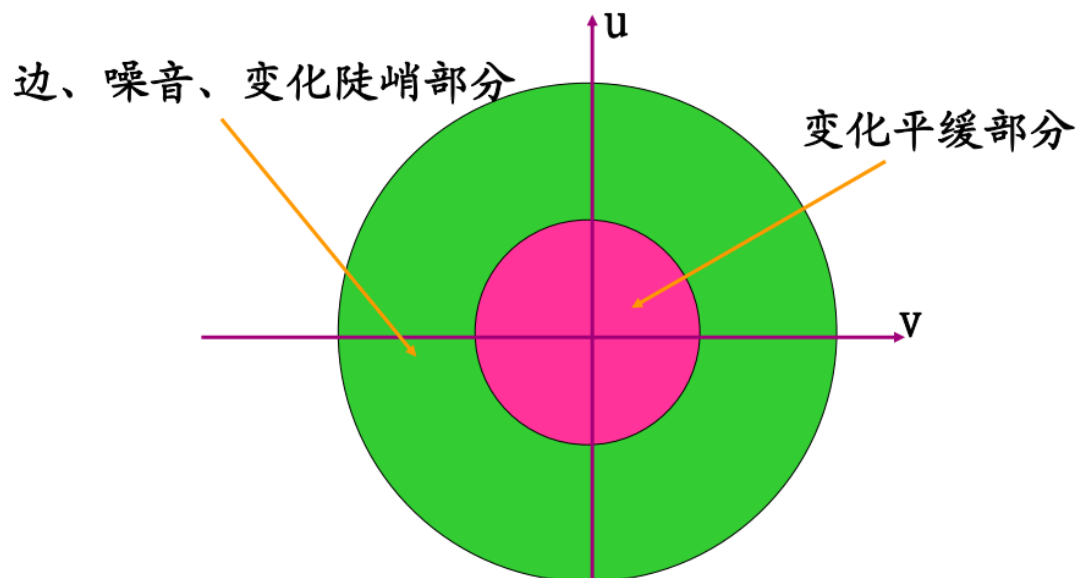
# 图像增强的频域法

频域增强的**理论基础**：**卷积理论**

频域增强的**原理**：频率平面与图像空域特性的关系

图像变化平缓的部分靠近频率平面的圆心，这个区域为低频区域

图像中的边、噪音、变化陡峭的部分，以放射方向离开频率平面的圆心，这个区域为高频区域



a b  
c d

**FIGURE 3.19**

High-frequency  
emphasis filtering.

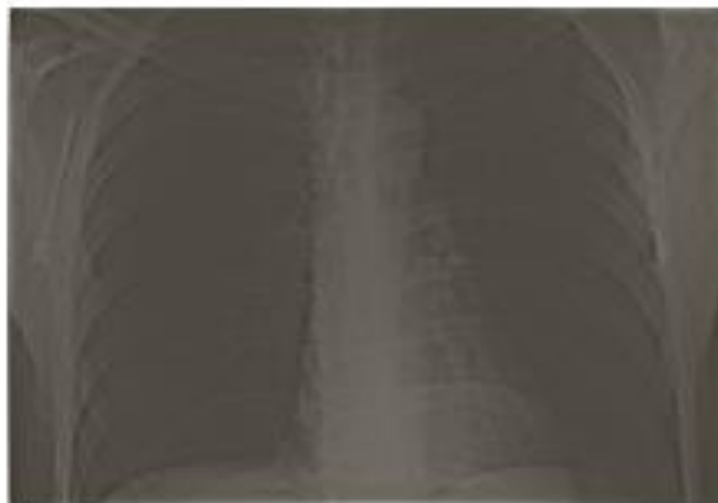
(a) Original  
image.

(b) Highpass  
filtering result.

(c) High-frequency  
emphasis result.

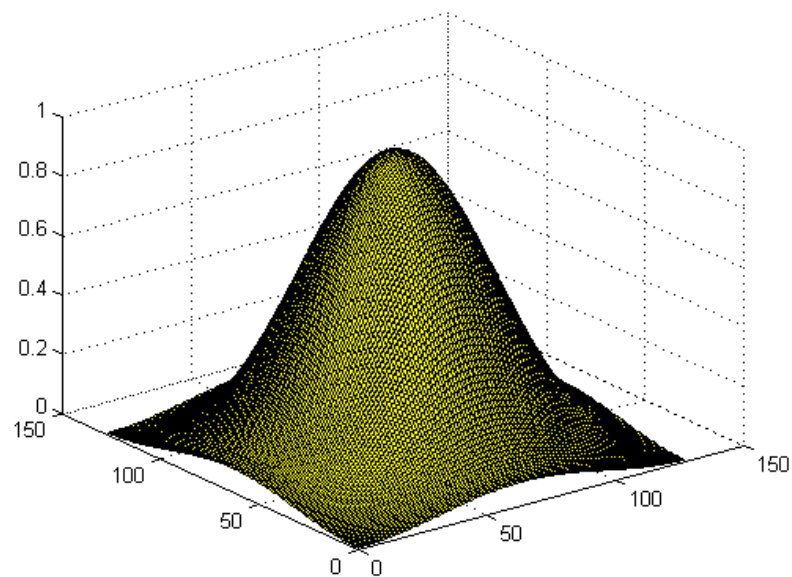
(d) Image (c) after  
histogram  
equalization.

(Original image  
courtesy of Dr.  
Thomas R. Gest,  
Division of  
Anatomical  
Sciences,  
University of  
Michigan Medical  
School.)

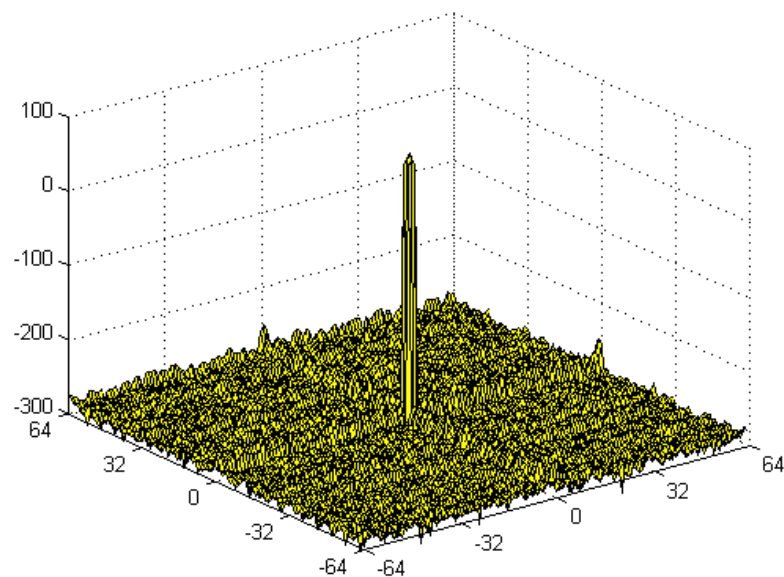


d: 联合使用高频强调滤波和直方图均衡的效果

## 例：2-D Hamming 窗及其频谱



时域窗（空域）



频谱



## 图像低通和高通滤波示例

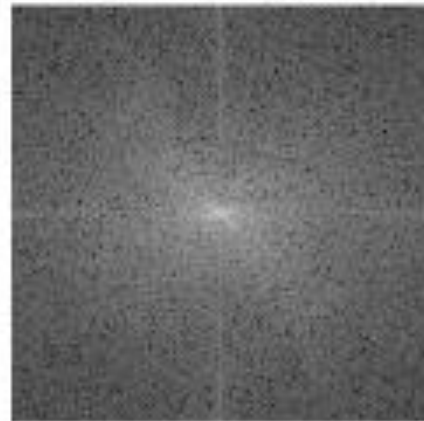
1. Src image



3. Image DFT



4. FFT Shift



2. Gray image



5. Low-pass filtering

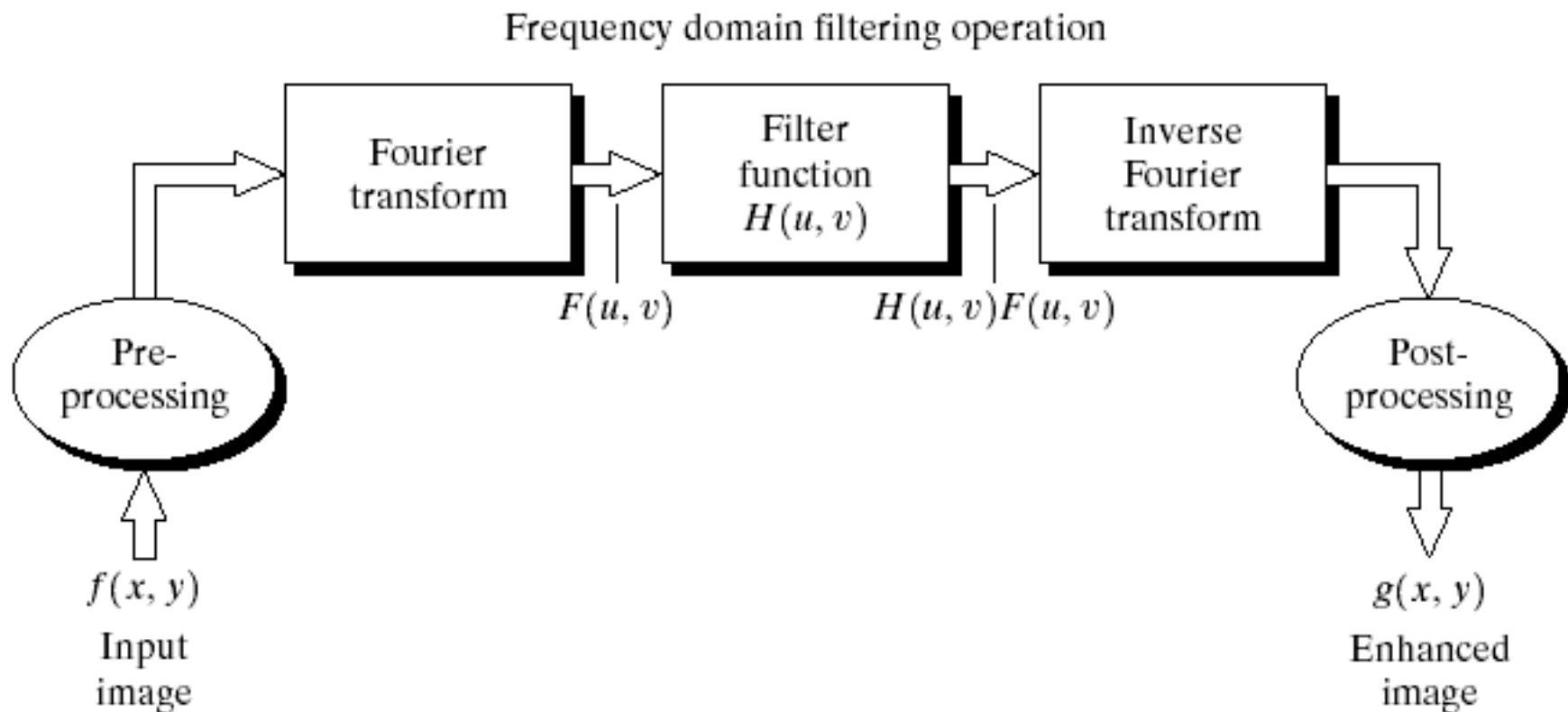


6. High-pass filtering



## 频域滤波的步骤

1. 计算图像的离散傅立叶变换 $F(u, v)$
2. 用滤波函数 $H(u, v)$ 乘以 $F(u, v)$
3. 计算乘积的离散傅立叶反变换



# 理想低通滤波器

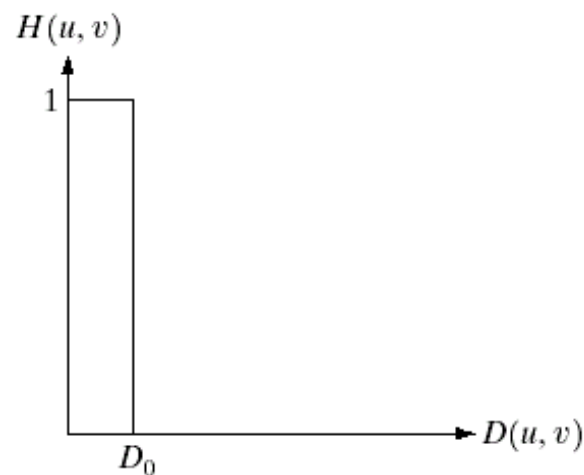
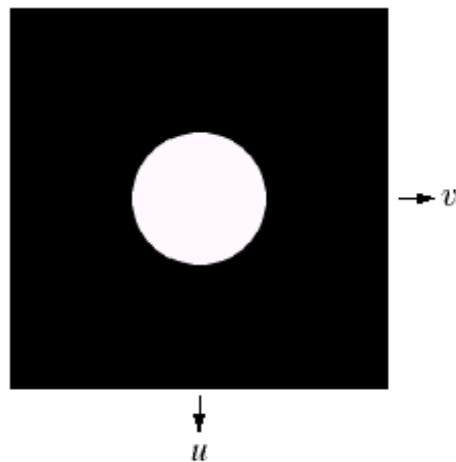
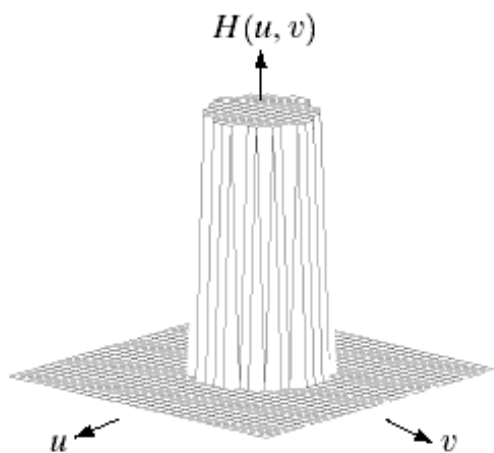
理想低通滤波器的传输函数:

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

$D_0$ : 截止频率, 通带的半径  
 $D(u, v)$ : 距离函数

其中 $D(u, v)$ 为:

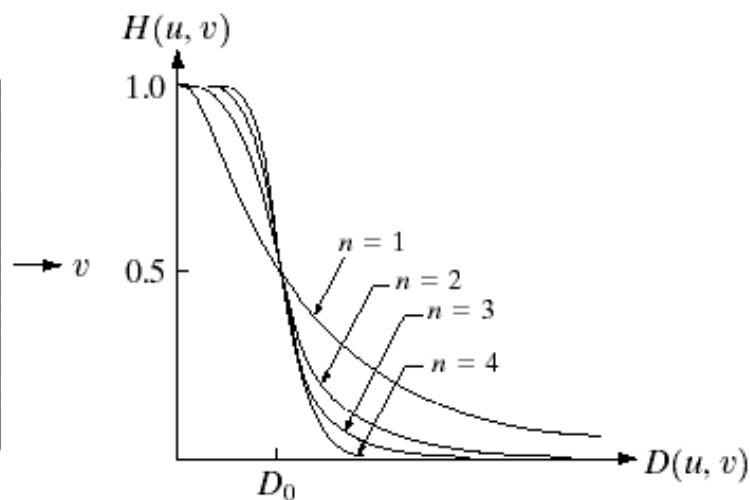
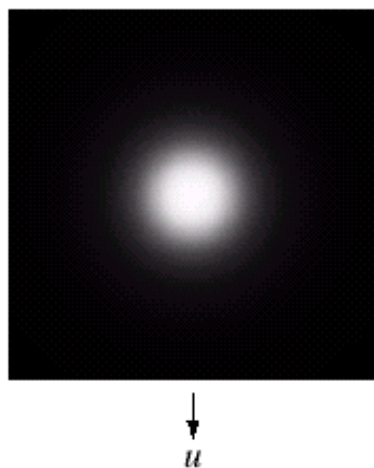
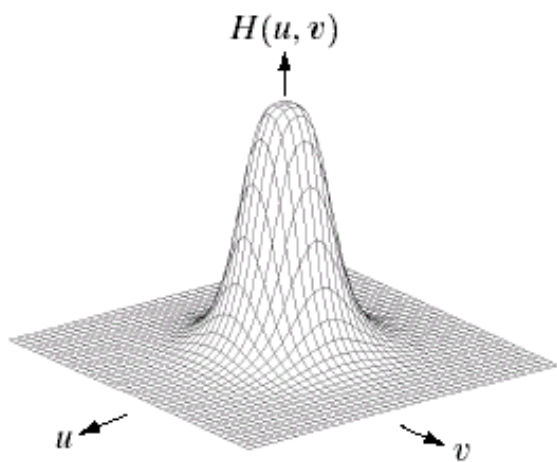
$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$



# 巴特沃斯低通与高通滤波器

截至频率为 $D_0$ 的 $n$ 阶Butterworth 低通滤波器传输函数定义为:

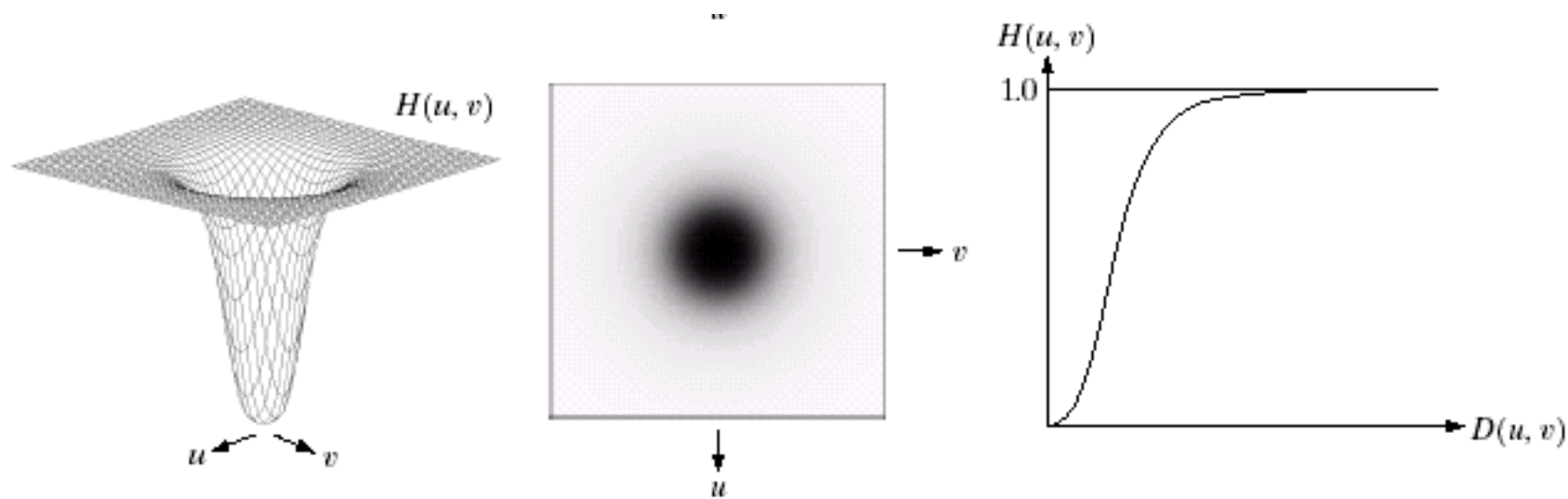
$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$



Butterworth 高通滤波器传输函数定义为:

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

其中,  $n$ 为阶数,  $D_0$ 为截至频率



# fftshift: 零频分量移到频谱中心

## 语法

$Y = \text{fftshift}(X)$

$Y = \text{fftshift}(X, \text{dim})$

## 说明

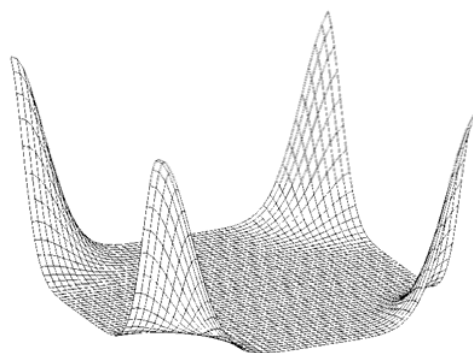
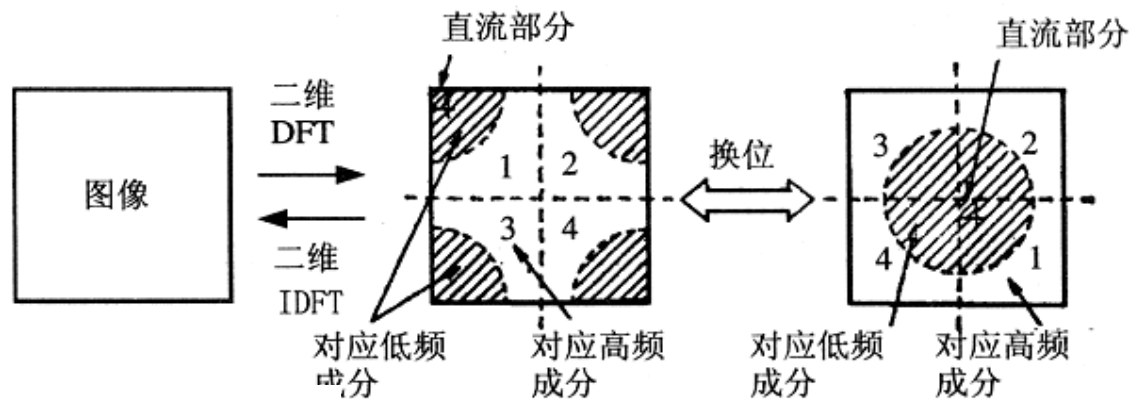
$Y = \text{fftshift}(X)$  通过将零频分量移动到数组中心，重新排列傅里叶变换  $X$ 。

如果  $X$  是向量，则  $\text{fftshift}$  会将  $X$  的左右两半部分进行交换。

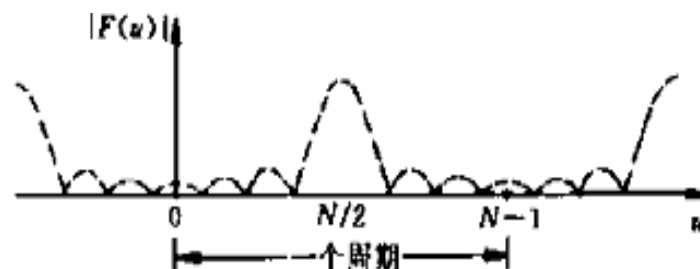
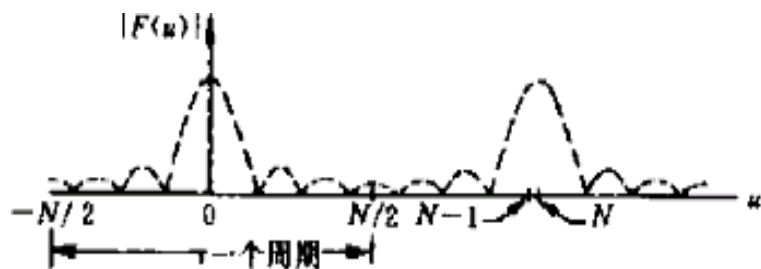
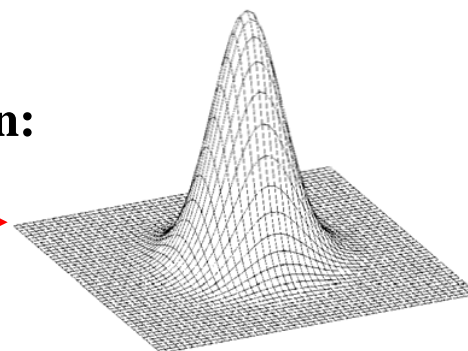
如果  $X$  是矩阵，则  $\text{fftshift}$  会将  $X$  的第一象限与第三象限交换，将第二象限与第四象限交换。

如果  $X$  是多维数组，则  $\text{fftshift}$  会沿每个维度交换  $X$  的半空间。

$Y = \text{fftshift}(X, \text{dim})$  沿  $X$  的维度  $\text{dim}$  执行运算。例如，如果  $X$  是矩阵，其行表示多个一维变换，则  $\text{fftshift}(X, 2)$  会将  $X$  的每一行的左右两半部分进行交换。



**MATLAB function:**  
*fftshift*

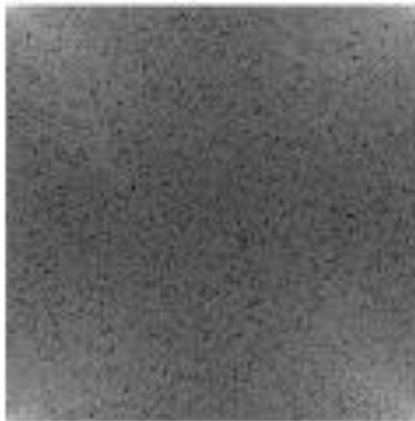


## 图像低通和高通滤波示例[代码](#)

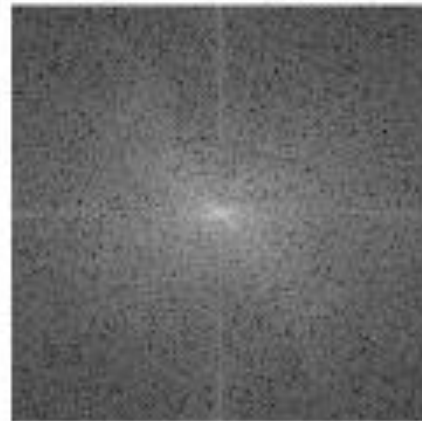
1. Src image



3. Image DFT



4. FFT Shift



2. Gray image



5. Low-pass filtering



6. High-pass filtering





## 平移矩阵中的信号

要处理多个一维信号，可将它们表示为一个矩阵的各行。然后使用维度参数计算傅里叶变换，并平移每一行的零频分量。

创建矩阵 A，其行表示两个一维信号，然后计算每个信号的傅里叶变换。绘制每个信号的功率。平移零频分量，然后绘制每个信号以零为中心的功率。

```
% sampling frequency  
fs = 100;
```

```
% time vector  
t = 0:(1/fs):(10-1/fs);
```

```
S1 = cos(2*pi*15*t);
```

```
S2 = cos(2*pi*30*t);
```

```
n = length(S1);
```

```
A = [S1; S2];
```

```
X = fft(A,[],2);
```

```
f = (0:n-1)*(fs/n); % frequency range
```

```
power = abs(X).^2/n; % power
```

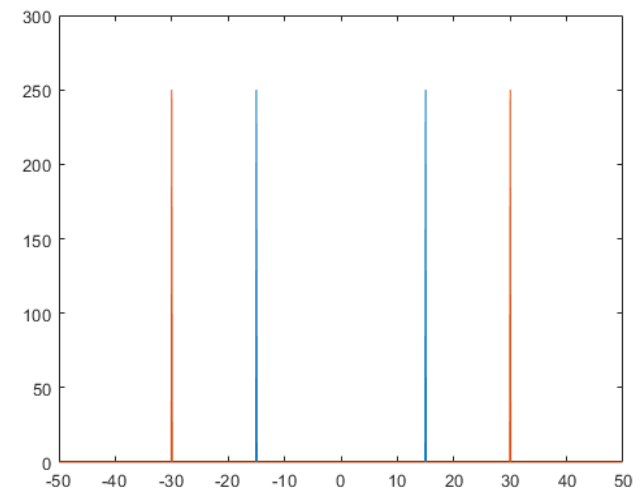
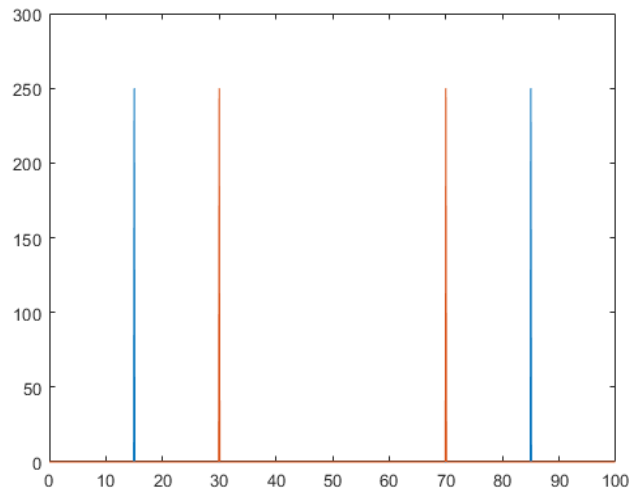
```
plot(f,power(1,:),f,power(2,:))
```

```
Y = fftshift(X,2);
```

```
fshift = (-n/2:n/2-1)*(fs/n); % zero-centered frequency range
```

```
powershift = abs(Y).^2/n; % zero-centered power
```

```
plot(fshift,powershift(1,:),fshift,powershift(2,:))
```



```

% 用sin(x+y)的图像，帮助理解二维图像傅立叶变换
img_size = 100; % 图片尺寸
x_step = 1;
y_step = 1;
image = zeros(img_size, img_size); % 先赋值0
for x = x_step:x_step:x_step*img_size % 按行生成数据
    for y=y_step:y_step:y_step*img_size % 按行对各列生成数据，相邻行向左移动一个数据
        image(x/x_step, y/y_step)=sin(4*pi*x/img_size + 4*pi*y/img_size);
    end % 相邻列向上移动一个数据
end
subplot(1,3,1); imshow(image); title('原图')

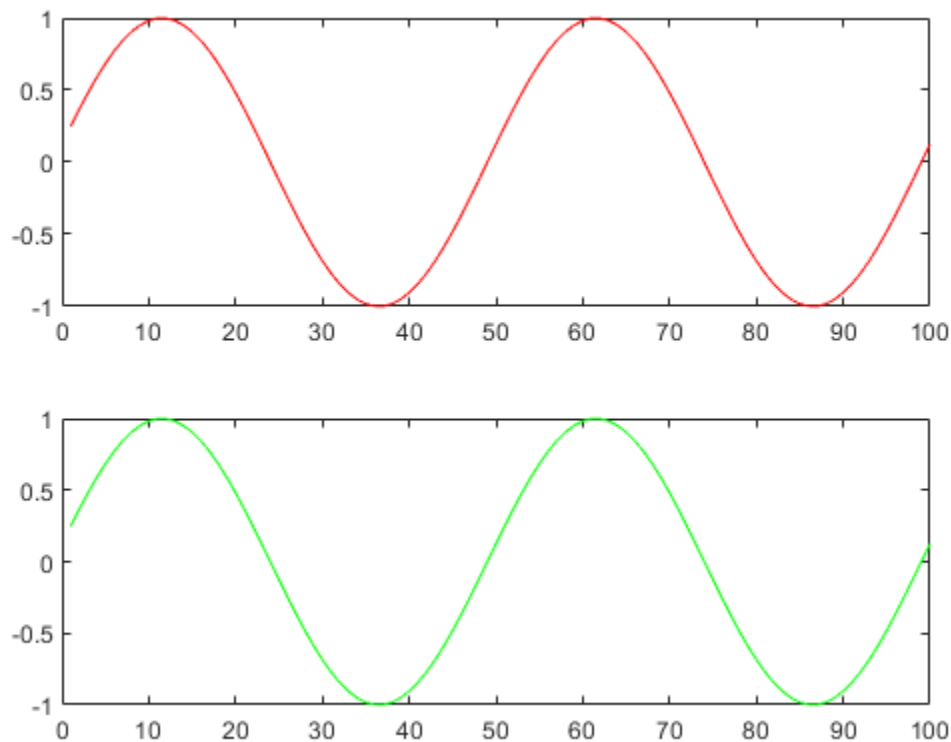
subplot(1,3,2); image = im2double(image);
F_unshift = fft2(image);
F_unshift_abs = abs(F_unshift);
T = log(F_unshift_abs+1);
imshow(T); title('傅立叶变换后，未频移')

subplot(1,3,3); F = fftshift(F_unshift);
F_abs = abs(F); T = log(F_abs+1);
imshow(T); title('傅立叶变换后，频移后')

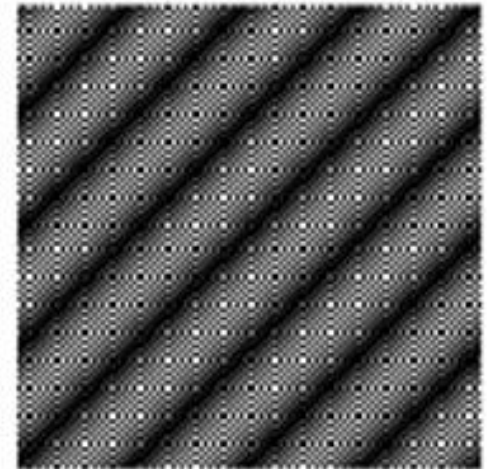
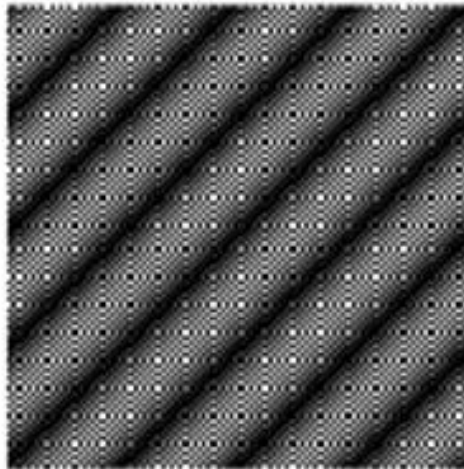
```



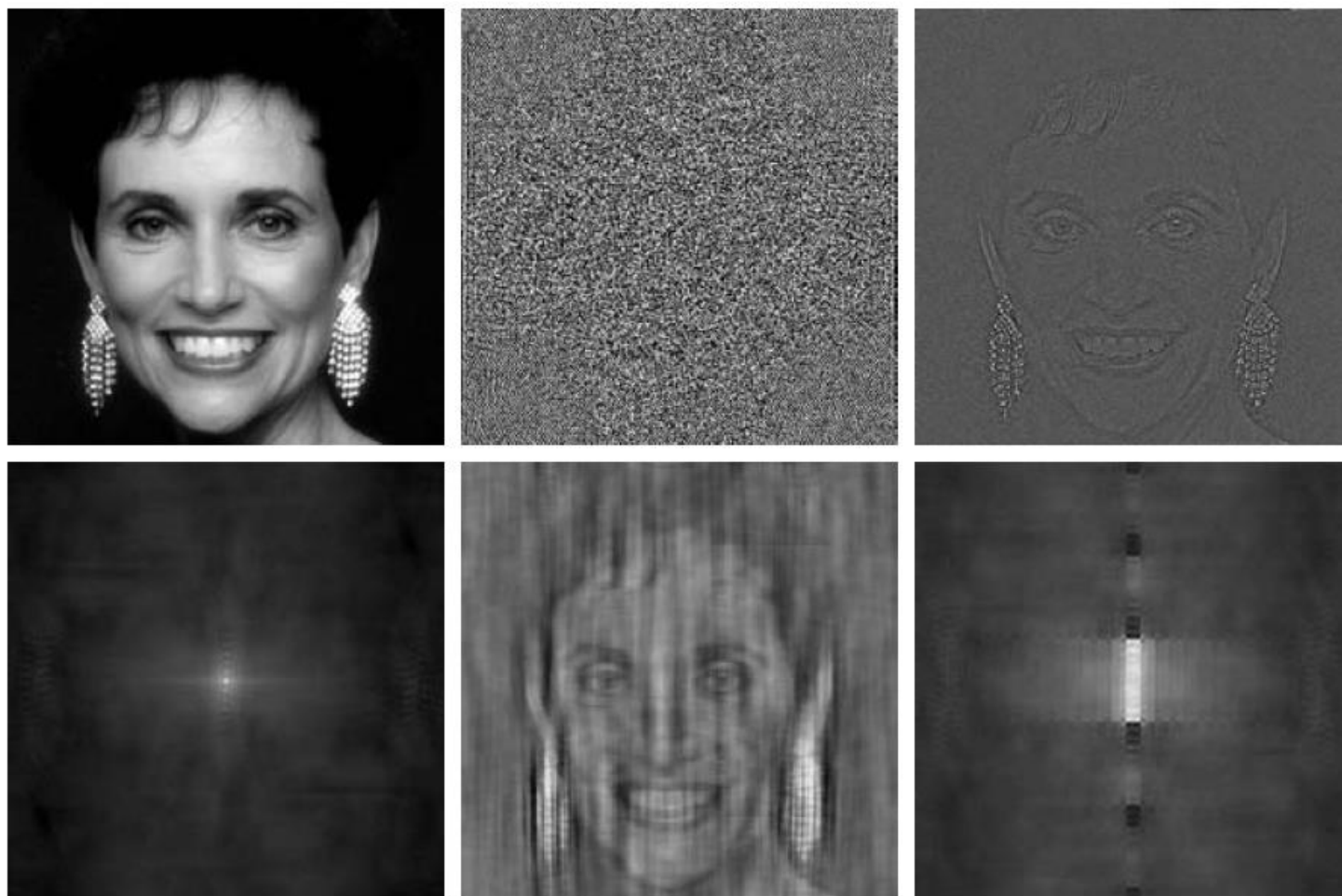
```
figure  
subplot(211);plot((1:size(image,1)),image(:,1),'r') % 获取行数, 绘制第1列 (各行) 数据  
subplot(212);plot((1:size(image,2)),image(1,:),'g') % 获取列数, 绘制第1行 (各列) 数据
```



```
% to get image from ifft2  
T1 = ifft2(F_unshift);  
F1 = ifft2(ifftshift(F_unshift));  
figure  
subplot(131);imshow(T1);  
subplot(132);imshow(F1);  
subplot(133);imshow(real(F1));
```

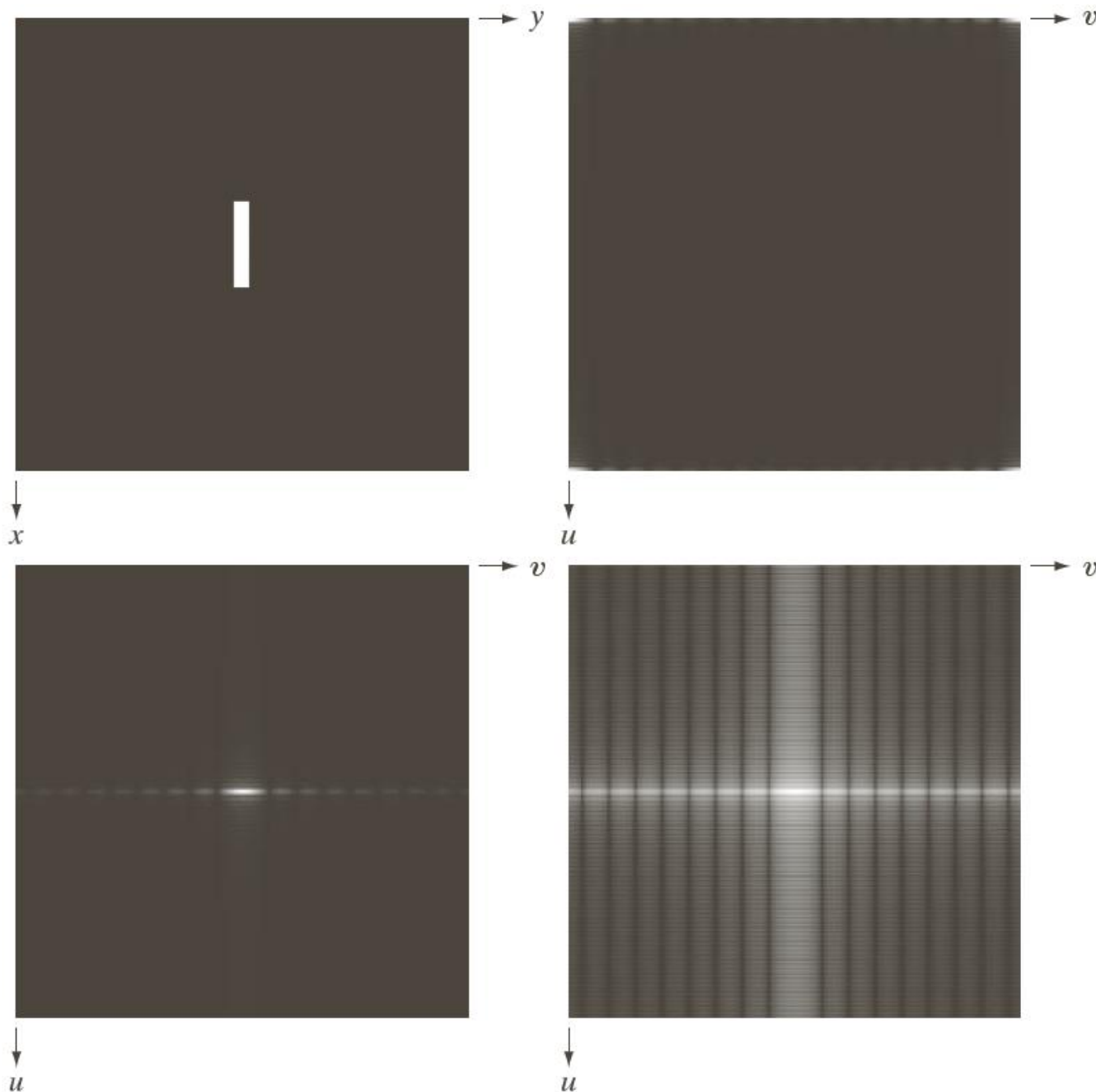


# 相位 的 意义



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.



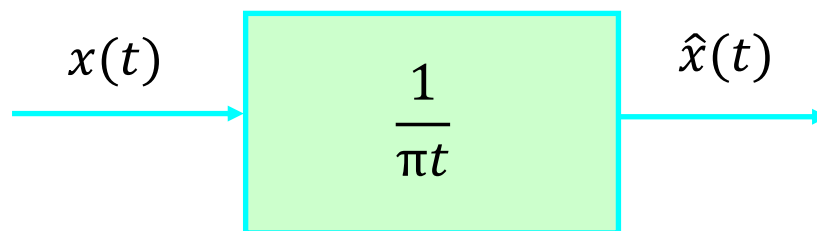
a	b
c	d

**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.

## 3.9 Hilbert 变换

$$\left. \begin{aligned} \hat{x}(t) &= \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \\ \hat{x}(t) &= \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(t - \tau)}{\tau} d\tau \end{aligned} \right\} \Rightarrow \hat{x}(t) = x(t) * \frac{1}{\pi t}$$



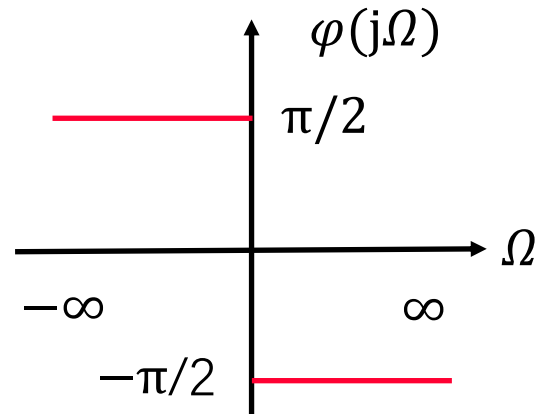
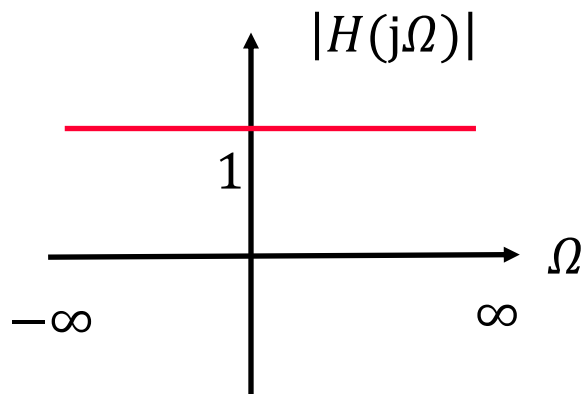
信号处理中重要的理论工具

$$h(t) = \frac{1}{\pi t} \Leftrightarrow H(j\Omega) = -j \operatorname{sgn}(\Omega) = \begin{cases} -j & \Omega > 0 \\ j & \Omega < 0 \end{cases}$$

$$H(j\Omega) = |H(j\Omega)|e^{j\varphi(\Omega)}$$

$$\therefore |H(j\Omega)| = 1$$

$$\varphi(\Omega) = \begin{cases} -\pi/2 & \Omega > 0 \\ \pi/2 & \Omega < 0 \end{cases}$$





令：  $z(t) = x(t) + j\hat{x}(t)$

  $x(t)$ 的解析 (Analytic) 信号

$$Z(j\Omega) = X(j\Omega) + j\hat{X}(j\Omega)$$

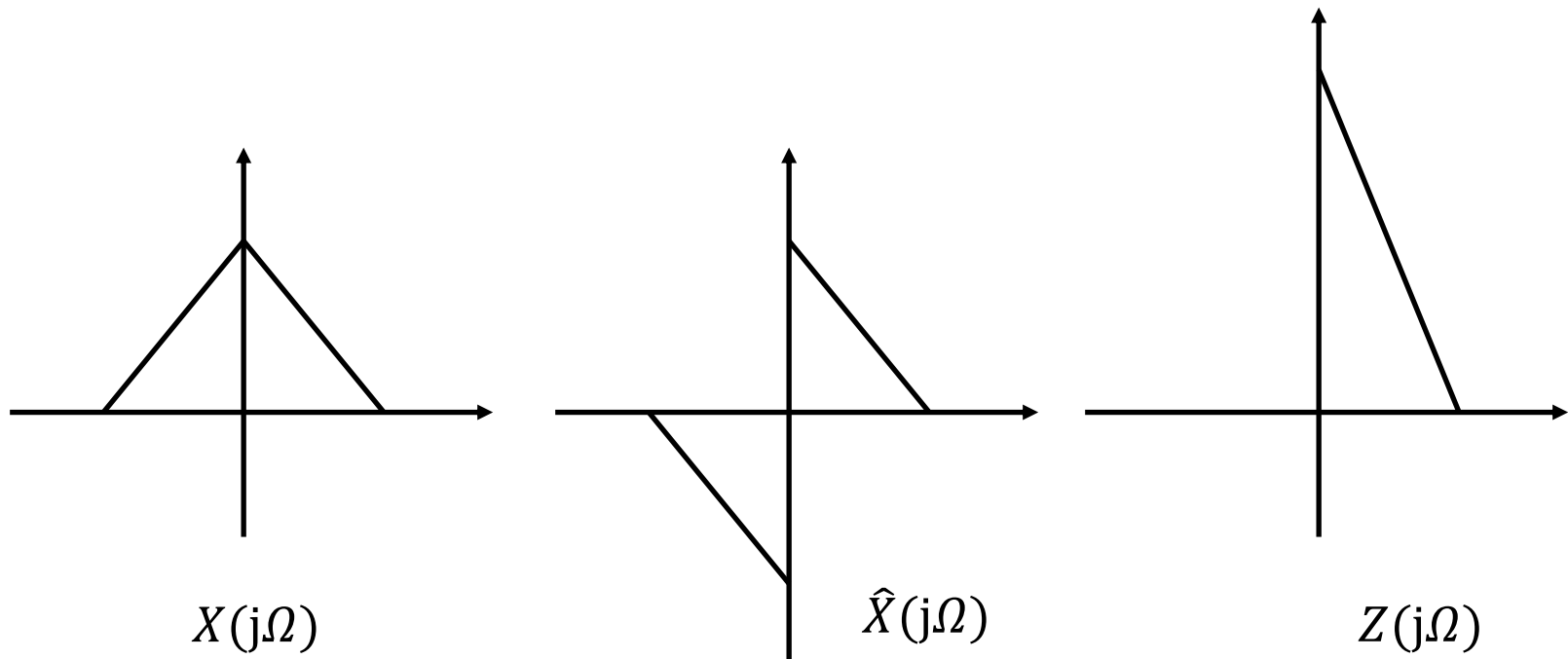
$$= X(j\Omega) + jX(j\Omega)\underline{H(j\Omega)}$$

$$H(j\Omega) = -j \operatorname{sgn}(\Omega) = \begin{cases} -j & \Omega > 0 \\ j & \Omega < 0 \end{cases}$$

$$Z(j\Omega) = \begin{cases} 2X(j\Omega) & \Omega > 0 \\ 0 & \Omega < 0 \end{cases}$$

Hilbert 变换可用于形成解析信号。

解析信号在通信领域中很有用，尤其是在带通信号处理中。



解析信号 $z(t)$ 的频谱只有正频率成分！显然，若对 $z(t)$ 抽样，抽样频率可降低一倍。另外，做时－频分析时，可减轻正、负频率处的交叉干扰。

## Hilbert 反变换

$$\hat{X}(j\Omega) = X(j\Omega)H(j\Omega) = X(j\Omega)[-j \operatorname{sgn}(\Omega)] = jX(j\Omega) \operatorname{sgn}(-\Omega)$$

$$X(j\Omega) = -j \operatorname{sgn}(-\Omega) \hat{X}(j\Omega)$$

$$x(t) = -\frac{1}{\pi t} * \hat{x}(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{\hat{x}(\tau)}{t - \tau} d\tau$$

例： 若

$$x(t) = A \cos(2\pi f_0 t)$$

可求出

$$\hat{x}(t) = A \sin(2\pi f_0 t)$$

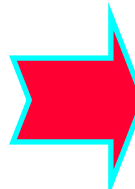
$$z(t) = x(t) + j\hat{x}(t) = Ae^{j2\pi f_0 t}$$

正、余弦函数构成一对 Hilbert 变换
-----------------------

## 离散信号的 Hilbert 变换

$$H(e^{j\omega}) = \begin{cases} -j & 0 < \omega < \pi \\ j & -\pi < \omega < 0 \end{cases}$$

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega$$

$$h(n) = \frac{1 - (-1)^n}{n\pi} = \begin{cases} 0 & n \text{ 为偶数} \\ \frac{2}{n\pi} & n \text{ 为奇数} \end{cases}$$


Hilbert 变换器的  
单位抽样响应

$$\hat{x}(n) = x(n) * h(n) = \frac{2}{\pi} \sum_{m=-\infty}^{\infty} \frac{x(n - 2m - 1)}{(2m + 1)}$$

解析信号:  $z(n) = x(n) + j\hat{x}(n)$

## 如何有效的计算Hilbert变换？

Step 1. 对 $x(n)$ 做 DFT, 得:  $X(k)$ ,  $k = 0, \dots, N - 1$

$$\text{Step 2. 令 } Z(k) = \begin{cases} X(k) & k = 0 \\ 2X(k) & k = 1, 2, \dots, \frac{N}{2} - 1 \\ 0 & k = \frac{N}{2}, \dots, N - 1 \end{cases}$$

Step 3. 对 $Z(k)$ 做逆 DFT, 得 $z(n)$

Step 4. 由 $\hat{x}(n) = \text{IDFT}[-j(Z(k) - X(k))]$

$$\text{得 } \hat{x}(n) = -j[z(n) - x(n)]$$


# Hilbert 变换的性质

$$\hat{X}(j\Omega) = X(j\Omega)[-j \operatorname{sgn}(\Omega)]$$

1. 信号通过Hilbert变换器后幅度谱不发生变化
2. 信号和其Hilbert变换是正交的

$$\begin{aligned}\int_{-\infty}^{\infty} x(t)\hat{x}(t)dt &= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega)[\hat{X}(j\Omega)]^* d\Omega \\ &= \frac{j}{2\pi} \int_{-\infty}^0 |X(j\Omega)|^2 d\Omega - \frac{j}{2\pi} \int_0^{\infty} |X(j\Omega)|^2 d\Omega = 0\end{aligned}$$

$$\sum_{n=-\infty}^{\infty} x(n)\hat{x}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})[\hat{X}(e^{j\omega})]^* d\omega = 0$$

但我们并不把Hilbert变换看作是正交变换

### 3. 卷积性质

$$\text{let } \begin{cases} x(t) \Leftrightarrow \hat{x}(t) \\ x_1(t) \Leftrightarrow \hat{x}_1(t) \\ x_2(t) \Leftrightarrow \hat{x}_2(t) \end{cases}$$

$$\begin{cases} \text{if:} & x(t) = x_1(t) * x_2(t) \\ \text{then:} & \hat{x}(t) = \hat{x}_1(t) * x_2(t) \\ & = x_1(t) * \hat{x}_2(t) \end{cases}$$

## 实因果信号傅立叶变换的一些内部关系

$$\text{let: } x(n) = x^*(n), \quad x(n) \equiv 0 \quad \text{for } n < 0$$

---



实因果信号

$$X(e^{j\omega}) = X_R(e^{j\omega}) + jX_I(e^{j\omega})$$

直角坐标

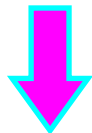
Hilbert  
变换关系

$$\left\{ \begin{array}{l} X_I(e^{j\omega}) = -\frac{1}{2\pi} \int_{-\pi}^{\pi} X_R(e^{j\theta}) \cot\left(\frac{\omega - \theta}{2}\right) d\theta \\ X_R(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X_I(e^{j\theta}) \cot\left(\frac{\omega - \theta}{2}\right) d\theta + x(0) \end{array} \right.$$



$$X(e^{j\omega}) = |X(e^{j\omega})|e^{j\varphi(\omega)} = |X(e^{j\omega})|e^{j \arg[X(e^{j\omega})]}$$

取对数



极坐标

$$\check{X}(e^{j\omega}) = \ln|X(e^{j\omega})| + j \arg[X(e^{j\omega})]$$

$$\begin{cases} \arg[X(e^{j\omega})] = -\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln|X(e^{j\theta})| \cot\left(\frac{\omega - \theta}{2}\right) d\theta \\ \ln|X(e^{j\omega})| = \frac{1}{2\pi} \int_{-\pi}^{\pi} \arg[X(e^{j\theta})] \cot\left(\frac{\omega - \theta}{2}\right) d\theta + \check{x}(0) \end{cases}$$

Hilbert 变换关系

$$\check{x}(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln|X(e^{j\omega})| d\omega$$

$$\check{X}(e^{j\omega}) \Leftrightarrow \check{x}(n)$$

$x(n)$ 的  
复倒谱

$\left\{ \begin{array}{l} \text{Spectrum} \\ \text{Cepstrum} \end{array} \right.$

函数 `hilbert` 计算实数序列  $x$  的 Hilbert 变换，并返回相同长度的复数结果，即  $y = \text{hilbert}(x)$ ， $y$  的实部是原始实数数据，虚部是实际 Hilbert 变换。 $y$  有时被称为解析信号。

离散时间解析信号的关键属性是其 Z 变换在单位圆的下半部分为 0，解析信号的许多应用都与此有关。例如，用解析信号避免带通采样操作的混叠效应。解析信号的幅值是原始信号的复包络。解析信号可用于计算时序的瞬时属性，即时序在任一时间点的属性，该过程要求信号是单分量的。

%要绘制一部分数据及其  
Hilbert 变换

```
t = 0:1/1024:1;
```

```
x = sin(2*pi*60*t);
```

```
y = hilbert(x);
```

```
plot(t(1:50),real(y(1:50)))
```

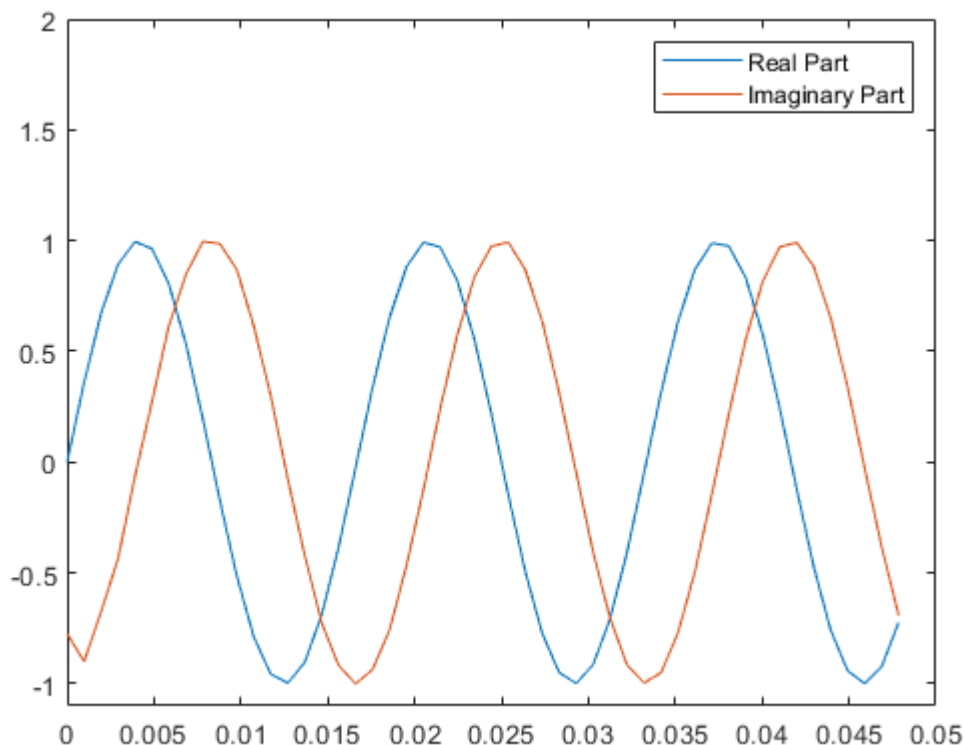
```
hold on
```

```
plot(t(1:50),imag(y(1:50)))
```

```
hold off
```

```
axis([0 0.05 -1.1 2])
```

```
legend('Real Part','Imaginary  
Part')
```



```
t = 0:1e-4:0.1;  
x = (1+cos(2*pi*50*t)).*cos(2*pi*1000*t);  
plot(t,x); xlim([0 0.04])  
  
y = hilbert(x);  
env = abs(y);  
plot_param = {'Color', [0.6 0.1 0.2], 'Linewidth', 2};
```

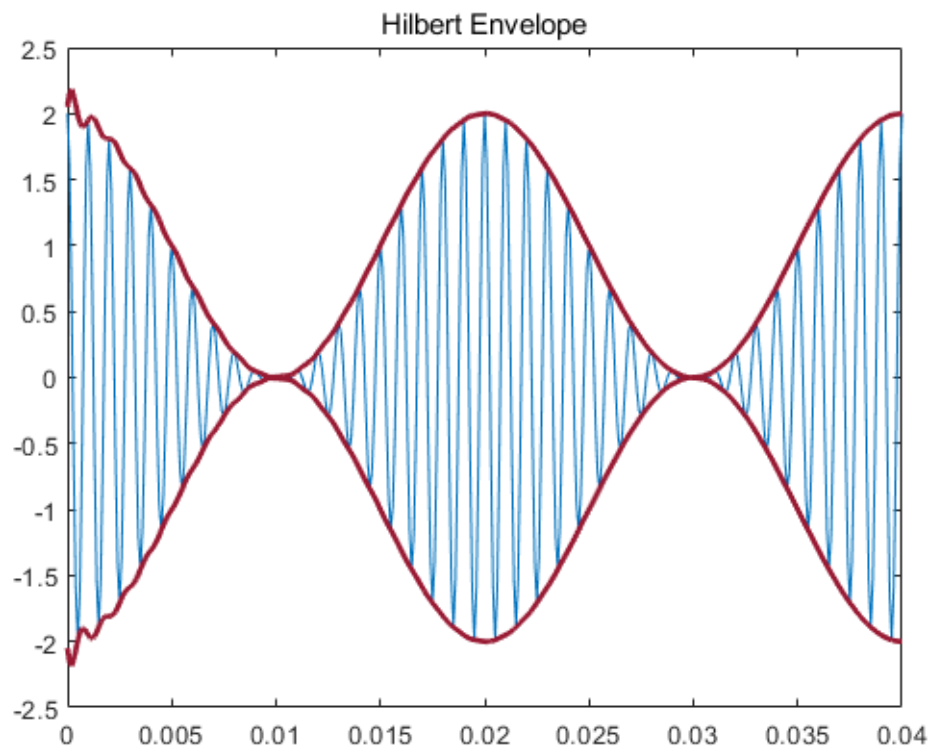
```
plot(t,x)  
  
hold on  
plot(t,[-1;1]*env,plot_param{:})  
hold off  
xlim([0 0.04])  
title('Hilbert Envelope')
```

## 信号包络提取

创建双边带幅值调制信号，载波频率为1kHz，调制频率为50Hz，调制深度为100%，采样率为10kHz。

使用hilbert提取包络，包络是由hilbert计算的解析信号的幅值，绘制包络和原始信号。将plot函数的名称-值对组参数存储在元胞数组中供以后使用。

分析信号的幅值捕获信号的缓慢变化特性，相位包含高频信息。



# 与本章有关的 MATLAB 文件

fftfilt.m 用叠接相加法实现卷积。格式是

$$y=\text{fftfilt}(h,x) \text{ 或 } y=\text{fftfilt}(h, x,N)$$

记 $x(n)$ 的长度为 $N_x$ ,  $h(n)$ 的长度为 $M$ 。

若采用第一个调用方式, 程序自动确定对 $x(n)$ 分段的长度 $L$ 及做FFT的长度 $N$ , 显然,  $N$ 是最接近 $(L + M)$ 的2的整次幂。分的段数为 $N_x/L$ 。

采用第二个调用方式, 使用者可自己指定做FFT的长度。建议使用第一个调用方式。

hilbert.m

文件用来计算信号的解析信号。调用的格式是：

$$y = \text{hilbert}(x)$$

$y$ 的实部就是 $x(n)$ ，虚部是 $x(n)$ 的Hilbert变换 $\hat{x}(n)$ 。

即：  $y(n) = x(n) + j\hat{x}(n)$