《数字图像处理》
第6讲 频域图像增强

冯建江

清华大学 自动化系

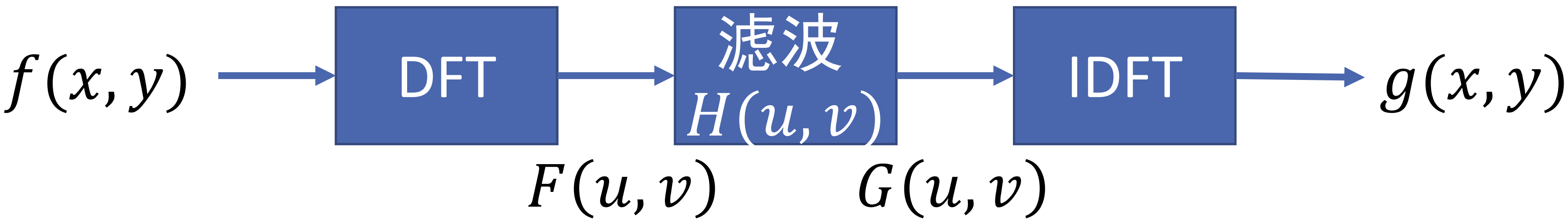
2017.11.02

内 容

- 频域滤波基础
- 图像平滑
- 图像锐化
- 选择性滤波

频域滤波

- 以图像增强为应用，介绍频域滤波。频域滤波还可用于图像恢复、压缩等。
- 给定输入图像$f(x,y)$，大小为$M \times N$
- 频域滤波表达式：$g(x,y) = \mathfrak{F}^{-1}[H(u,v)F(u,v)]$
- 其中，$F(u,v)$是$f(x,y)$的DFT，$H(u,v)$是滤波器，$g(x,y)$是输出图像，均为$M \times N$数组



二维离散傅里叶变换

$f(x,y)$为$M \times N$数字图像。二维离散傅里叶变换（DFT）和离散傅里叶反变换（IDFT）分别为

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

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

$$u = 0,1,2,\ldots,M-1$$

$$v = 0,1,2,\ldots,N-1$$

$$x = 0,1,2,\ldots,M-1$$

$$y = 0,1,2,\ldots,N-1$$

幅度谱和相位谱

$$F(u,v) = R(u,v) + jI(u,v)$$
$$F(u,v) = |F(u,v)|e^{j\phi(u,v)}$$

- 幅度谱（magnitude spectrum），也叫傅里叶频谱（Fourier spectrum）

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

- 相位谱（phase spectrum），也叫相位角（phase angle）

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right], \qquad \phi(u,v) \in [0,2\pi]$$

- 功率谱（power spectrum）

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

一些重要的离散傅里叶变换对

8) Discrete unit impulse $\delta(x, y) \Leftrightarrow 1$

9) Rectangle $\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$

10) Sine $\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$

$$j\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$$

11) Cosine $\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$

$$\frac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$$

The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν for frequency variables. These results can be used for DFT work by sampling the continuous forms.

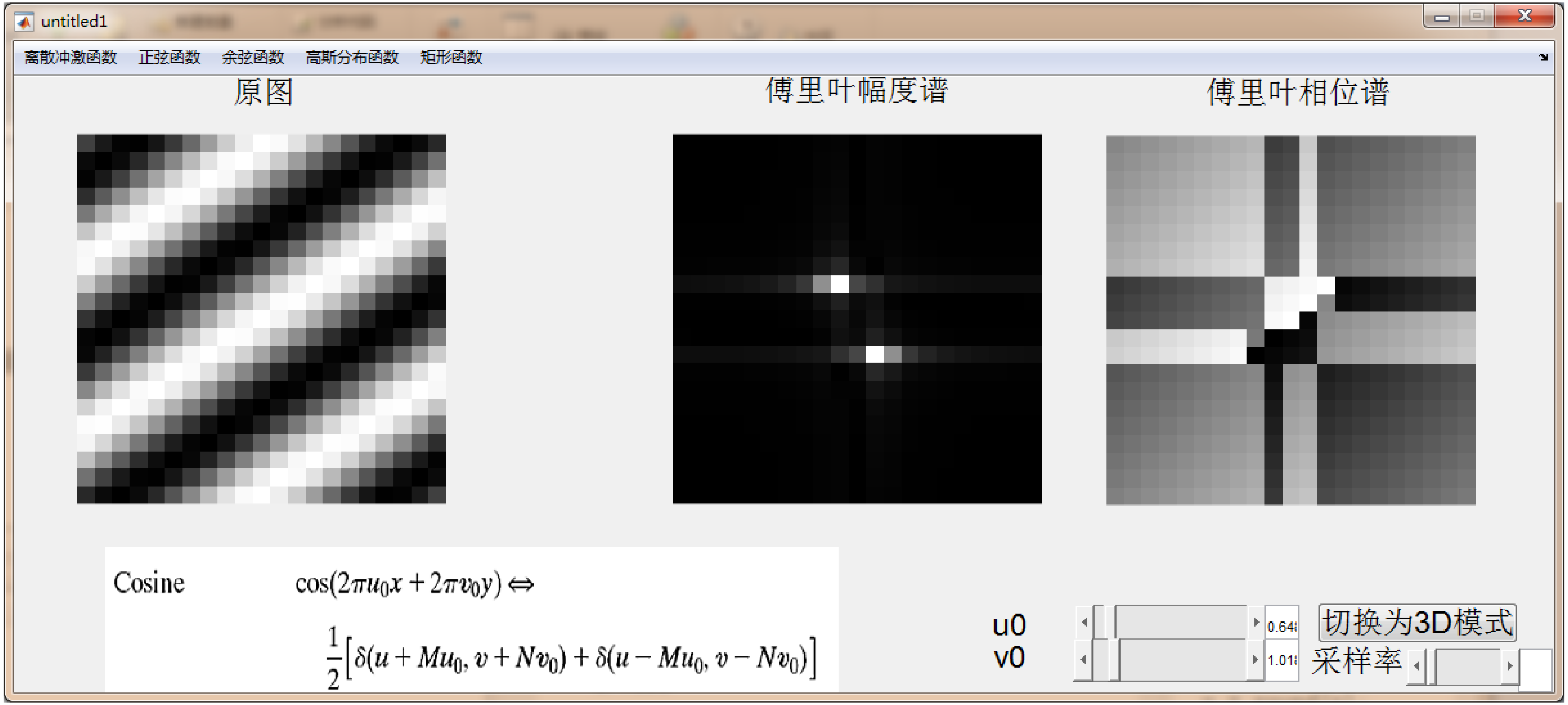
12) *Differentiation* (The expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$.)

$$\left(\frac{\partial}{\partial t}\right)^m \left(\frac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu, \nu)$$

$$\frac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \frac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$$

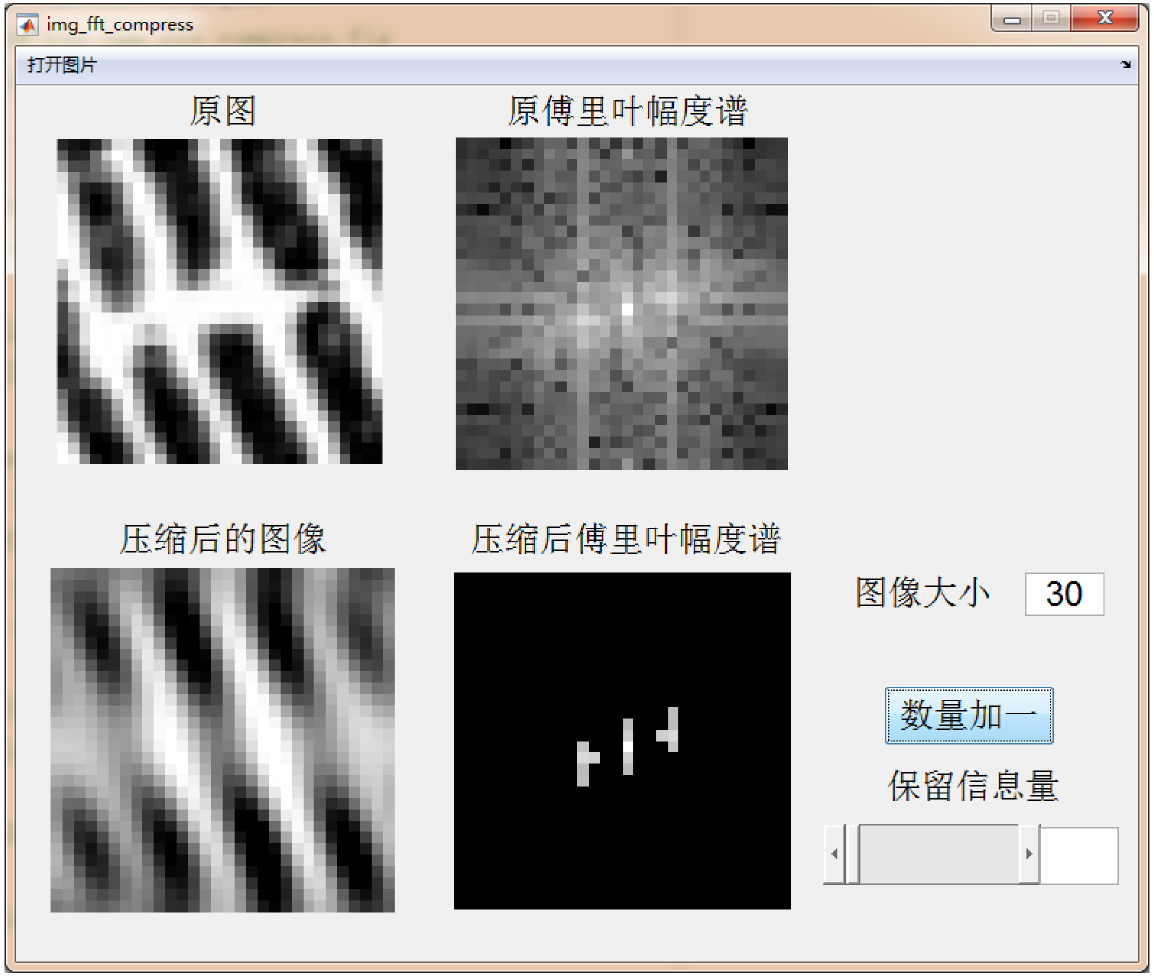
13) *Gaussian* $A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ (A is a constant)

Demo： 重要函数的DFT

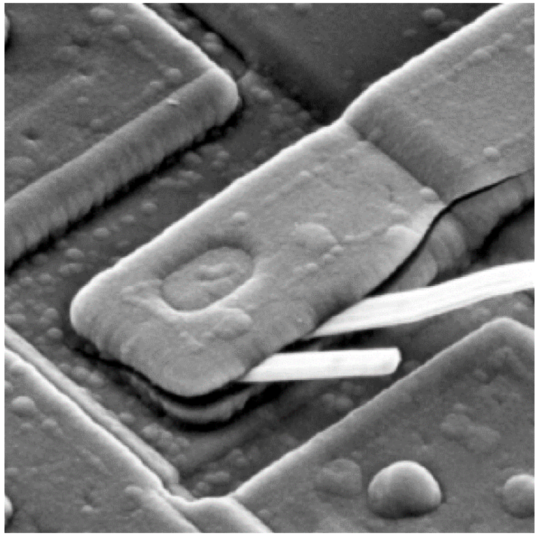


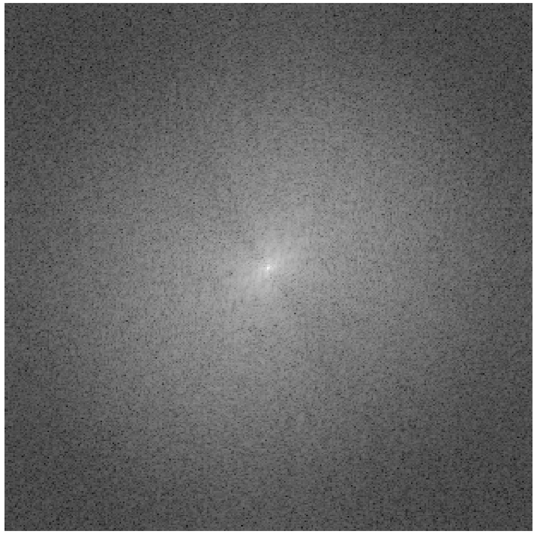
Demo：重建效果

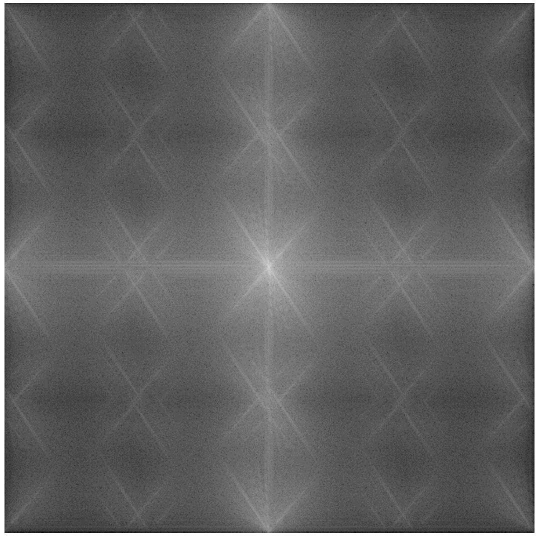
- 上节课的demo演示了傅里叶变换各系数的计算
- 下面的demo演示利用部分系数做傅里叶反变换（对图像进行重建），加深大家对于傅里叶变换的理解

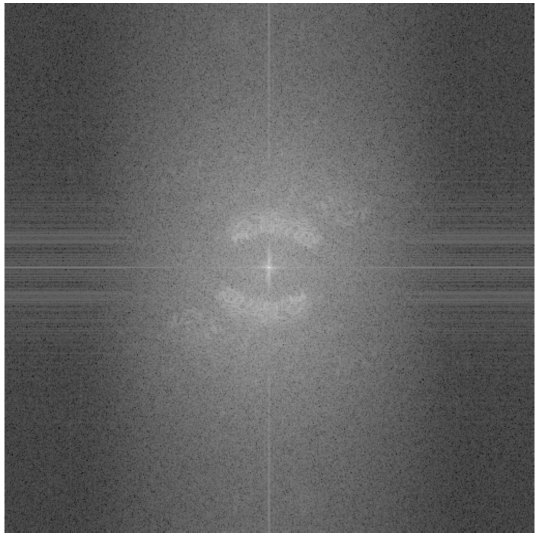


连线题：连接图像和对应的幅度谱

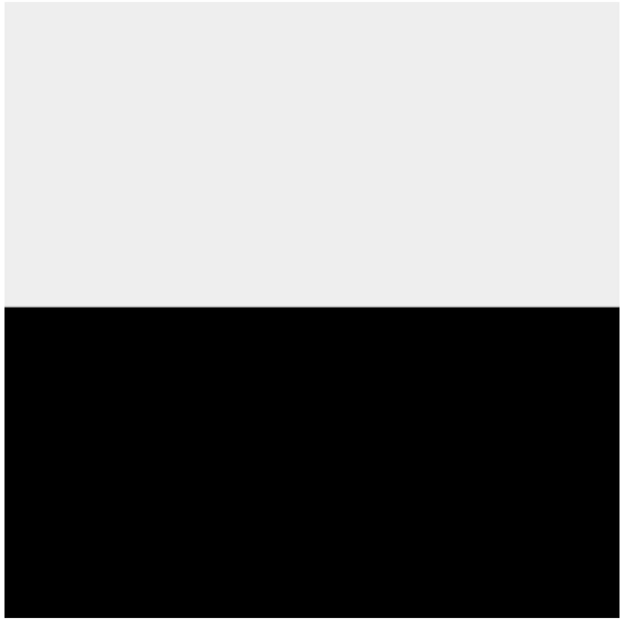




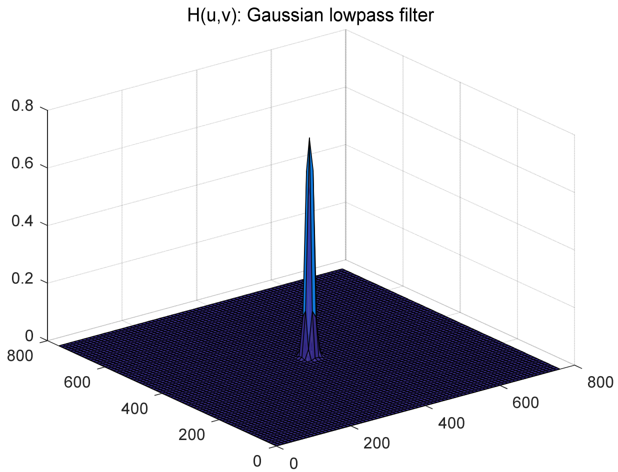




交叠误差（wraparound error）



输入图像



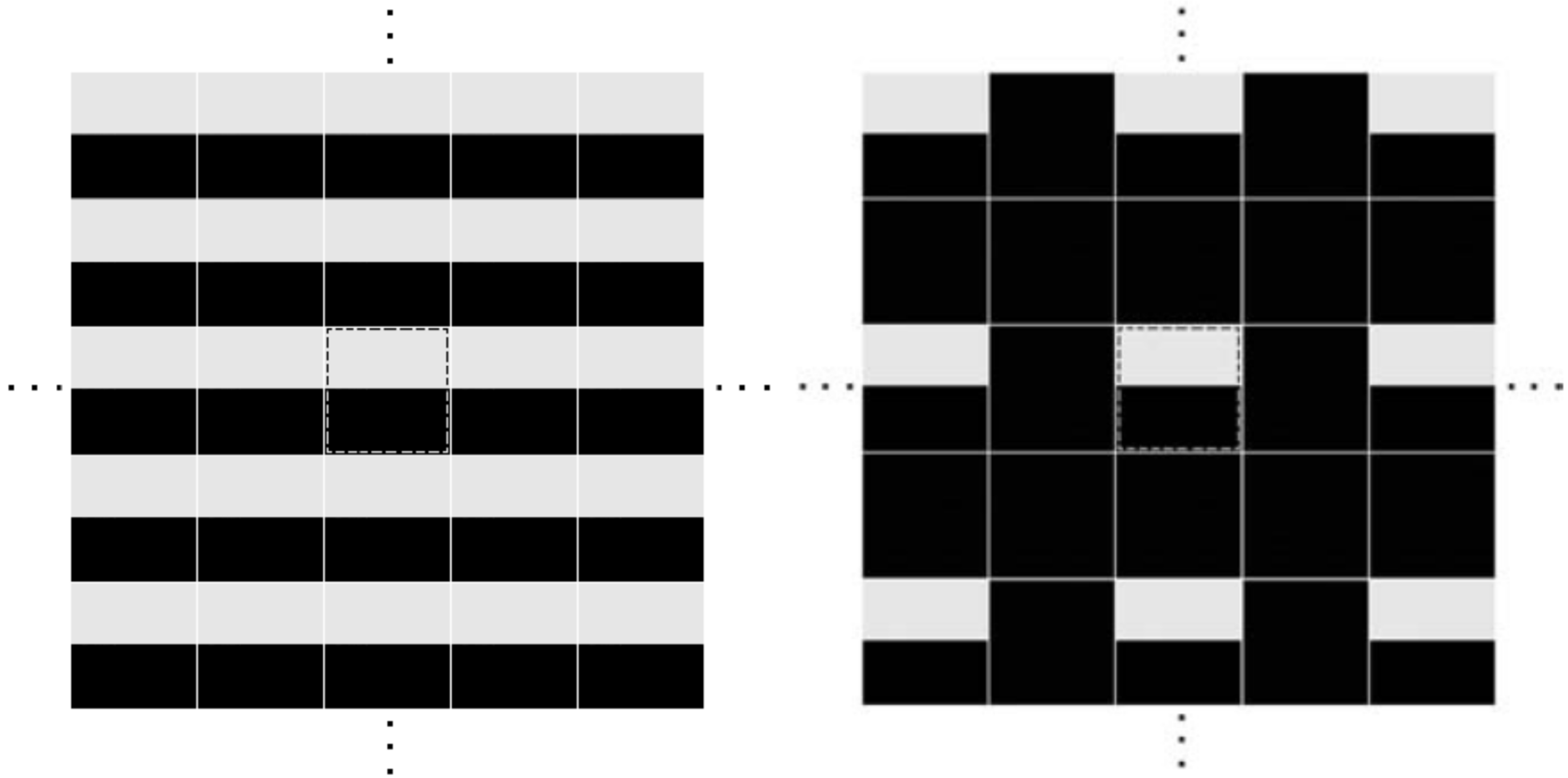
高斯低通滤波器

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

未补零图像的滤波结果

```matlab
%fig0432_zeroPadding
close all
f = imread('..\data\Fig0432(a)(square_original).tif');
f = im2double(f);
[M,N] = size(f);
P = max([M N]);
f = padarray(f,[P-M P-N],0,'post');

F = fftshift(fft2(f));

% Gaussian lowpass filter
[X, Y] = meshgrid(1:P);
D0 = P/100;
H = exp(-((X-P/2-1).^2+(Y-P/2-1).^2)/(2*D0*D0));

% filtering
G = F.*H;

figure(1),imshow(f),title('f(x,y): Input Image')
figure(2),imshow(log(1+abs(F)),[]),title('|F(u,v)|: Spectrum of f')
figure(3),surf(X(1:10:end,1:10:end),Y(1:10:end,1:10:end),H(1:10:end,1:10:end)),
title('H(u,v): Gaussian lowpass filter')

g = real(ifft2(ifftshift(G)));
figure(4),imshow(log(1+abs(G)),[]),title('G(u,v)=F(u,v)H(u,v)')
figure(5),imshow(g),title('g(x,y): filtering result without zero padding')
```

交叠误差的原因

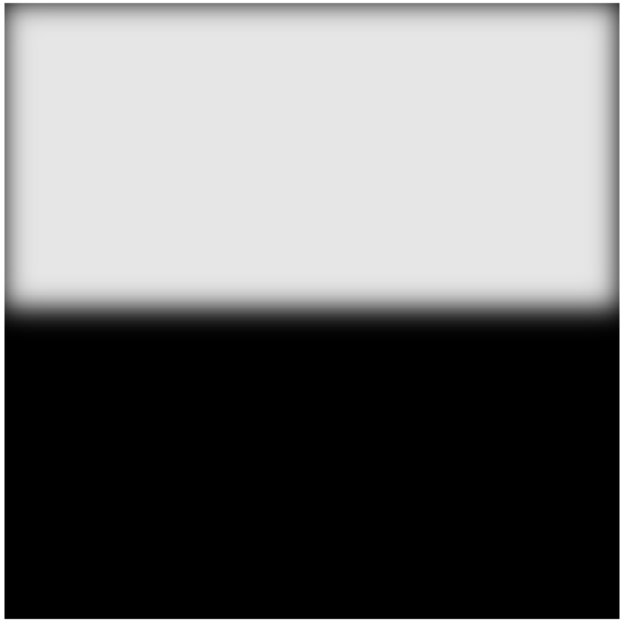


a b

FIGURE 4.33 2-D image periodicity inherent in using the DFT. (a) Periodicity without image padding. (b) Periodicity after padding with 0s (black). The dashed areas in the center correspond to the image in Fig. 4.32(a). (The thin white lines in both images are superimposed for clarity; they are not part of the data.)

补零

补零图像



补零图像的
滤波结果（已去零）

未补零图像的
滤波结果

```matlab
%fig0432_zeroPadding
close all
f = imread('..\data\Fig0432(a)(square_original).tif');
f = im2double(f);
[M,N] = size(f);
P = 2*max([M N]);
f = padarray(f,[P-M P-N],0,'post');

F = fftshift(fft2(f));

% Gaussian lowpass filter
[X, Y] = meshgrid(1:P);
D0 = P/100;
H = exp(-((X-P/2-1).^2+(Y-P/2-1).^2)/(2*D0*D0));

% filtering
G = F.*H;

figure(1),imshow(f),title('f(x,y): Input Image')
figure(2),imshow(log(1+abs(F)),[]),title('|F(u,v)|: Spectrum of f')
figure(3),surf(X(1:10:end,1:10:end),Y(1:10:end,1:10:end),H(1:10:end,1:10:end)),
title('H(u,v): Gaussian lowpass filter')

g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);
figure(4),imshow(log(1+abs(G)),[]),title('G(u,v)=F(u,v)H(u,v)')
figure(5),imshow(g),title('g(x,y): filtering result with zero padding')
```

频域滤波的步骤

① 计算P和Q

② 扩充图像

③ 计算傅里叶变换，频谱居中

④ 用滤波器进行滤波

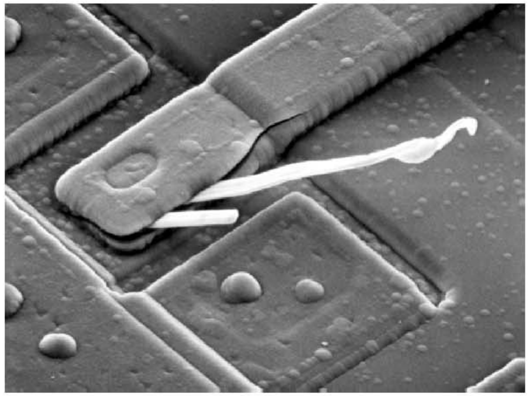
⑤ 频谱移位，傅里叶反变换

⑥ 取出左上角的子图像

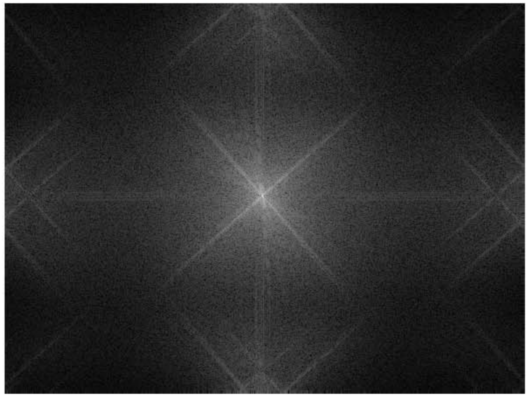
例子：去直流成分

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

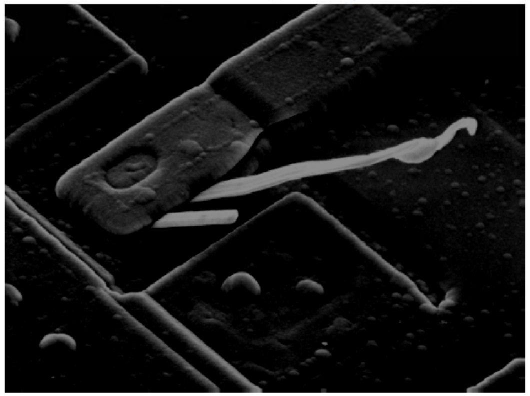
I: Original Image



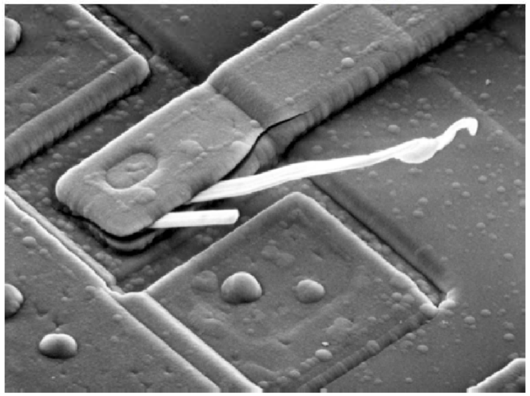
|F|: Spectrum of I



I2: Result of filtering I by removing dc

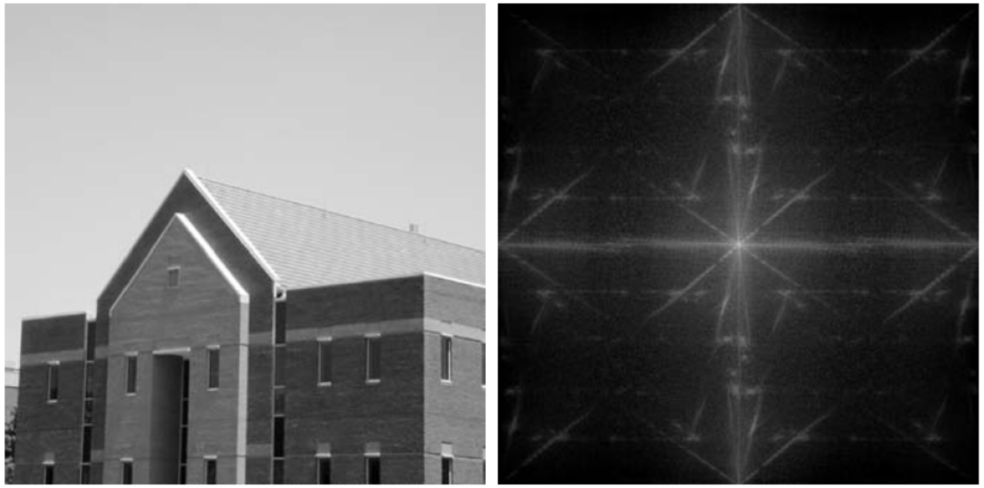


I3=I2+mean2(I)



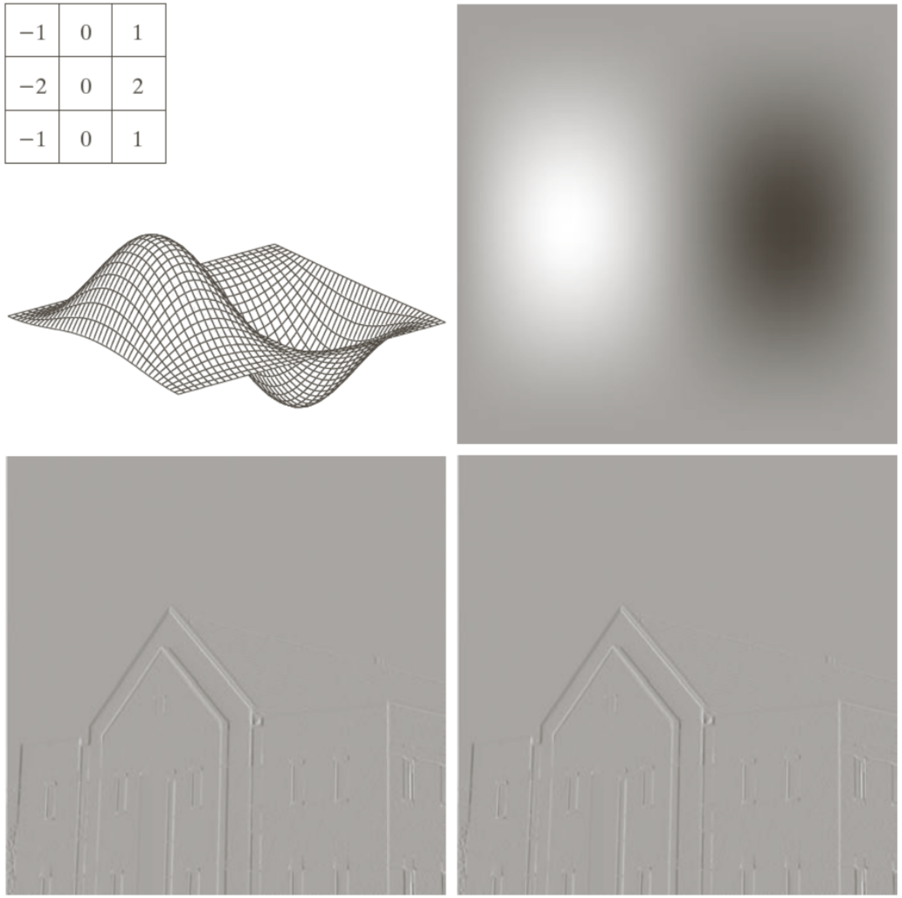
```matlab
%fig0430_RemoveDC
I = imread('..\data\Fig0429(a)(blown_ic).tif');
I = im2double(I);
[M,N] = size(I);
F = fftshift(fft2(I));

% check that F(M/2+1,N/2+1) is the center
abs(F(M/2-1:M/2+3,N/2-1:N/2+3))

figure(1),subplot(2,2,1),imshow(I),title('I: Original Image')
subplot(2,2,2),imshow(log(1+abs(F)),[]),title('|F|: Spectrum of I')

F2 = F;
F2(M/2+1,N/2+1) = 0;% remove DC
I2 = real(ifft2(ifftshift(F2)));
mean2(I)
mean2(I2)
subplot(2,2,3),imshow(I2),title('I2: Result of filtering I by removing dc')
I3 = I2+mean2(I);
subplot(2,2,4),imshow(I3),title('I3=I2+mean2(I)')
```

空域滤波器的频域实现



a b

FIGURE 4.38
(a) Image of a building, and (b) its spectrum.



−1	0	1
−2	0	2
−1	0	1

a b
c d

FIGURE 4.39
(a) A spatial mask and perspective plot of its corresponding frequency domain filter. (b) Filter shown as an image. (c) Result of filtering Fig. 4.38(a) in the frequency domain with the filter in (b). (d) Result of filtering the same image with the spatial filter in (a). The results are identical.

```
f = imread('..\data\Fig0438(a)(bld_600by600).tif');
f = im2double(f);
[M, N] = size(f);

h = [-1 0 1; -2 0 2; -1 0 1];
g_spatial = imfilter(f, h);
figure(1),imshow(f,[]),ax=gca;
figure(2),imshow(g_spatial,[]),ax(end+1)=gca;

% pad
P = max([M N])+2;
fp = padarray(f,[P-M P-N],0,'post');
hp = zeros(P,P);
hp(P/2:P/2+2,P/2:P/2+2) = rot90(rot90(h));% convulution kernal

% DFT
[DX, DY] = meshgrid(0:P-1,0:P-1);
shift_matrix = (-ones(P,P)).^(DX+DY);
F = fft2(fp.*shift_matrix);
H = fft2(hp.*shift_matrix).*shift_matrix;
H = 1i*imag(H);
figure(3),surf(imag(H(1:20:end,1:20:end))),axis ij;
figure(4),imshow(imag(H),[]);
G = F.*H;
gp = real(ifft2(G)).*shift_matrix;
g = gp(1:M,1:N);
figure(5),imshow(g,[]),ax(end+1)=gca;
linkaxes(ax);

d = abs(g_spatial - g);
max(d(:))
```

频域滤波与空域滤波的关系

- 空域的线性滤波器可以在频域实现（卷积定理）
- 为什么要讲频域滤波？
- 因为为了深入理解图像，需要在频域进行分析，在频域设计滤波器
- 但频域滤波的不足是运算量较大，且需要浮点运算
- 实际中的常见做法，在频域设计滤波器，用小尺寸的空域滤波器近似，提高速度。

内 容

- 频域滤波基础
- 图像平滑
- 图像锐化
- 选择性滤波

图像平滑

- 图像平滑：去除细节、噪声
- 第三节讲过空域的图像平滑方法，如均值滤波、加权均值滤波
- 细节、噪声往往对应高频，平滑可用低通滤波实现
- 下面介绍三种低通滤波器

均值滤波器
（averaging filter）

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

加权均值滤波器
（weighted averaging filter）

$$\frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

理想低通滤波器（ideal lowpass filter）

转移函数（transfer function）为

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

$$D(u,v)=\sqrt{(u-P/2)^2+(v-Q/2)^2}$$

参数D_0为截止频率（cutoff frequency）

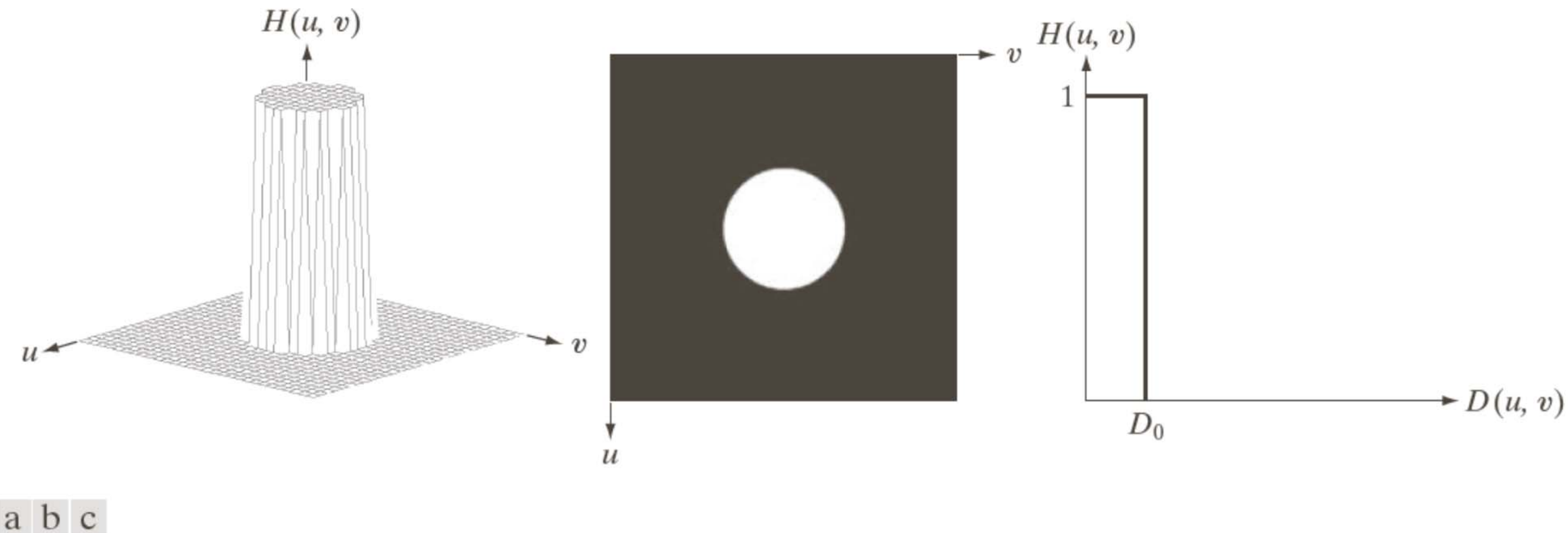


FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

理想低通滤波器

- 为什么是理想的？
- 无法用模拟电子电路实现。
- 无法用空域滤波器完美实现。
- 因为其脉冲响应是sinc函数，空间无限大。
- 当然，用计算机可以在频域仿真。

截止频率与能量比

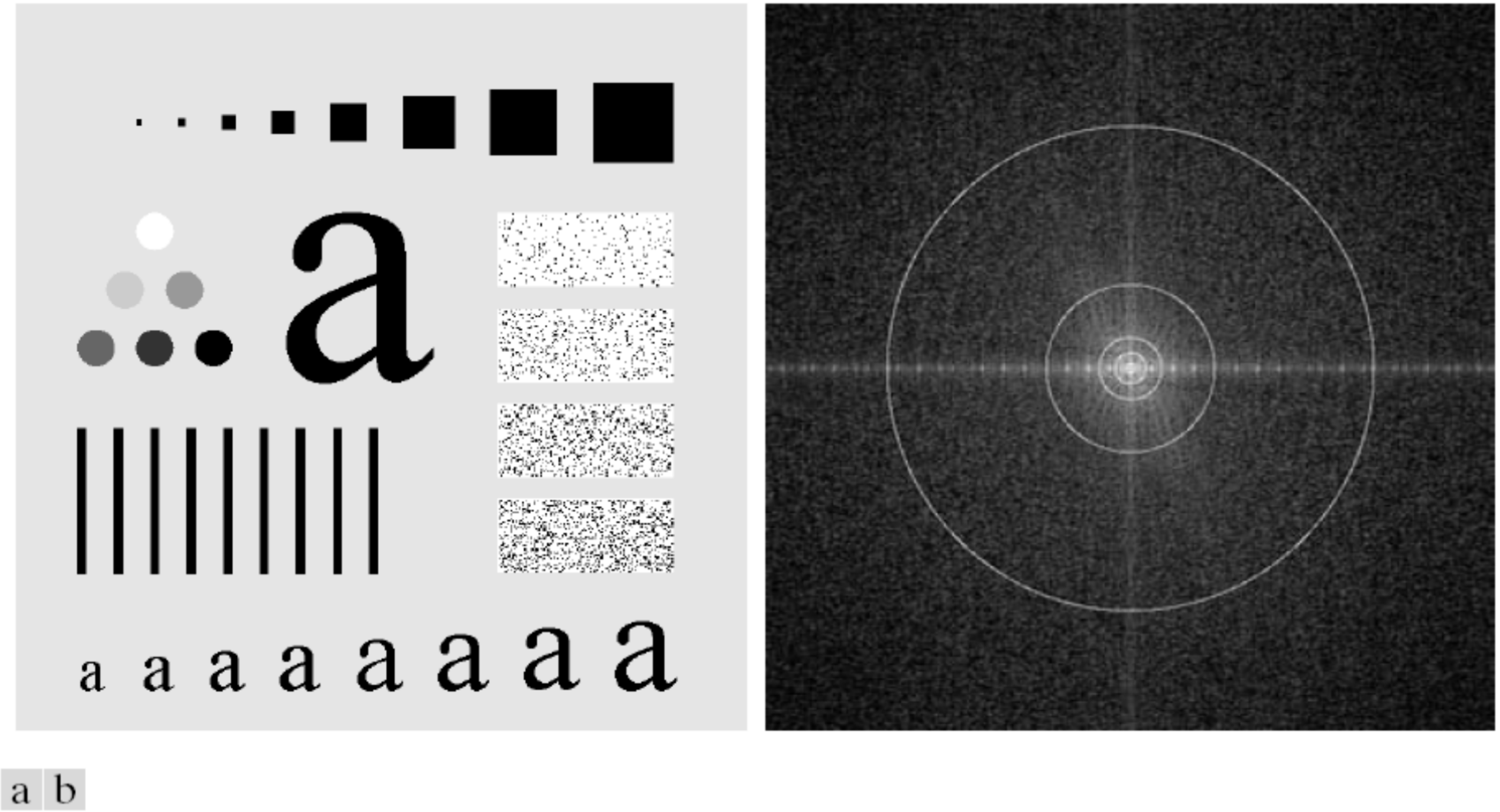
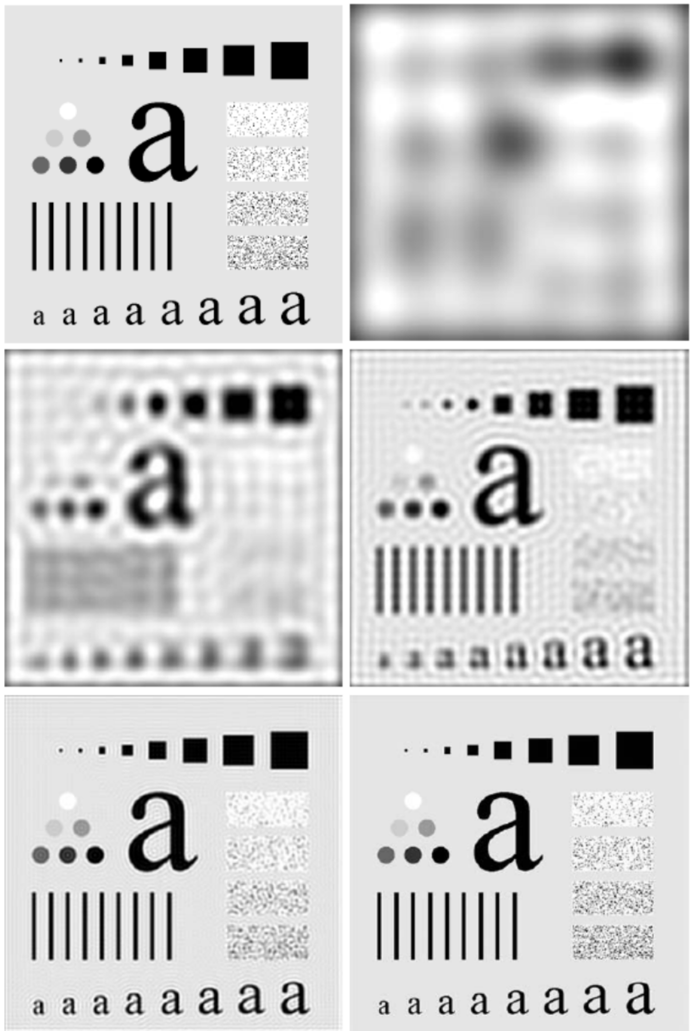


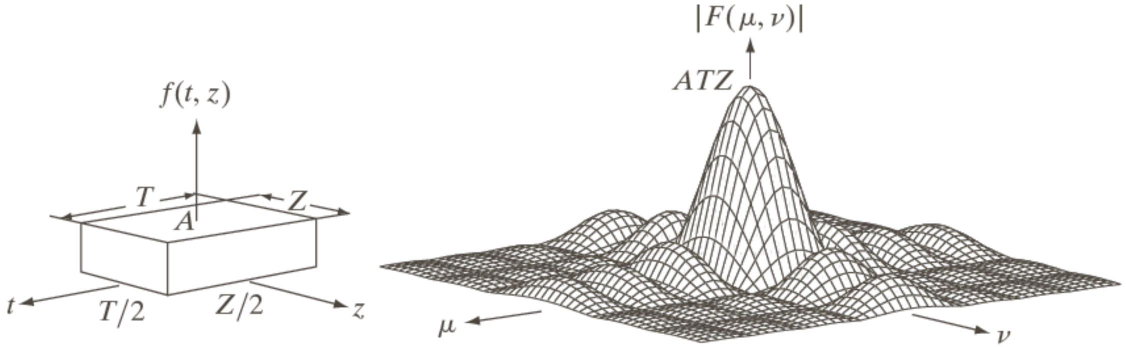
FIGURE 4.41 (a) Test pattern of size 688 × 688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the padded image power, respectively.

理想低通滤波器的例子



a b
c d
e f

FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.



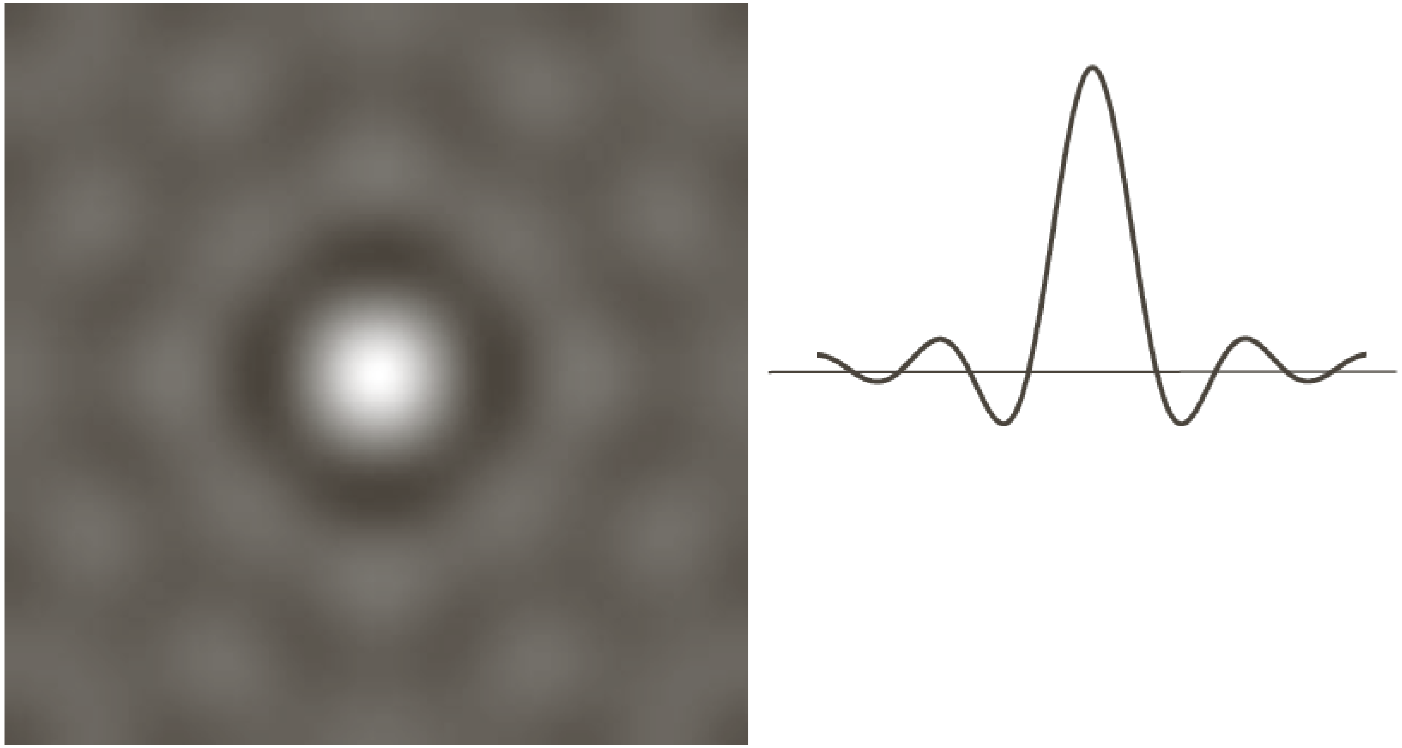
振铃现象（ringing effect）

```
f = imread('..\data\Fig0442(a)(characters_test_pattern).tif');
[M,N] = size(f);
P = max(2*[M N]);% Padding size.

D0 = 60;% Use the same parameters as Figure 4.42, (10,30,160,460)
H = ilpf(D0,P);

F = fftshift(fft2(f,P,P));
G = F.*H;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:5:end,1:5:end)),colormap('default');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G)),[]);
figure(6),imshow(g,[]);
```

```
function H = ilpf(D0,M)
% Create a ideal low pass filter

H = zeros(M,M);
[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2);
MASK = (D<=D0);
H(MASK) = 1;
```

ILPF的脉冲响应$h(x,y)$

脉冲响应$h(x,y)$即$H(x,y)$的IDFT



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.

```
D0=5;
P=1000;
H = ilpf(D0,P);
g = real(fftshift(ifft2(ifftshift(H))));
close all
figure(1),imshow(g,[]);
figure(2),mesh(g(1:25:end,1:25:end))
figure(3),plot(1:P,g(P/2+1,:));
```

巴特沃斯低通滤波器（Butterworth lowpass filter）

n阶巴特沃斯低通滤波器的转移函数为

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

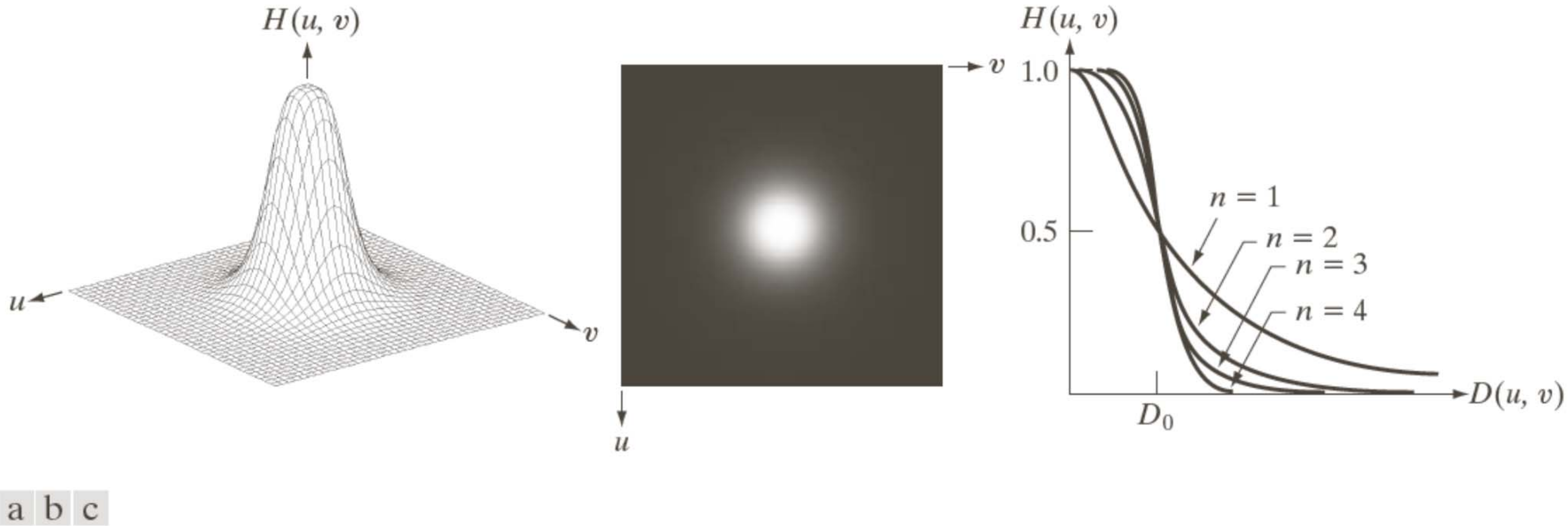


FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

BLPF的脉冲响应$h(x,y)$

脉冲响应$h(x,y)$即$H(x,y)$的IDFT

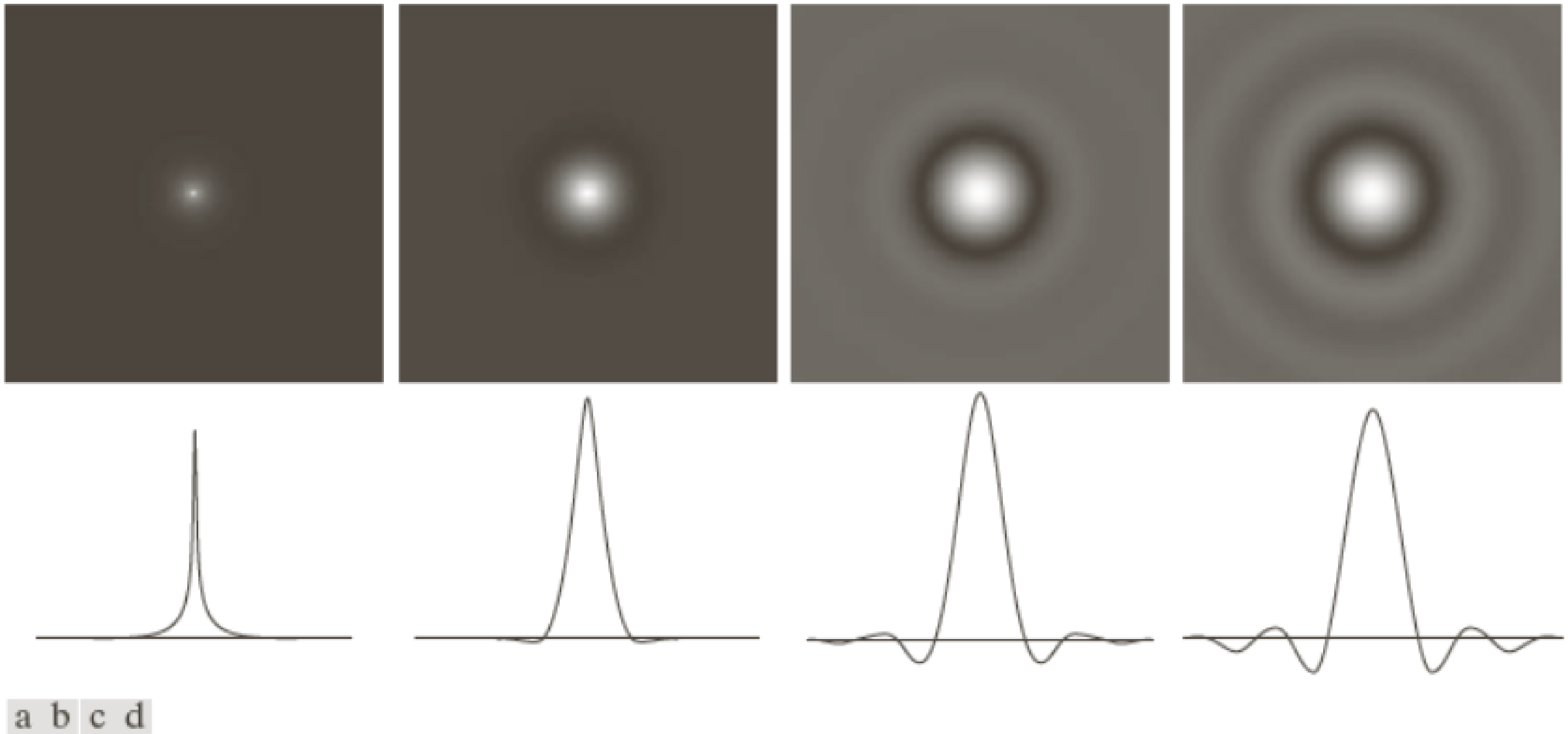


FIGURE 4.46 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding intensity profiles through the center of the filters (the size in all cases is 1000 × 1000 and the cutoff frequency is 5). Observe how ringing increases as a function of filter order.

```
D0=5;
P=1000;
ns = [1 2 5 20];
close all
for k = 1:length(ns)
    H = blpf(D0,ns(k),P);
    g = real(fftshift(ifft2(ifftshift(H))));

figure(1),subplot(2,2,k),imshow(g,[]),title(['\it{n}='
sprintf('%d',ns(k))]);
    figure(2),subplot(2,2,k),mesh(g(1:25:end,1:25:end))
    figure(3),subplot(2,2,k),plot(1:P,g(P/2+1,:));
end
```

```
function H = blpf(D0,n,M)
% Create a Butterworth low pass filter

[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2)/D0;
H = 1./(1+D.^(2*n));
```

BLPF滤波结果：不同截止频率

FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

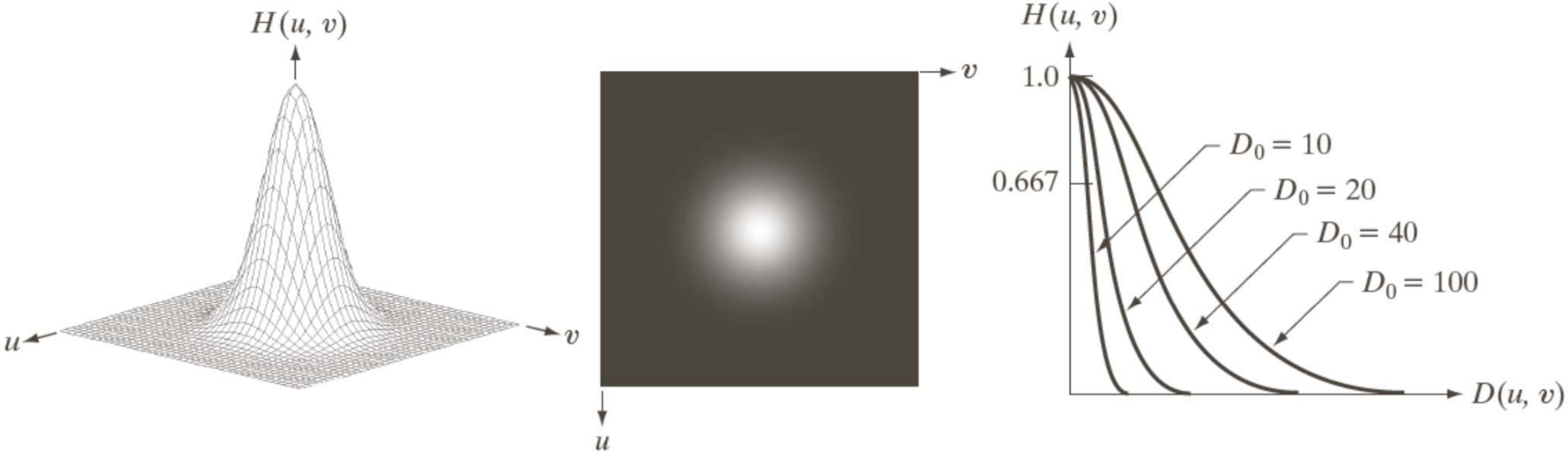
```
f = imread('..\data\Fig0442(a)(characters_test_pattern).tif');
[M,N] = size(f);
P = max(2*[M N]);% Padding size.

D0=30;
n=2;
H = blpf(D0,n,P);

F = fftshift(fft2(f,P,P));
G = F.*H;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:5:end,1:5:end)),colormap('default');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G)),[]);
figure(6),imshow(g,[]);
```

```
function H = blpf(D0,n,M)
% Create a Butterworth low pass filter

[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2)/D0;
H = 1./(1+D.^(2*n));
```

高斯低通滤波器

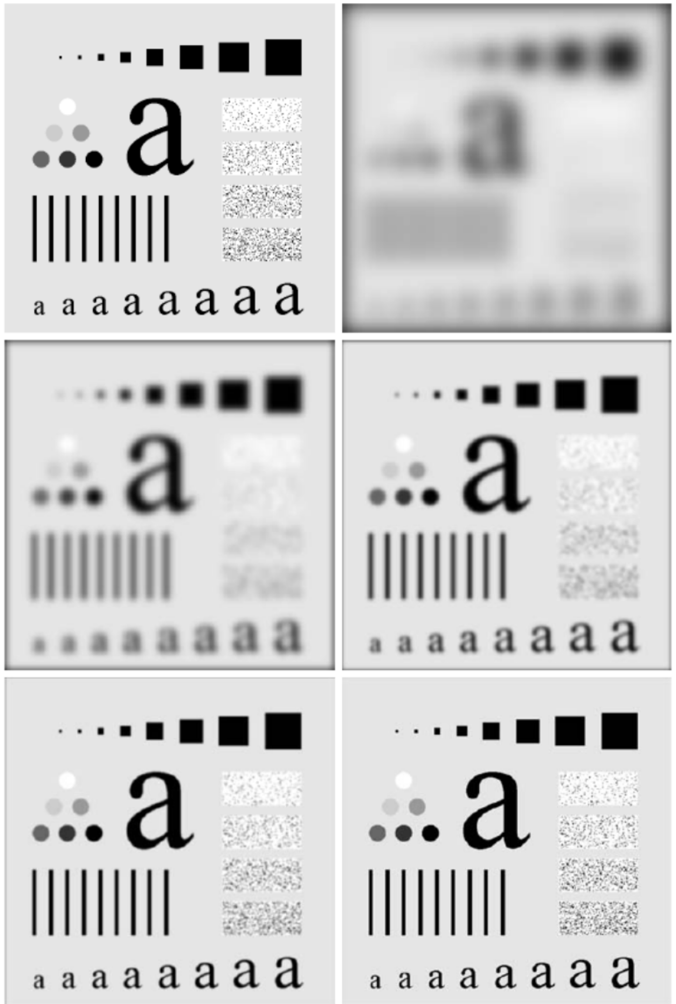
高斯低通滤波器的转移函数为

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



a b c

FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.



a b
c d
e f

FIGURE 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.

```matlab
f = imread('..\data\Fig0442(a)(characters_test_pattern).tif');
[M,N] = size(f);
P = max(2*[M N]);% Padding size.

D0=30;
H = glpf(D0,P);

F = fftshift(fft2(f,P,P));
G = F.*H;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:5:end,1:5:end)),colormap('default');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G)),[]);
figure(6),imshow(g,[]);
```

```matlab
function H = glpf(D0,M)
% Create a Gaussian low pass filter

[DX, DY] = meshgrid(1:M);
D2 = (DX-M/2-1).^2+(DY-M/2-1).^2;
H = exp(-D2/(2*D0*D0));
```

内 容

- 频域滤波基础
- 图像平滑
- 图像锐化
- 选择性滤波

图像锐化

- 图像中的边缘、突变对应着频域中的高频成分
- 通过弱化低频、强化高频，可实现图像锐化
- 下面介绍几类频域的图像锐化方法
  1. 三种高通滤波器
  2. 拉普拉斯算子
  3. 非锐化掩膜
  4. 同态滤波

高通滤波器$H_{\mathrm{HP}}(u,v)$可由对应的低通滤波器$H_{\mathrm{LP}}(u,v)$得到

$$H_{\mathrm{HP}}(u,v)=1-H_{\mathrm{LP}}(u,v)$$

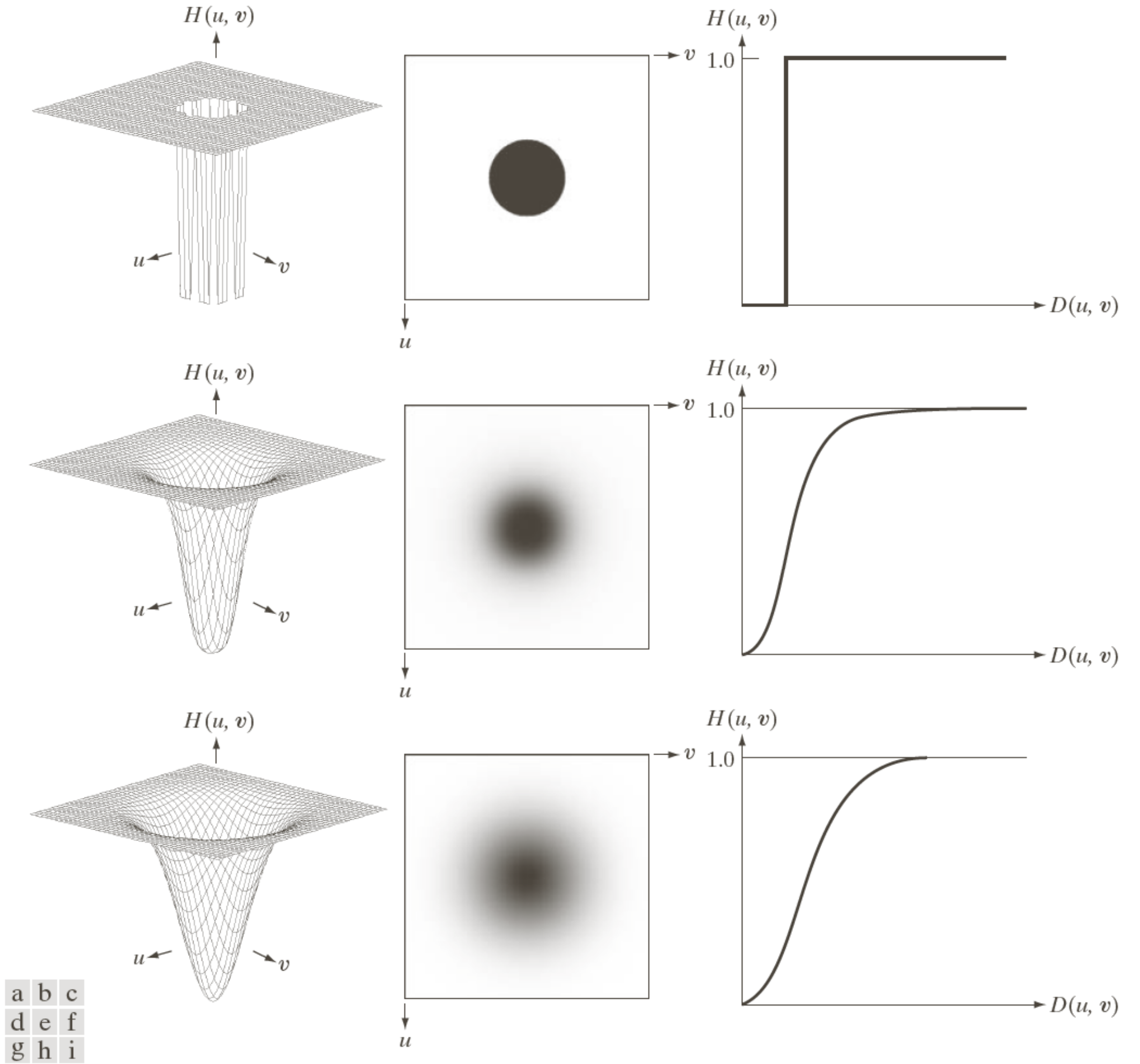


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

高通滤波器的脉冲响应

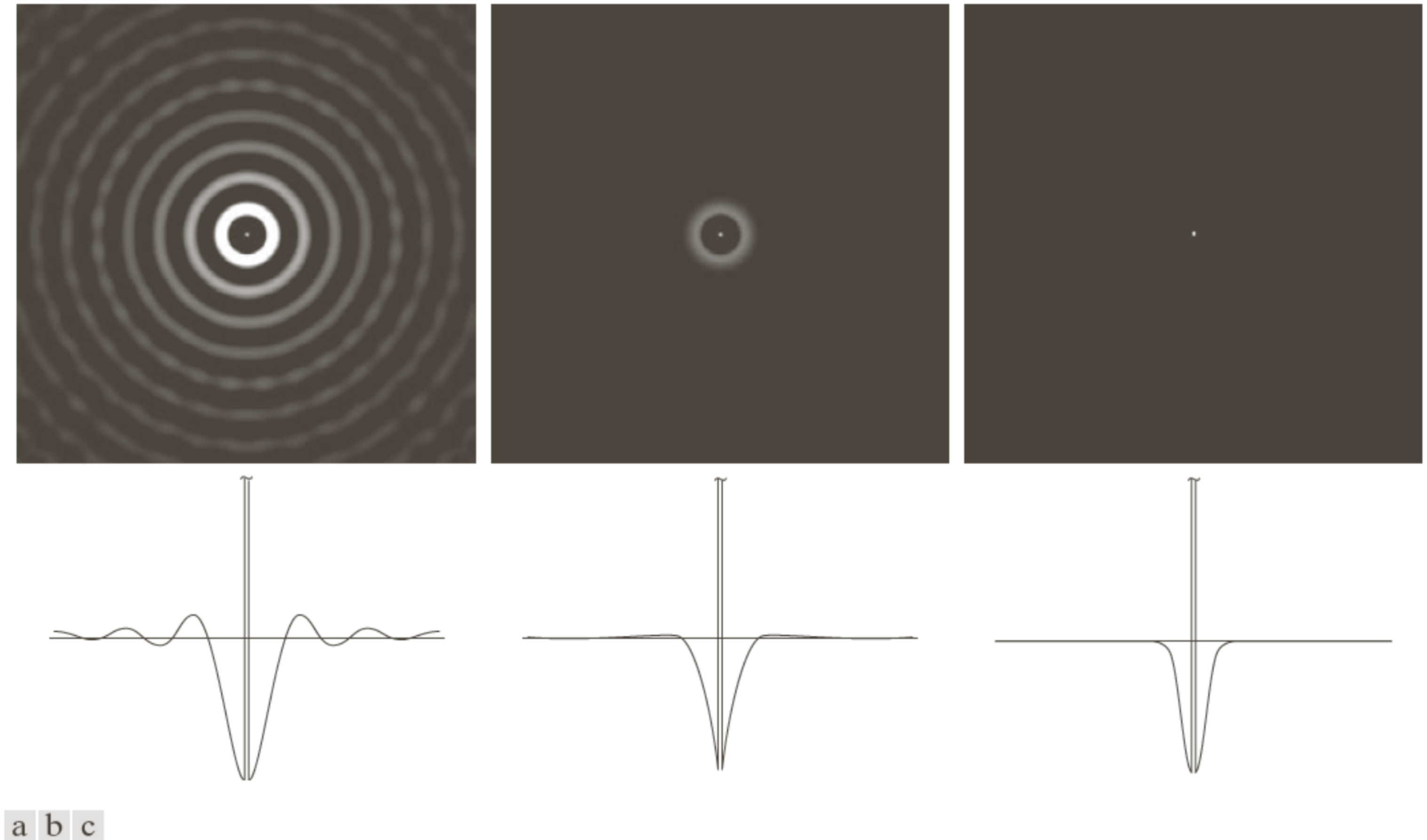


FIGURE 4.53 Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

理想高通滤波器

a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60$, and 160.

D_0越大，边缘越细

巴特沃斯高通滤波器

a b c

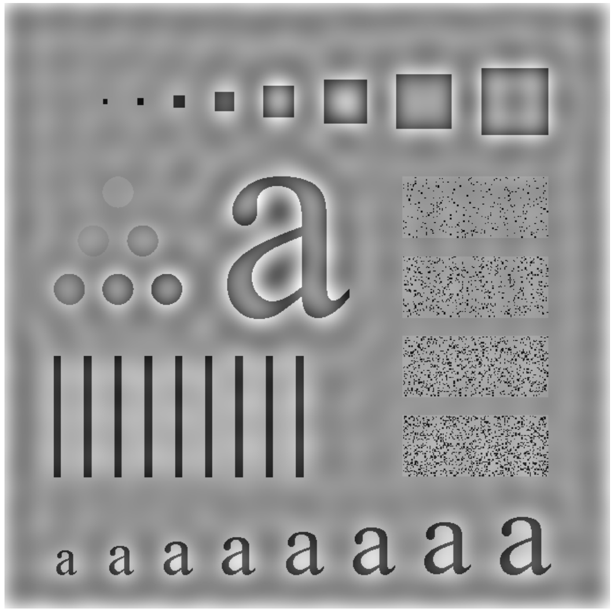
FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

高斯高通滤波器

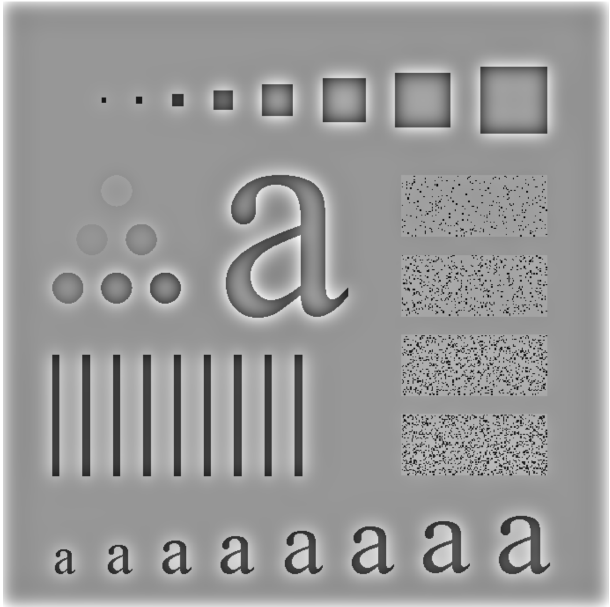
a b c

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60$, and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

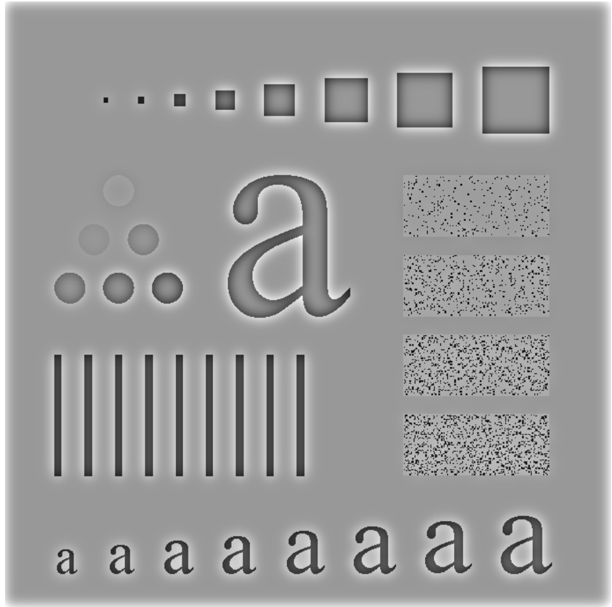
3种高通滤波器的对比



理想



巴特沃斯



高斯

- D_0均为30
- 理想高通滤波器的振铃现象明显

拉普拉斯算子

拉普拉斯算子$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$，各向同性

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

90度不变

0	1	0
1	−4	1
0	1	0

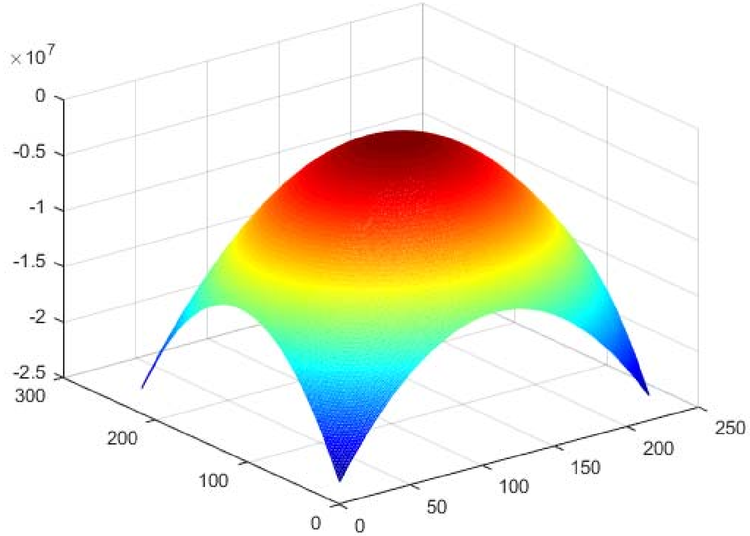
45度不变

1	1	1
1	−8	1
1	1	1

拉普拉斯算子的频域实现

拉普拉斯算子对应的转移函数为

$$H(u,v) = -4\pi^2(u^2+v^2)$$



证明：傅里叶变换的微分性质：

$$\frac{\partial^2 f(x,z)}{\partial t^2} \Leftrightarrow (j2\pi\mu)^2 F(\mu,\nu)$$

$$\frac{\partial^2 f(t,z)}{\partial z^2} \Leftrightarrow (j2\pi\nu)^2 F(\mu,\nu)$$

$$\mathfrak{F}\left[\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right] = -4\pi^2(\mu^2+\nu^2)F(\mu,\nu)$$

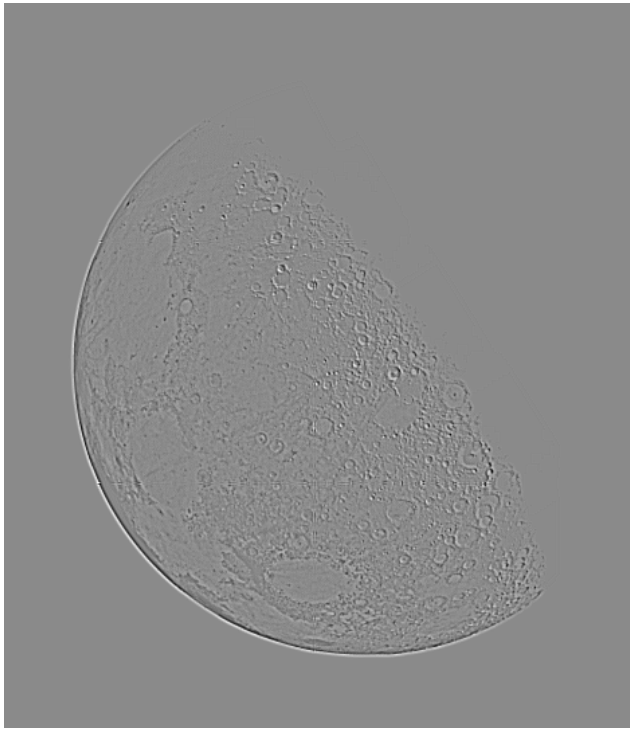
对于离散图像和傅里叶变换，上式替换为离散形式。

拉普拉斯算子的频域实现

$$\nabla^2 f(x,y) = \mathfrak{F}^{-1}[H(u,v)F(u,v)]$$
$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

由于$H(u,v)$数值大，$f(x,y)$和$\nabla^2 f(x,y)$都需要归一化处理

$f(x,y)$



$\nabla^2 f(x,y)$

$g(x,y)$

```matlab
f = imread('..\data\Fig0458(a)(blurry_moon).tif');
f = im2double(f);
[M,N] = size(f);
P = max(2*[M N]);% Padding size.
F = fftshift(fft2(f,P,P));

[DX, DY] = meshgrid(1:P);
H = -4*pi*pi * ((DX-P/2-1).^2 + (DY-P/2-1).^2);
G1 = H.*F;
g1 = real(ifft2(ifftshift(G1)));
g1 = g1(1:M,1:N);
g = f - g1/max(g1(:));

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:5:end,1:5:end)),colormap('jet');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G1)),[]);
figure(6),imshow(g1,[]);
figure(7),imshow(g,[]);
```

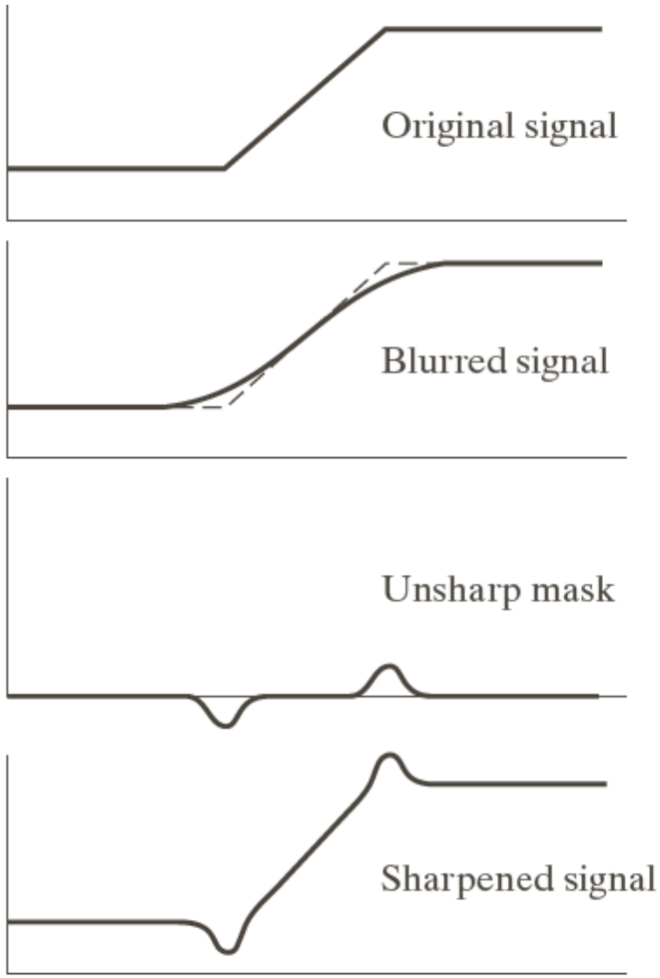
非锐化掩膜（Unsharp masking）

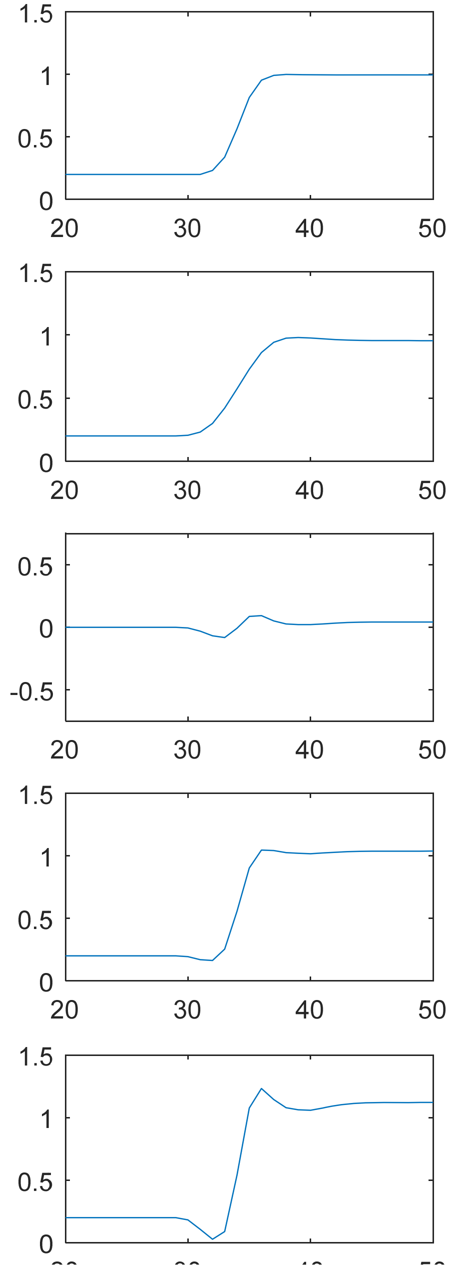
1. 平滑原图像$f(x,y)$
2. 从原图像中减去模糊图像$\bar{f}(x,y)$，得到掩膜图像$g_{\text{mask}}(x,y)$
3. 将掩膜加到原图像中

$$g(x,y) = f(x,y) + k * g_{\text{mask}}(x,y)$$

$k > 1$时，称为高提升滤波（highboost filtering）

非锐化掩膜（Unsharp masking）





$$k = 3$$

非锐化掩膜的频域实现

$$g_{\text{mask}}(x,y)=f(x,y)-f_{\text{LP}}(x,y)$$

其中，$f_{\text{LP}}(x,y)=\mathfrak{F}^{-1}[H_{\text{LP}}(u,v)F(u,v)]$，$H_{\text{LP}}(u,v)$为频域低通滤波器，$F(u,v)$为$f(x,y)$的DFT。

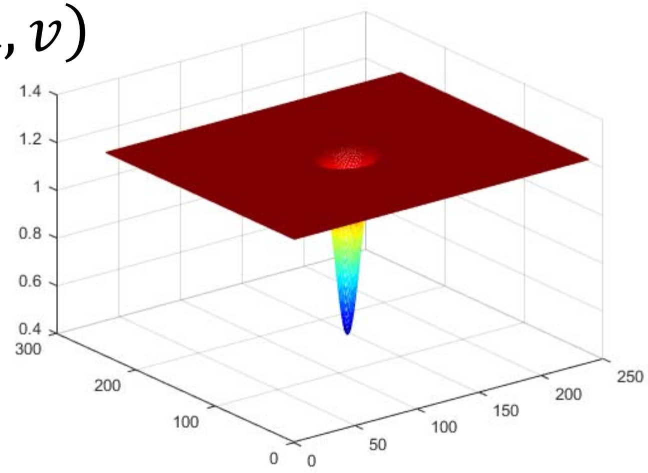
$$g(x,y)=f(x,y)+k*g_{\text{mask}}(x,y)$$

$$g(x,y)=\mathfrak{F}^{-1}\left[F(u,v)+k*\left(F(u,v)-H_{\text{LP}}(u,v)F(u,v)\right)\right]$$
$$=\mathfrak{F}^{-1}\{\left[1+k*\left(1-H_{\text{LP}}(u,v)\right)\right]F(u,v)\}$$
$$=\mathfrak{F}^{-1}\{[1+k*H_{\text{HP}}(u,v)]F(u,v)\}$$

高频强调滤波器（high-frequency-emphasis filter）：

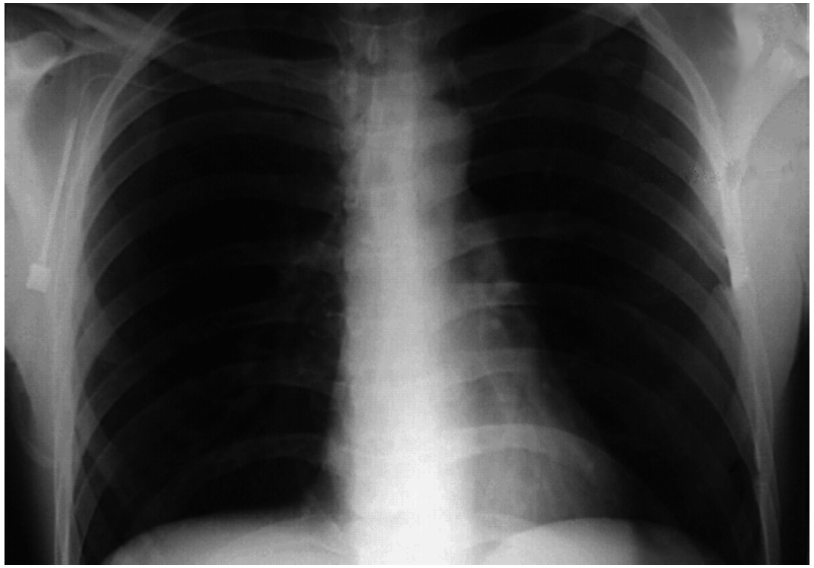
$$H_{\text{HFE}}(u,v)=k_1+k_2*H_{\text{HP}}(u,v)$$

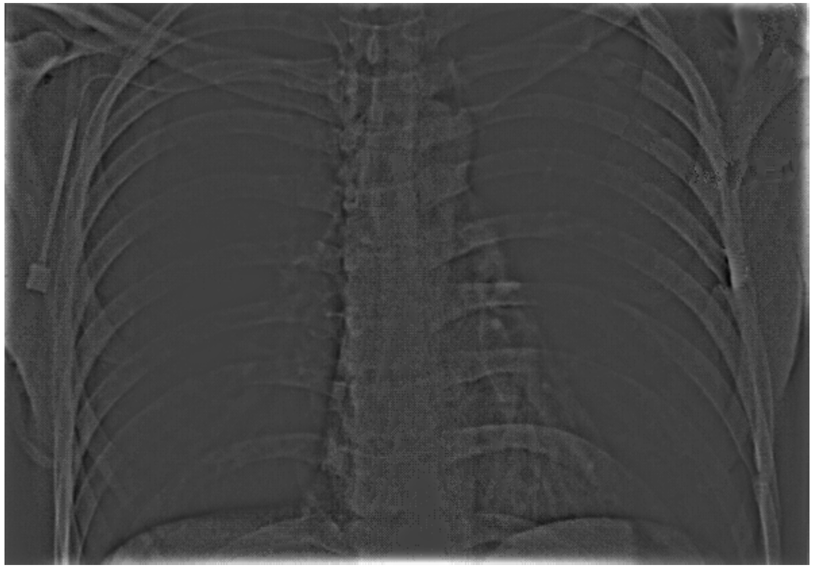
$0.5+0.75*H_{\text{GHPF}}(u,v)$



高频强调滤波应用

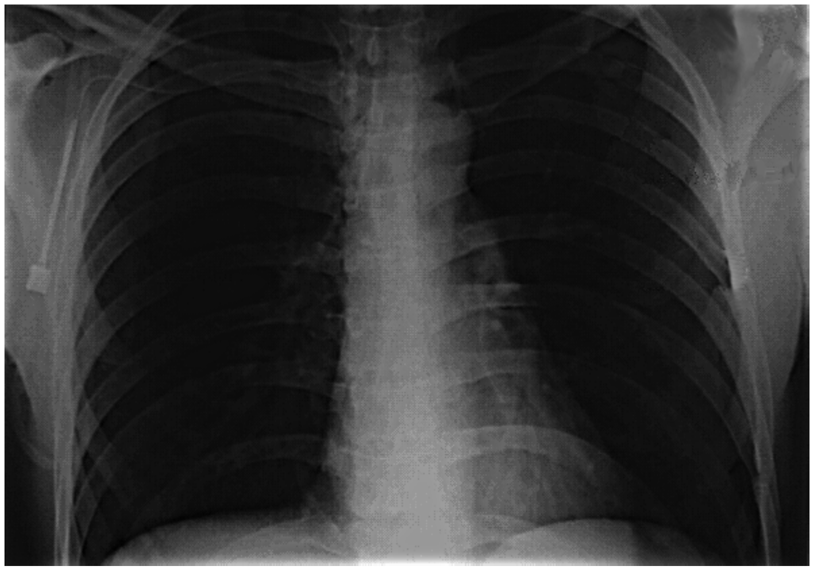
X光胸片原图

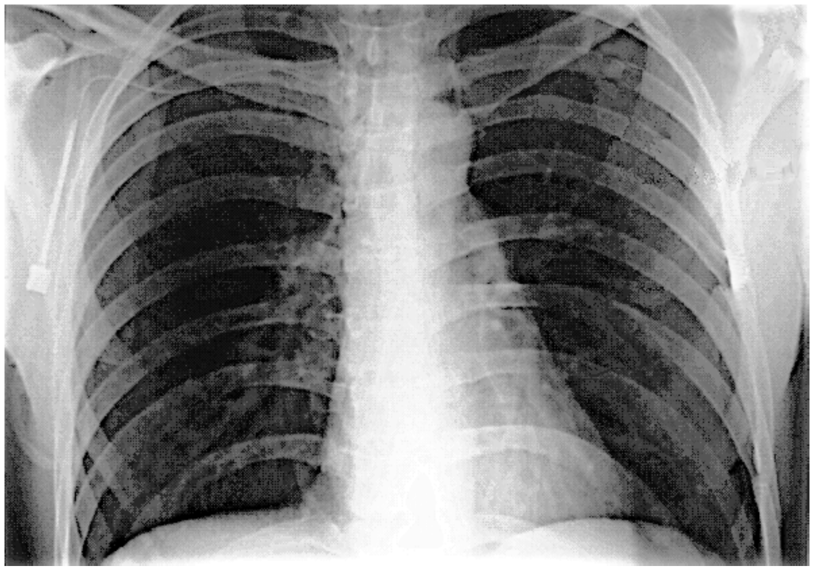




高通滤波

高频强调滤波



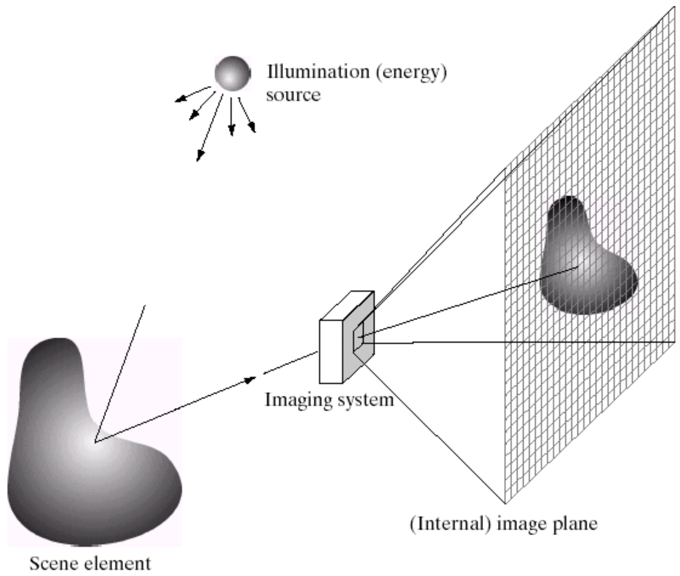


对左图做直方图均衡

```matlab
%ex0421_highFrequencyEmphasis
f = imread('..\data\Fig0459(a)(orig_chest_xray).tif');
f = im2double(f);
[M,N] = size(f);
P = max(2*[M N]);% Padding size.
F = fftshift(fft2(f,P,P));

D0=40;
H = 0.5+0.75*(1-glpf(D0,P));
G = H.*F;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

H_hp = 1-glpf(D0,P);
G_hp = H_hp.*F;
g_hp = real(ifft2(ifftshift(G_hp)));
g_hp = g_hp(1:M,1:N);

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:5:end,1:5:end)),colormap('jet');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G)),[]);
figure(6),imshow(g,[]);
g2 = histeq(g);
figure(7),imshow(g2);
figure(8),imshow(g_hp,[]);
% g3 = histeq(f);
% figure(8),imshow(g3);
```

同态滤波（Homomorphic filtering）

- 一个简单的成像模型：$f(x,y) = i(x,y)r(x,y)$，即图像等于入射光强 $i(x,y)$ 乘以反射系数 $r(x,y)$ 。
- $i(x,y)$取决于光源，$r(x,y)$取决于物体本身。
- 有时，光照不均匀会使图像质量不好。希望对光照进行处理。
- 通常可假设：光照变化缓慢，而反射系数变换剧烈。
- 但是，由于二者相乘，两个成分在频域分不开。



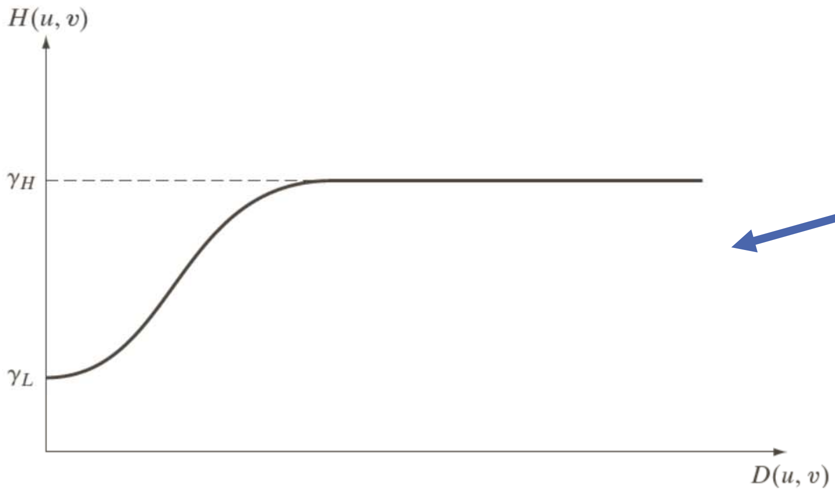
同态滤波

同态滤波的思想：取$f(x,y)$的对数，将乘变为和

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

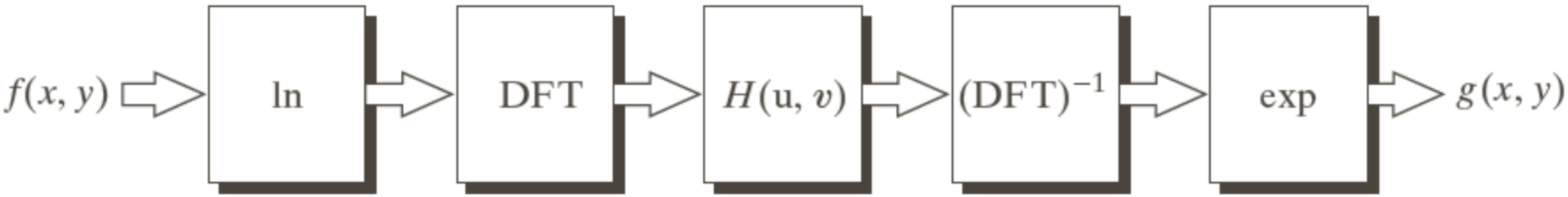
$$Z(u,v) = \mathfrak{F}[\ln i(x,y)] + \mathfrak{F}[\ln r(x,y)]$$

然后，设计频域滤波器，对低频和高频进行不同的处理。



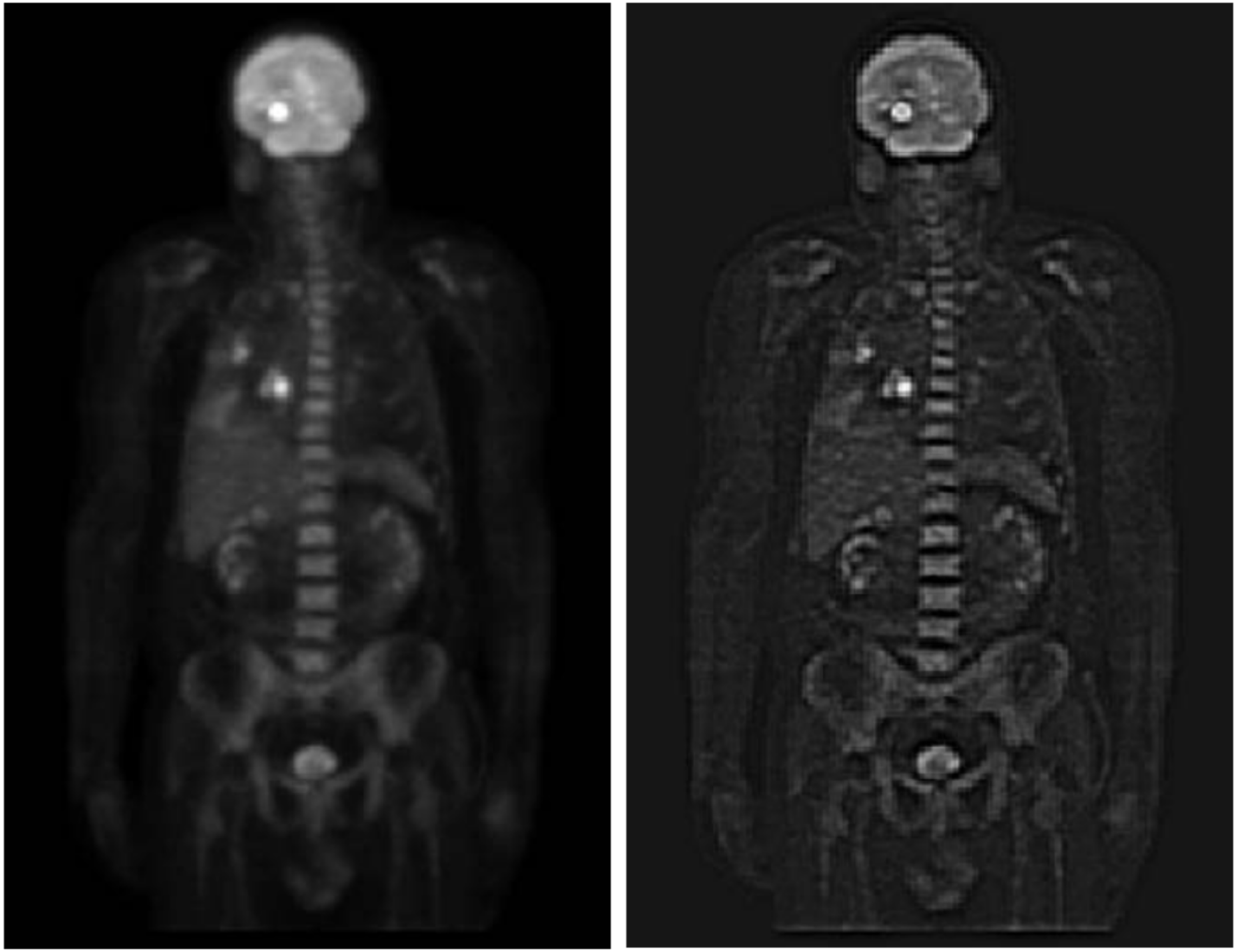
滤波器的径向横断线

$\gamma_L < 1$，$\gamma_H > 1$



同态滤波的流程图

同态滤波的应用（1）



a b

FIGURE 4.62
(a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI PET Systems.)

```
f = imread('..\data\Fig0462(a)(PET_image).tif');
f = double(f);
[M,N] = size(f);
z = log(f+1);

P = max([M N]);
zp = padarray(z,[P-M P-N],0,'post');

Z = fftshift(fft2(zp));

% filter
[X, Y] = meshgrid(1:P);
D0 = 80;
gamma_h = 2;
gamma_l = 0.25;
c = 1;
H = (gamma_h-gamma_l) * (1-exp(-c*((X-P/2-1).^2+(Y-P/2-1).^2)/(D0*D0))) + gamma_l;

% filtering
S = Z.*H;

figure(1),imshow(f,[]),title('f(x,y)')
figure(2),imshow(log(1+abs(S)),[]),title('|S(u,v)|')
figure(3),surf(X(1:10:end,1:10:end),Y(1:10:end,1:10:end),H(1:10:end,1:10:end)),tit
le('H(u,v)')
figure(4),imshow(H,[]),title('H(u,v)')

sp = real(ifft2(ifftshift(S)));
s = sp(1:M,1:N);
g = exp(s-1);
figure(5),imshow(g,[]),title('g(x,y)')
```

同态滤波的应用（2）

- 光照变化对于人脸识别算法是难题
- 不同人类似光照下的相似度>同一人不同光照下的相似度
- 有人提出用同态滤波将光照归一化

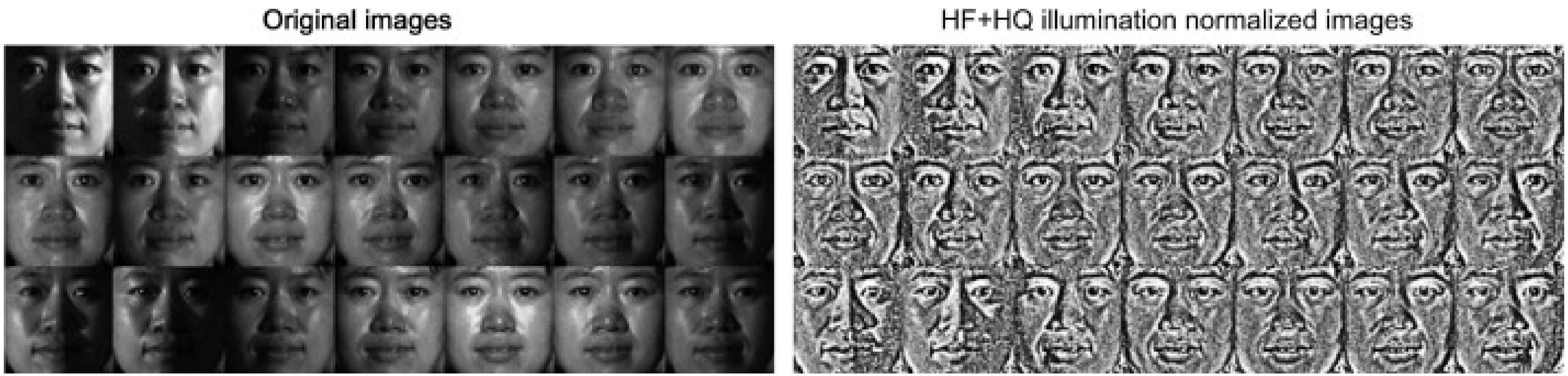


Fig. 17. Original images and their illumination normalized images obtained by the proposed method (HF + HQ) on the CMU PIE face database.

Chun-Nian Fan, Fu-Yan Zhang, Homomorphic filtering based illumination normalization method for face recognition, Pattern Recognition Letters, Volume 32, Issue 10, 15 July 2011, Pages 1468-1479. http://www.sciencedirect.com/science/article/pii/S0167865511001000

内 容

- 频域滤波基础
- 图像平滑
- 图像锐化
- 选择性滤波

选择性滤波（selective filtering）

- 带通滤波器（bandpass filter）
- 带阻滤波器（bandreject filter）
- 陷波滤波器（notch filter）
- Gabor滤波器

理想带通和带阻滤波器

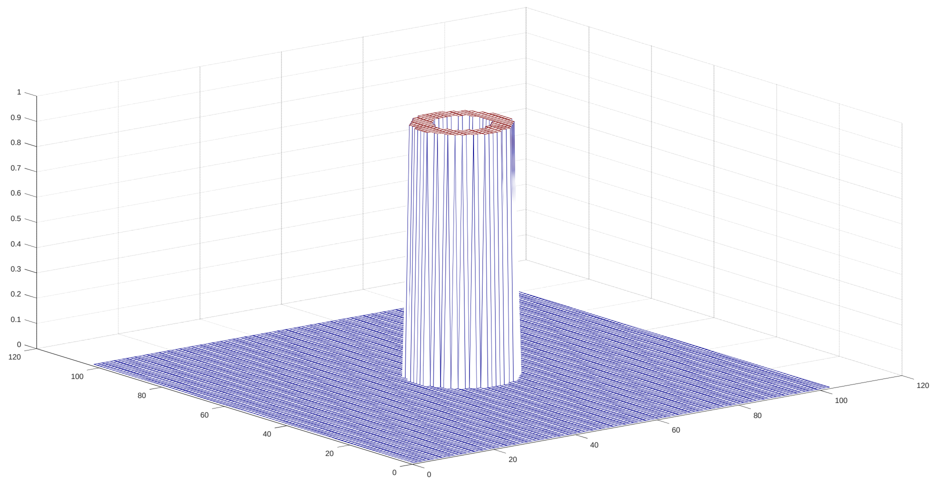
理想带通滤波器的转移函数为

$$H_{\text{IBPF}}(u,v)=\begin{cases}1 & \text{if } D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\ 0 & \text{otherwise}\end{cases}$$

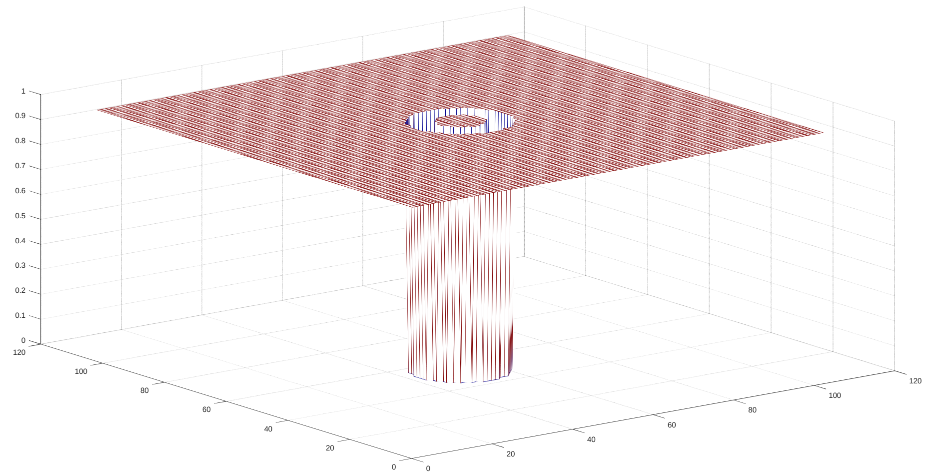
$$D(u,v)=\sqrt{(u-P/2)^2+(v-Q/2)^2}$$

参数D_0为频率中心，W为带宽

理想带阻滤波器的转移函数为$1-H_{\text{IBPF}}(u,v)$

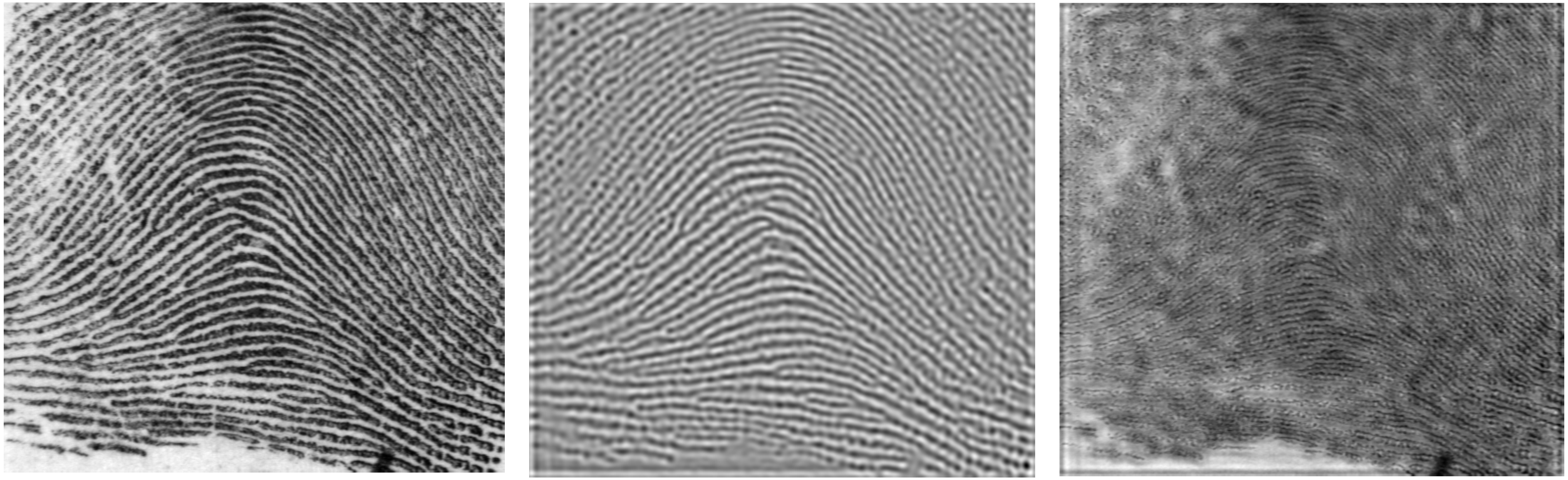


带通



带阻

理想带通和带阻滤波器的应用



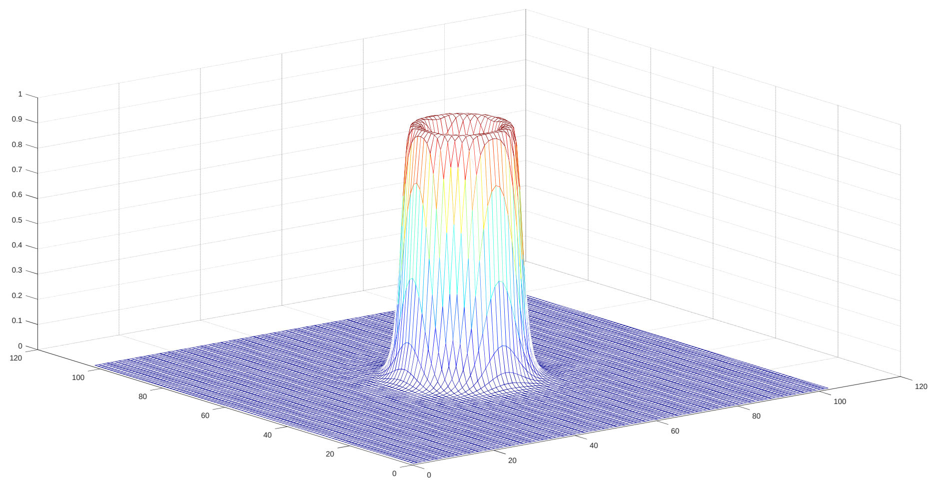
$D_0 = 90$，$W = 70$

巴特沃斯带通和带阻滤波器

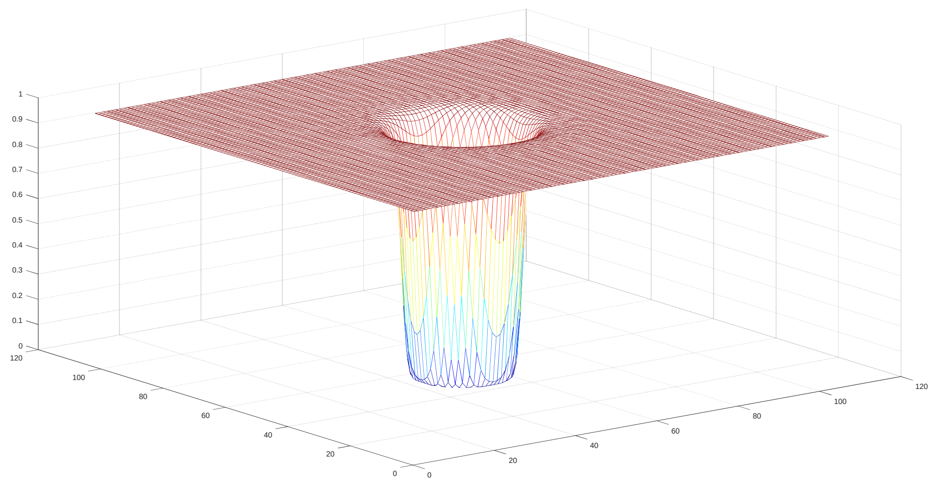
巴特沃斯带通滤波器的转移函数为

$$H_{\mathrm{BBPF}}(u,v)=\frac{1}{1+\left[\frac{D^2-{D_0}^2}{DW}\right]^{2n}}$$

巴特沃斯带阻滤波器的转移函数为$1-H_{\mathrm{BBPF}}(u,v)$

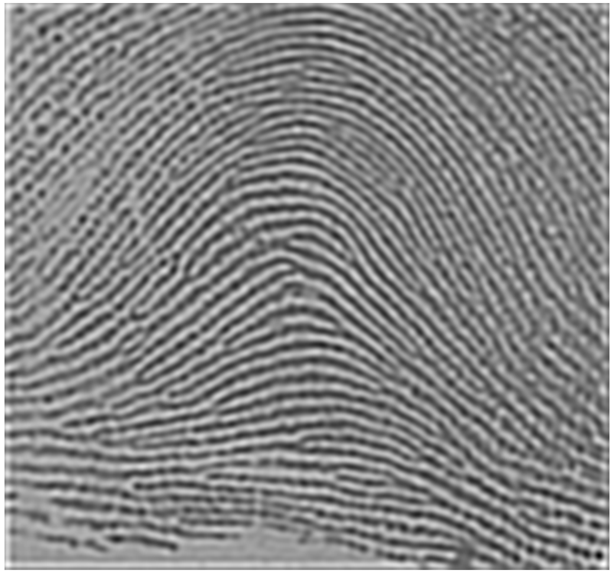


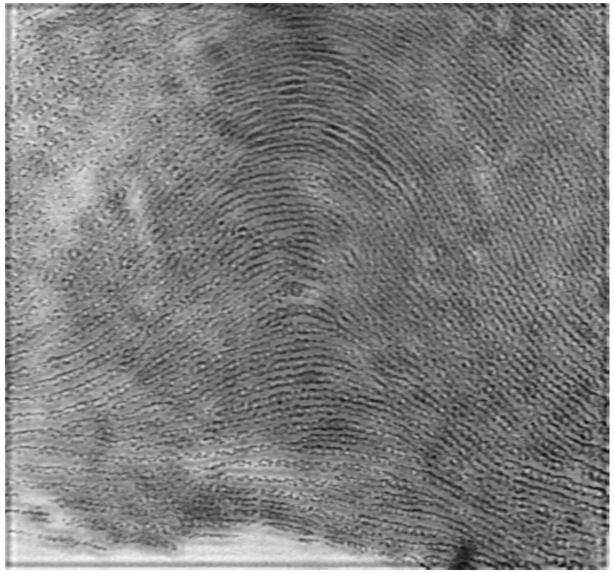
带通



带阻

巴特沃斯带通和带阻滤波器的应用





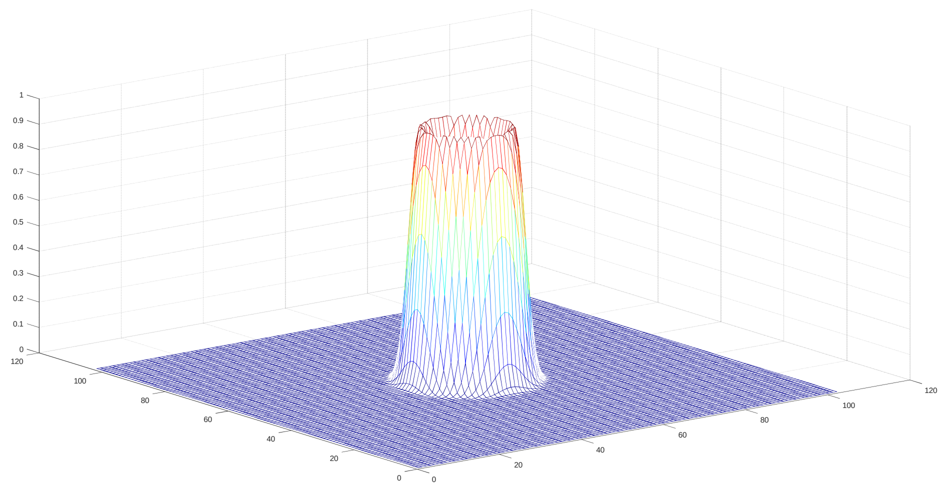
$D_0 = 90$，$W = 70$

高斯带通和带阻滤波器

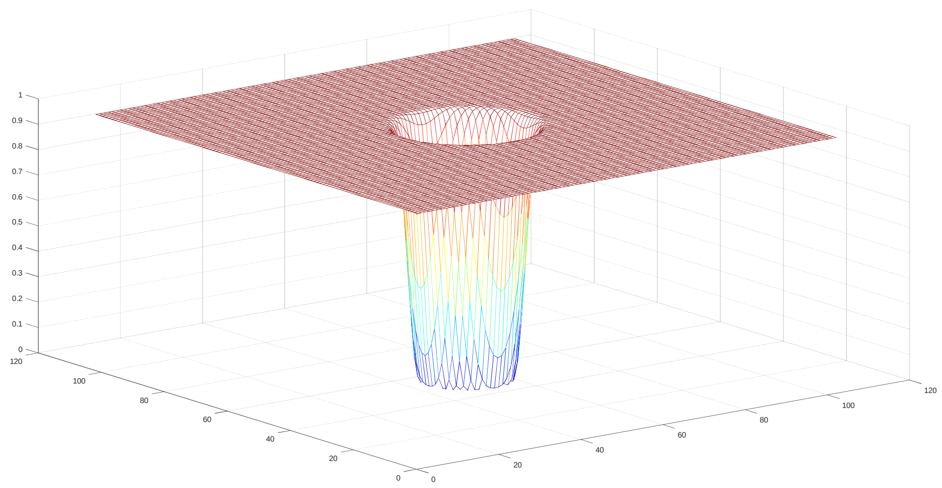
高斯带通滤波器的转移函数为

$$H_{\mathrm{GBPF}}(u,v)=e^{-\left[\frac{D^2-D_0{}^2}{DW}\right]^2}$$

高斯带阻滤波器的转移函数为$1-H_{\mathrm{GBPF}}(u,v)$

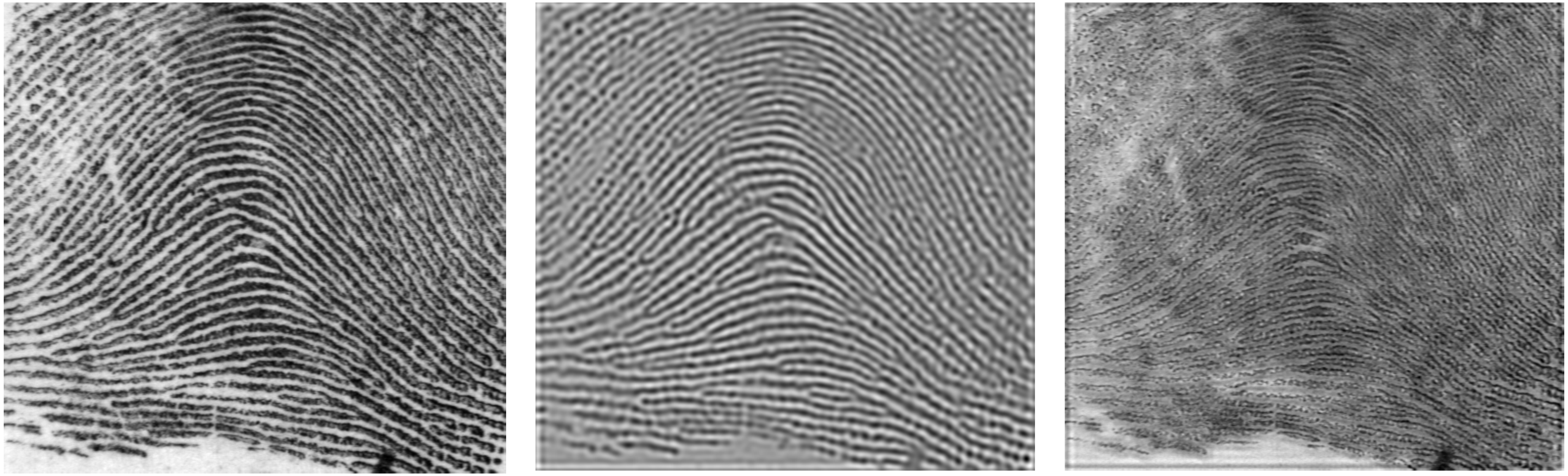


带通



带阻

高斯带通和带阻滤波器的应用



$D_0 = 90$，$W = 70$

```matlab
f = imread('..\data\F0020.bmp');
[M,N] = size(f);
P = max(2*[M N]);% Padding size.

D0=90;
W=70;
n=2;
% H = 1-ibpf(D0,W,P);
H = 1-bbpf(D0,W,n,P);
% H = gbpf(D0,W,P);

F = fftshift(fft2(f,P,P));
G = F.*H;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);
figure(3),mesh(H(1:10:end,1:10:end)),co
lormap('jet');
figure(4),imshow(H,[]);
figure(5),imshow(log(1+abs(G)),[]);
figure(6),imshow(g,[]);
```

```matlab
function H = ibpf(D0,W,M)
% Create a ideal band pass filter

H = zeros(M,M);
[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2);
MASK = (D<=D0+W/2) & (D>=D0-W/2);
H(MASK) = 1;
```

```matlab
function H = bbpf(D0,W,n,M)
% Create a Butterworth band pass filter

[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2);
H = 1./(1+((D.^2-D0^2)/(D0*W)).^(2*n));
```

```matlab
function H = gbpf(D0,W,M)
% Create a Gaussian band pass filter

[DX, DY] = meshgrid(1:M);
D = sqrt((DX-M/2-1).^2+(DY-M/2-1).^2);
H = exp(-((D.^2-D0^2)/(D0*W)).^2);
```

陷波滤波器（notch filter）

- 陷波滤波器阻断（或通过）指定频率附近的频率成分。
- 陷波阻断滤波器由成对的高通滤波器组成

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

$H_k(u,v)$和$H_{-k}(u,v)$分别为中心位于(u_k,v_k)和$(-u_k,-v_k)$的高通滤波器。成对是考虑到图像的DFT是共轭对称的。

- 例如，巴特沃斯陷波阻断滤波器（notch reject filter）：

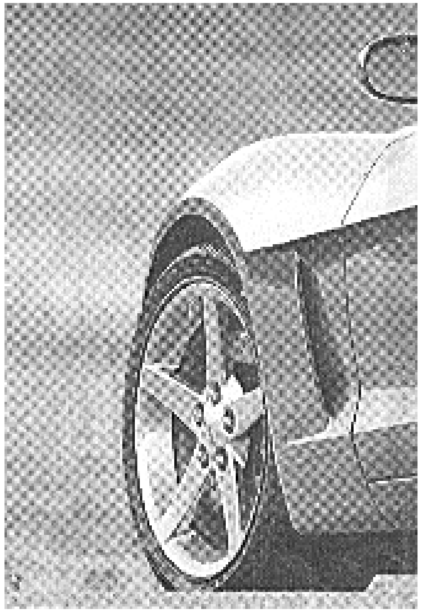
$$H_{\mathrm{NR}}(u,v)=\prod_{k=1}^{3} \frac{1}{1+[D_{0k}/D_k(u,v)]^{2n}} \frac{1}{1+[D_{0k}/D_{-k}(u,v)]^{2n}}$$

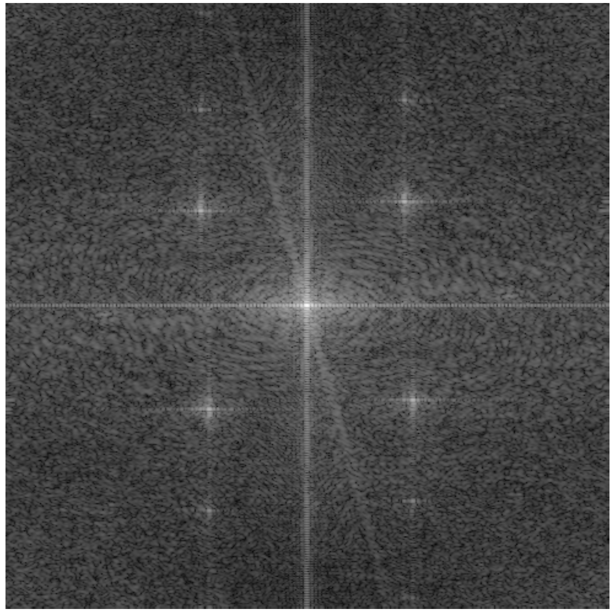
$$D_k(u,v)=\sqrt{(u-P/2-u_k)^2+(v-Q/2-v_k)^2}$$

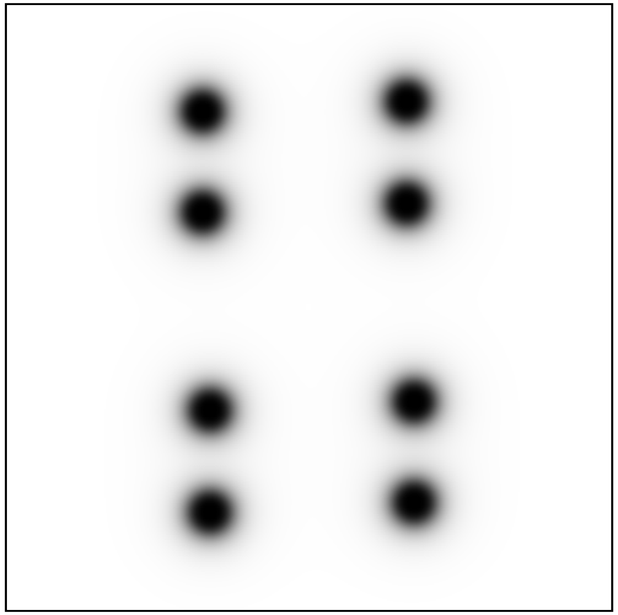
$$D_{-k}(u,v)=\sqrt{(u-P/2+u_k)^2+(v-Q/2+v_k)^2}$$

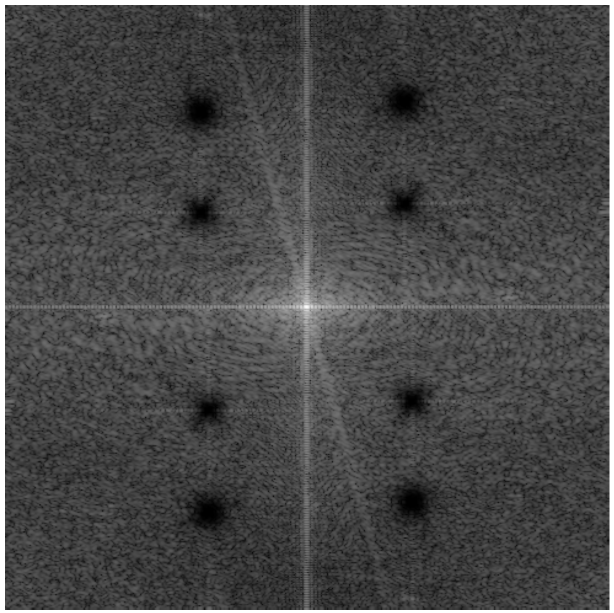
$$H_{\mathrm{NP}}(u,v)=1-H_{\mathrm{NR}}(u,v)$$

陷波滤波器的应用









```matlab
%ex0423_notch
f = imread('..\data\Fig0464(a)(car_75DPI_Moire).tif');
f = im2double(f);
[M,N] = size(f);
P = max(2*[M N]);% Padding size.
F = fftshift(fft2(f,P,P));

close all
figure(1),imshow(f,[]);
figure(2),imshow(log(1+abs(F)),[]);

% Creat 4 pairs of notch reject filters
p = [327 163; 327 80; 333 324; 333 406];% locations of maxima, found by imtool
H = ones(P,P);
[DX, DY] = meshgrid(1:P);
D0 = 20;
for k = 1:4
    Dk1 = sqrt((DX-p(k,1)).^2+(DY-p(k,2)).^2);
    Dk2 = sqrt((DX-P-2+p(k,1)).^2+(DY-P-2+p(k,2)).^2);
    H1 = 1./(1+(D0./Dk1).^(2*n));
    H2 = 1./(1+(D0./Dk2).^(2*n));
    H = H.*H1.*H2;
end
figure(3),imshow(H,[]);

% Filtering
G = H.*F;
g = real(ifft2(ifftshift(G)));
g = g(1:M,1:N);

figure(4),imshow(log(1+abs(G)),[]);
figure(5),imshow(g,[]);
```

Gabor滤波器

- Gabor滤波器（简化版，角度为0度，长宽比为1）

$$h(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}e^{-j2\pi fx}$$

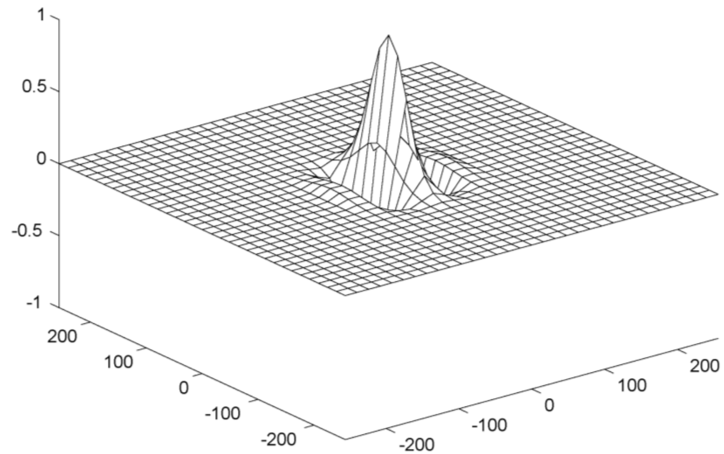
- 实部偶对称

$$h_{\mathrm{e}}(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}\cos(2\pi fx)$$

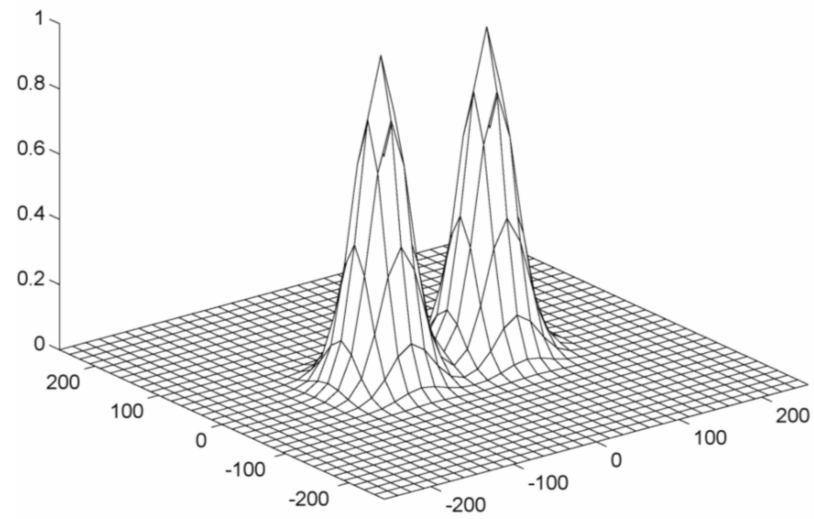
- 虚部奇对称

$$h_{\mathrm{o}}(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}\sin(2\pi fx)$$

- 偶对称Gabor滤波器是限波带通滤波器



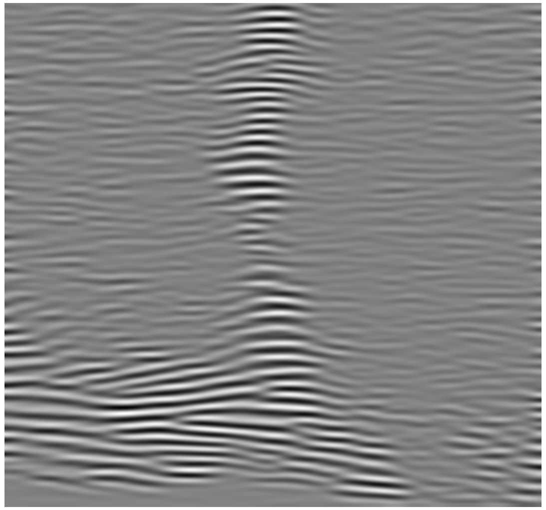
$h_{\mathrm{e}}(u,v)$

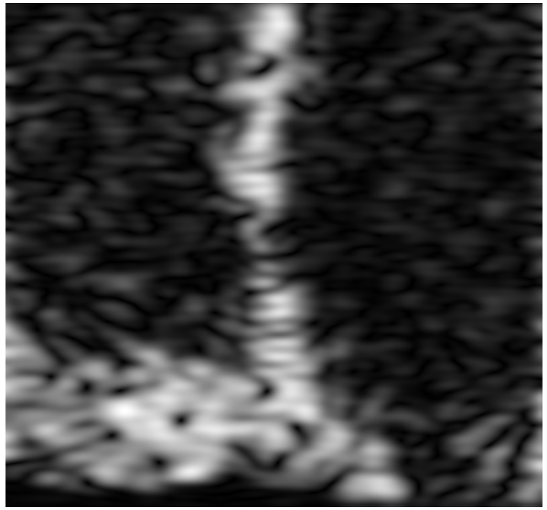


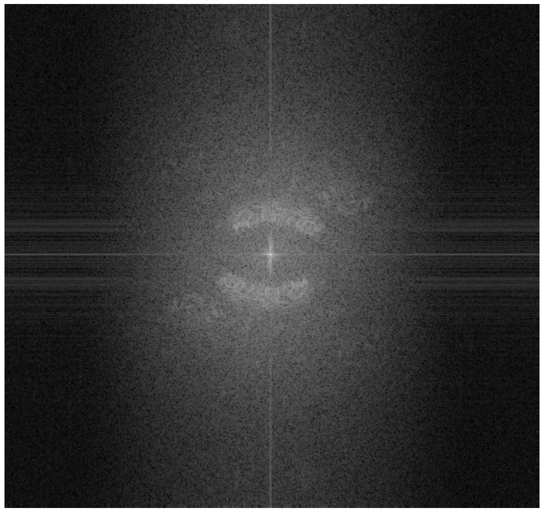
转移函数$F_{\mathrm{e}}(u,v)$

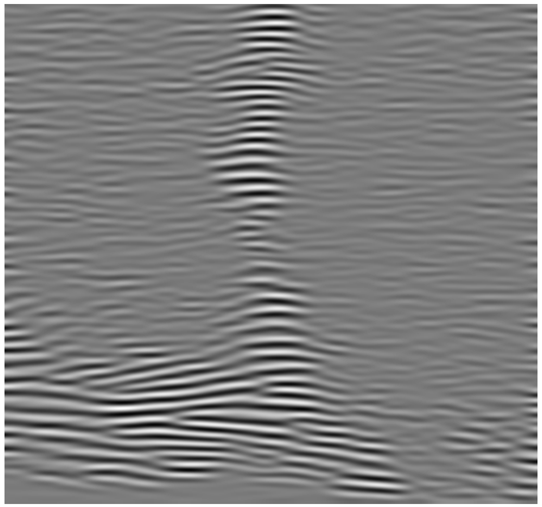
Gabor滤波器的应用

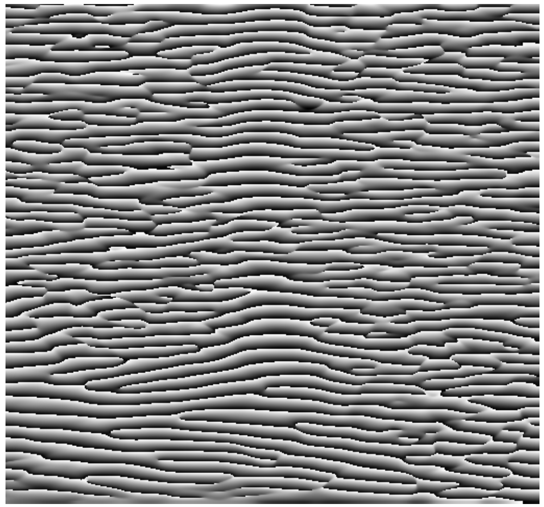

Original image


Gabor real


Gabor magnitude


Magnitude spectrum

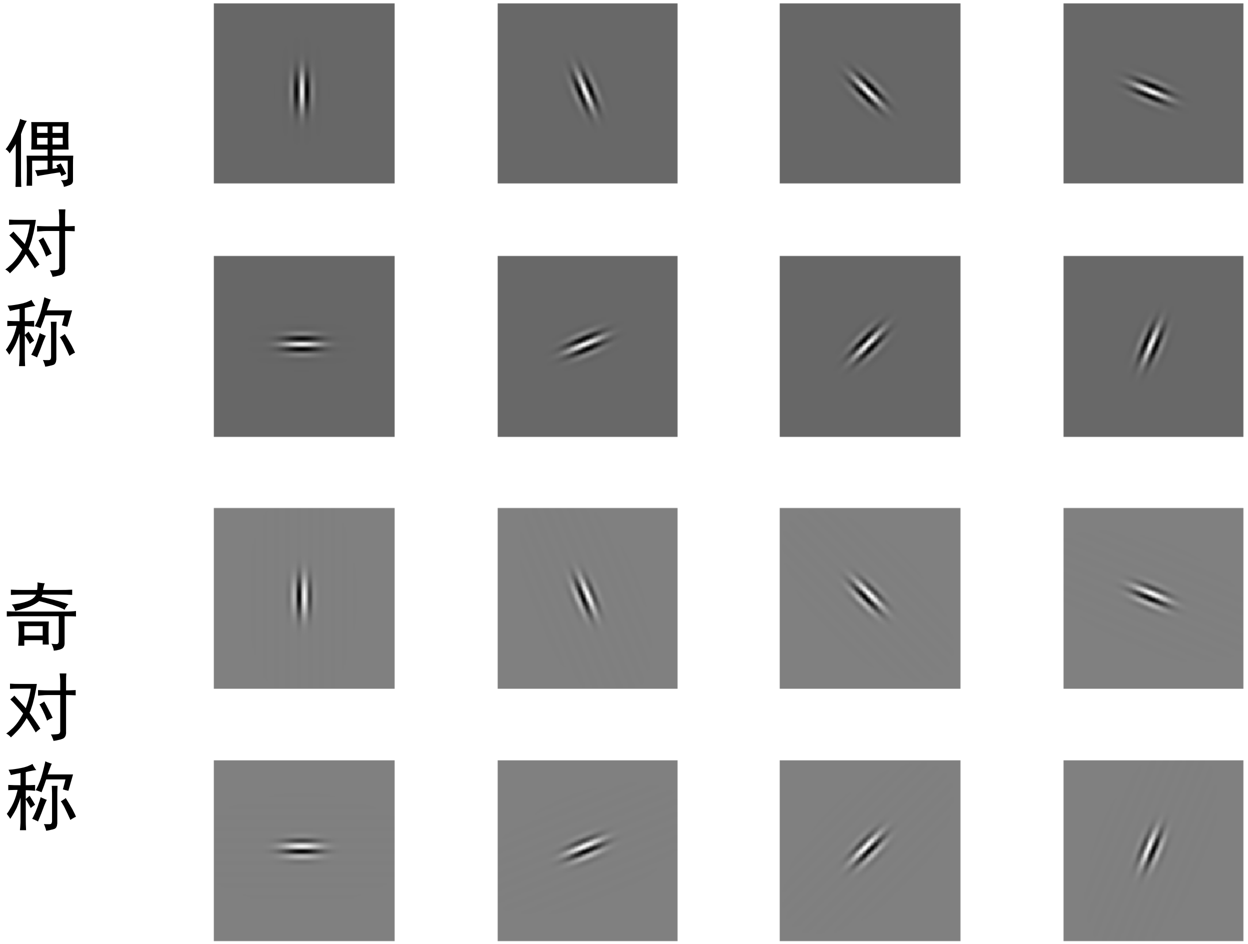

Gabor imaginary


Gabor phase

```
I = imread('..\data\F0020.bmp');
wavelength = 10;
orientation = 90;
[mag,phase] = imgaborfilt(I,wavelength,orientation);

figure(1),clf
subplot(2,3,1),imshow(I),title('Original image');
subplot(2,3,2),imshow(mag,[]),title('Gabor magnitude');
subplot(2,3,3),imshow(phase,[]),title('Gabor phase');
subplot(2,3,4),imshow(mag.*cos(phase),[]),title('Gabor real');
subplot(2,3,5),imshow(mag.*sin(phase),[]),title('Gabor imaginary');
```

Gabor滤波器组

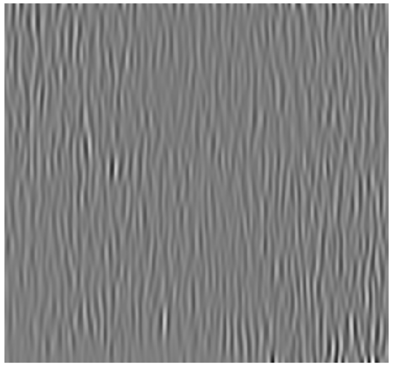
- Gabor滤波器常常以滤波器组的形式使用
- 波长10像素，8个方向的Gabor滤波器



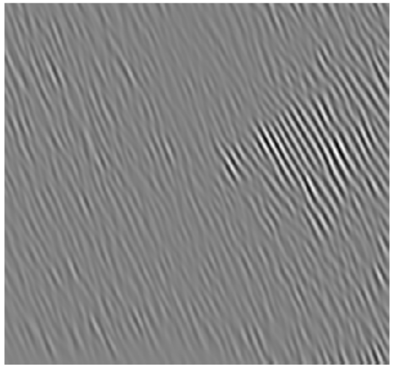
Gabor滤波器组

偶对称Gabor滤波器的滤波结果

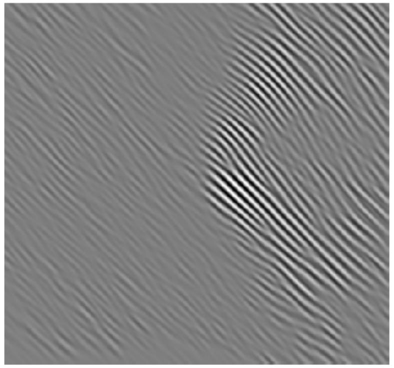
Orientation=0.0, Wavelength=10



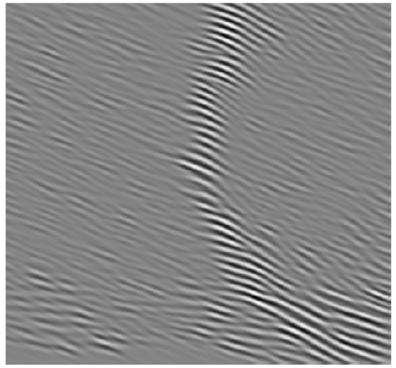
Orientation=22.5, Wavelength=10



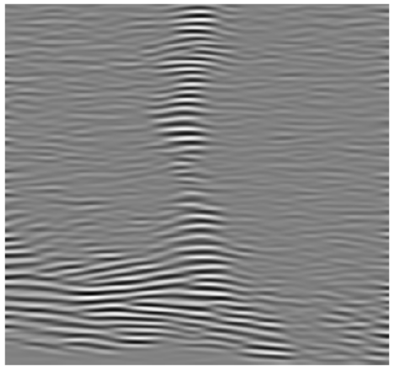
Orientation=45.0, Wavelength=10



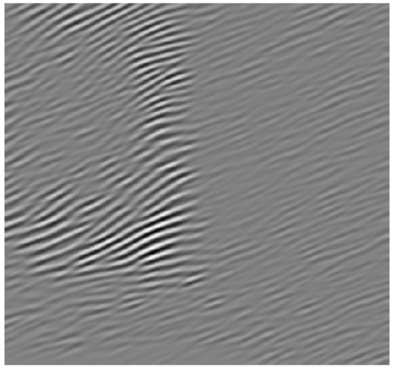
Orientation=67.5, Wavelength=10



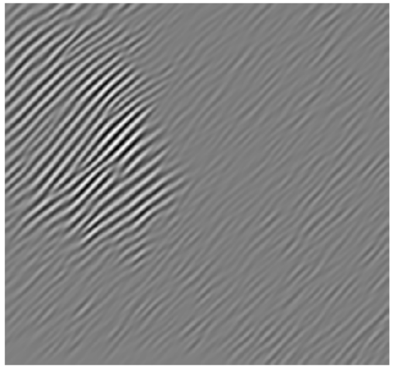
Orientation=90.0, Wavelength=10



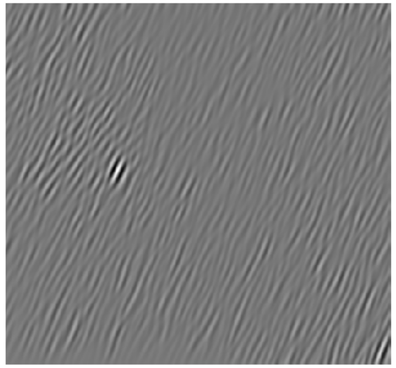
Orientation=112.5, Wavelength=10



Orientation=135.0, Wavelength=10



Orientation=157.5, Wavelength=10



```
% Gabor filter
I = imread('..\data\F0020.bmp');

% Create array of Gabor filters. This filter bank contains two orientations and two
wavelengths.
orientation_num = 8;
orientations = [0:orientation_num-1]*180/orientation_num;
wavelengths = 10;
gaborArray = gabor(wavelengths,orientations);
% Apply filters to input image.
[mag,phase] = imgaborfilt(I,gaborArray);

close all,
figure(5), imshow(I), set(5, 'name', 'Original image');
ax = gca;
figure(1), set(1, 'name', 'Even Gabor');
figure(2), set(2, 'name', 'Odd Gabor');
figure(3), set(3, 'name', 'Magnitude');
figure(4), set(4, 'name', 'Phase');
for p = 1:orientation_num
    figure(1),ax(end+1) = subplot(2,4,p);
    imshow(mag(:,:,p).*cos(phase(:,:,p)),[]);
    figure(2),ax(end+1) = subplot(2,4,p);
    imshow(mag(:,:,p).*sin(phase(:,:,p)),[]);
    figure(3),ax(end+1) = subplot(2,4,p);
    imshow(mag(:,:,p),[]);
    figure(4),ax(end+1) = subplot(2,4,p);
    imshow(phase(:,:,p),[]);
    for k = 1:4
        figure(k),subplot(2,4,p)
        theta = gaborArray(p).Orientation;
        lambda = gaborArray(p).Wavelength;
        title(sprintf('Orientation=%.1f, Wavelength=%d',theta,lambda));
    end
end
linkaxes(ax);
```