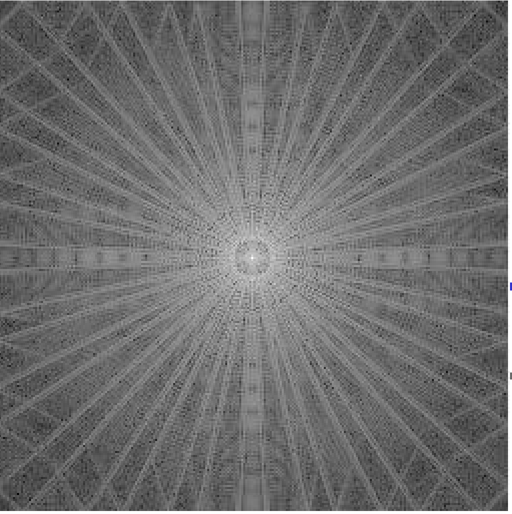
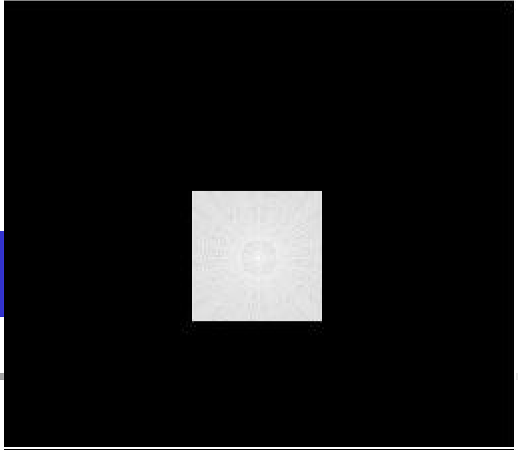
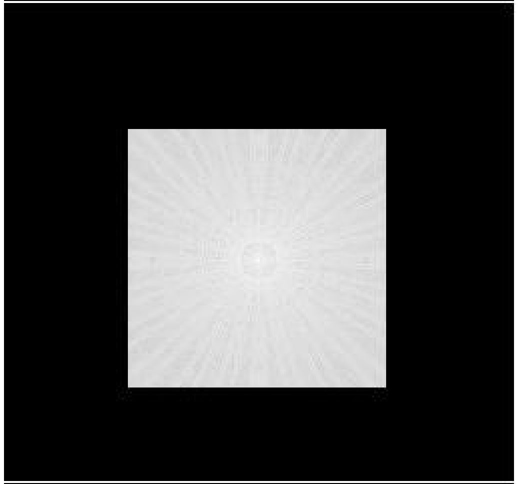
Digital Image Processing

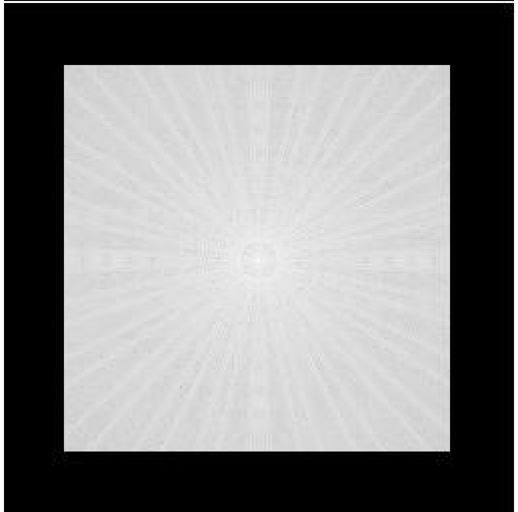
Frequency Domain Filtering

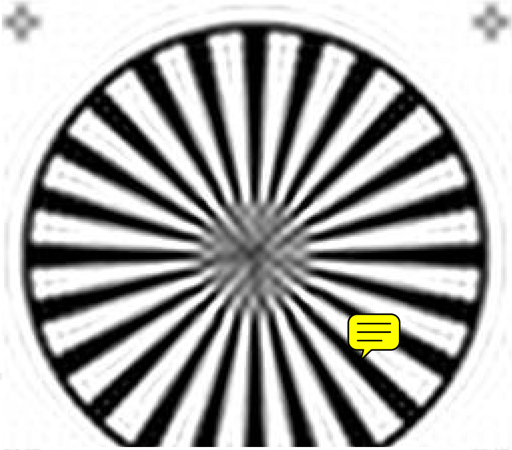


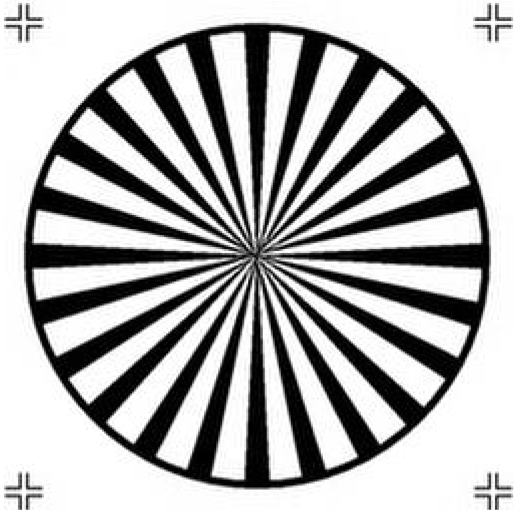
– Matl



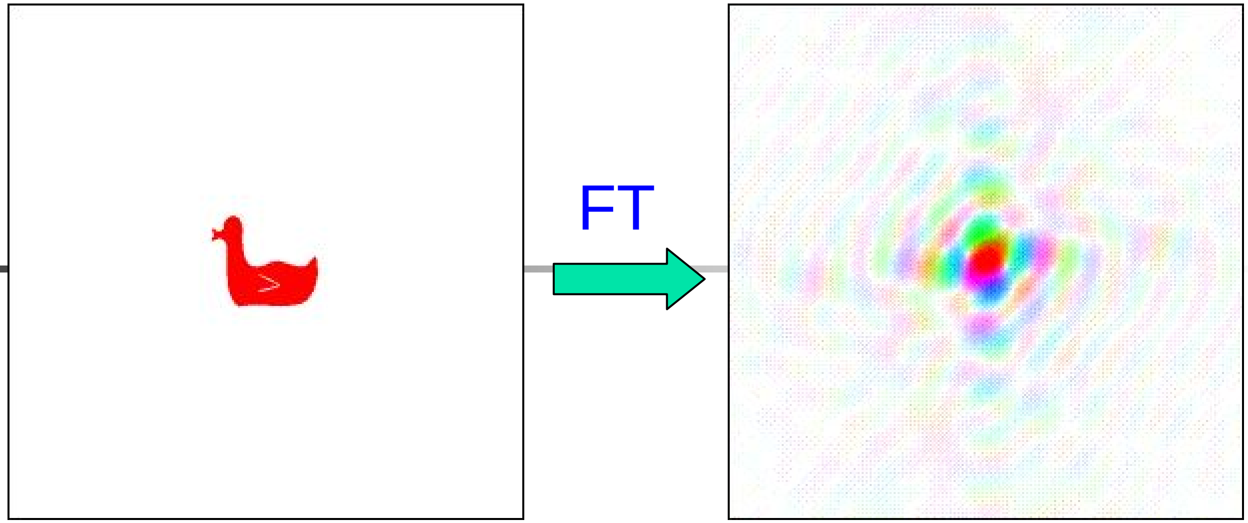




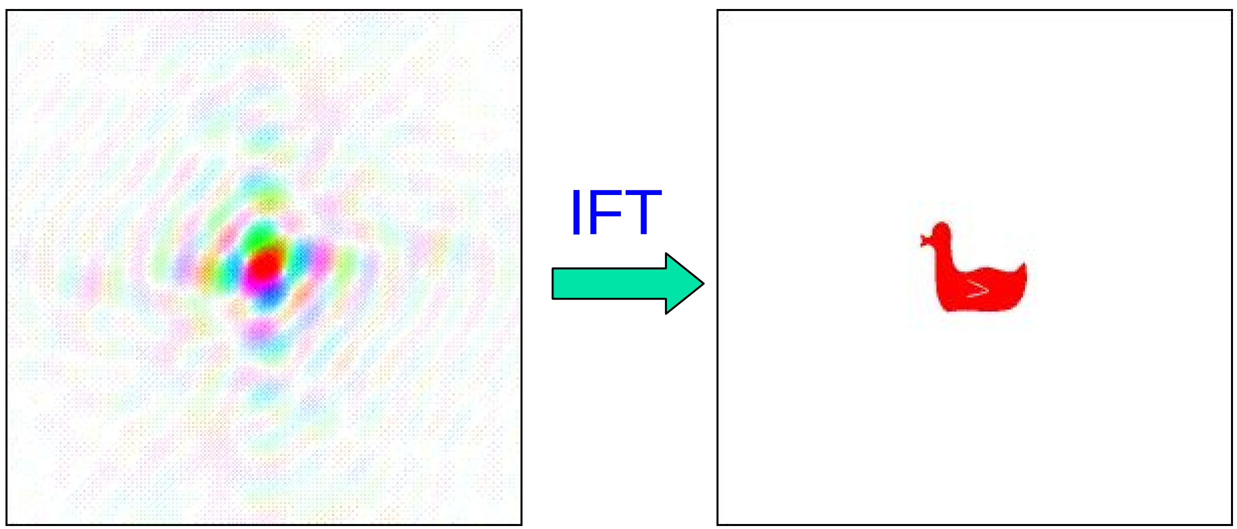




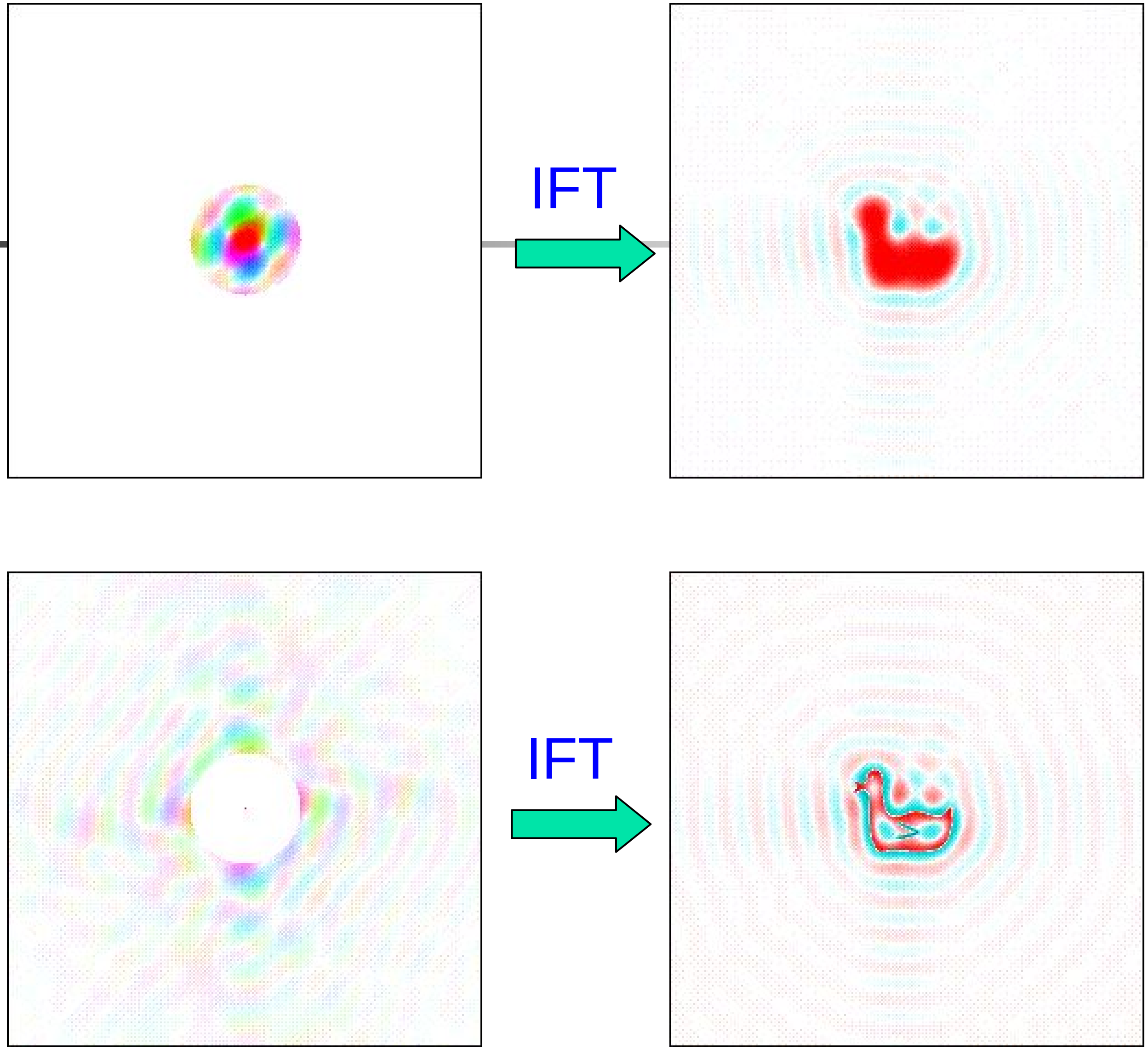
```
clear all; close all;
a=imread('testpat1.png');b=im2double(a);
figure;imshow(b);
Fb = fft2(b);Fbshift=fftshift(Fb);
figure;imshow(log(abs(Fbshift)+0.00000001),[]);

FMask=zeros(256,256);FMask([96:160],[96:160])=1.0;
Fbband=Fbshift.*FMask;
figure;imshow(log(abs(Fbband)+0.0000001),[]);
Fbband=ifftshift(Fbband);bandrebuilt=ifft2(Fbband);
figure;imshow(bandrebuilt);

FMask=zeros(256,256);FMask([64:192],[64:192])=1.0;
Fbband=Fbshift.*FMask;
figure;imshow(log(abs(Fbband)+0.0000001),[]);
Fbband=ifftshift(Fbband);bandrebuilt=ifft2(Fbband);
figure;imshow(bandrebuilt);

FMask=zeros(256,256);FMask([32:224],[32:224])=1.0;
Fbband=Fbshift.*FMask;
figure;imshow(log(abs(Fbband)+0.0000001),[]);
Fbband=ifftshift(Fbband);bandrebuilt=ifft2(Fbband);
figure;imshow(bandrebuilt);
```

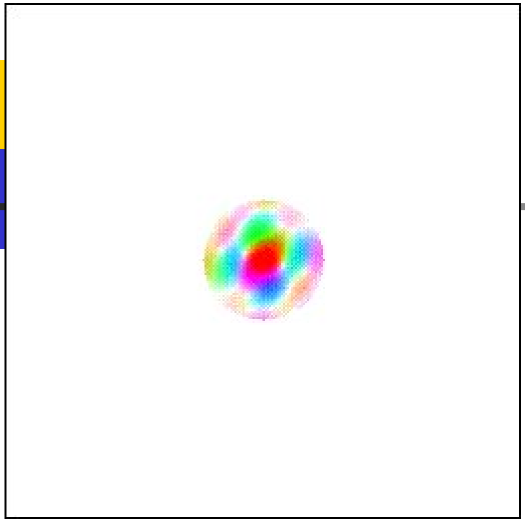


Signal decomposition(Coefficient computation)

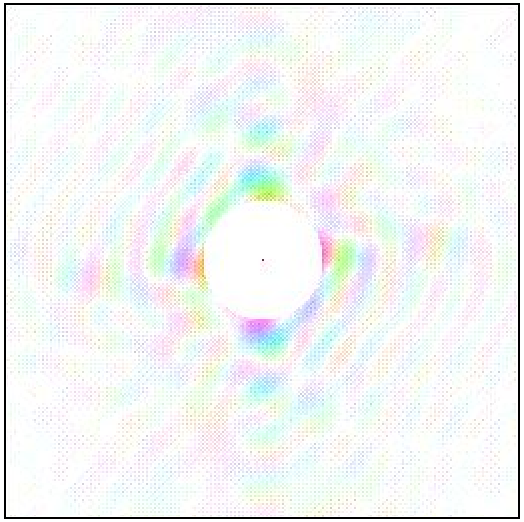


Signal reconstruction


IFT
IFT



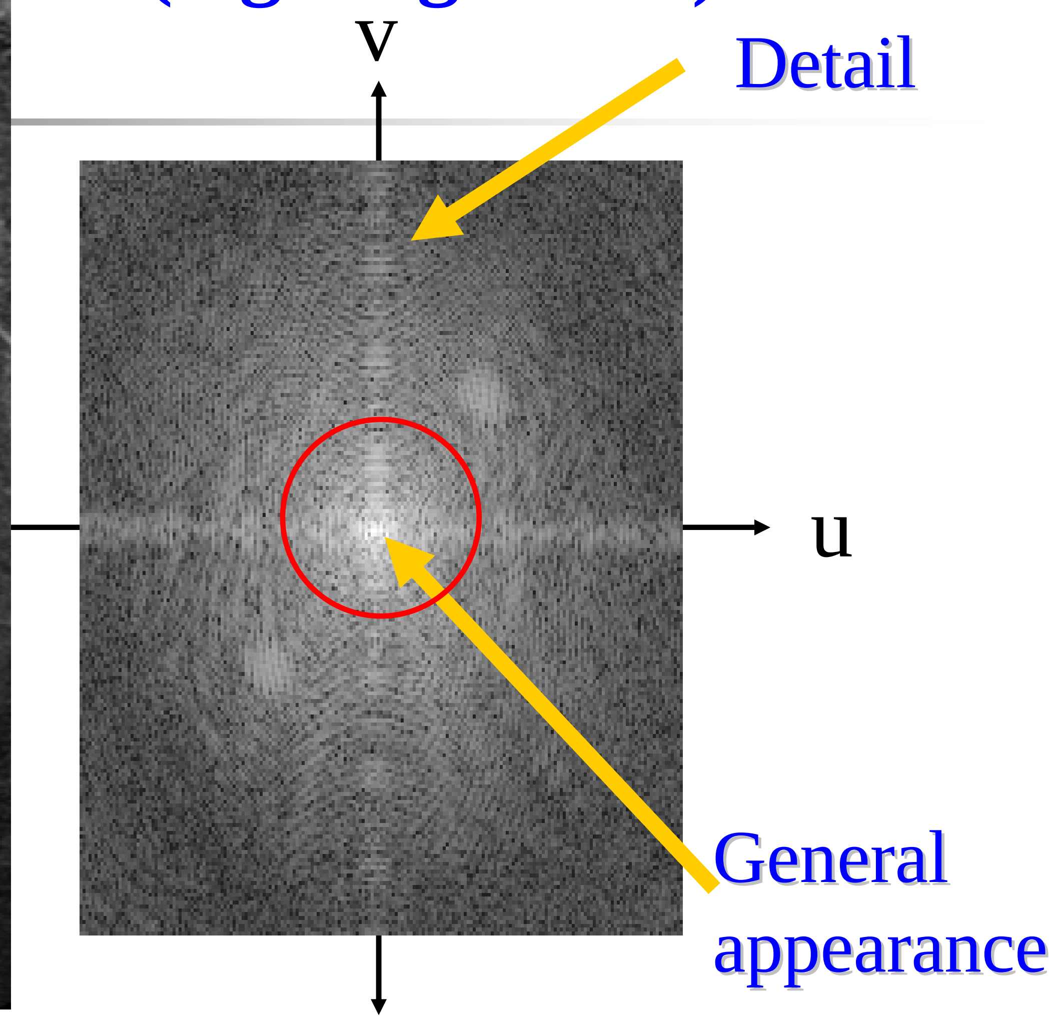
The central part of FT, i.e. the low frequency components are responsible for the general gray-level appearance of an image.



The high frequency components of FT are responsible for the detail information of an image.

Image

Frequency Domain (log magnitude)



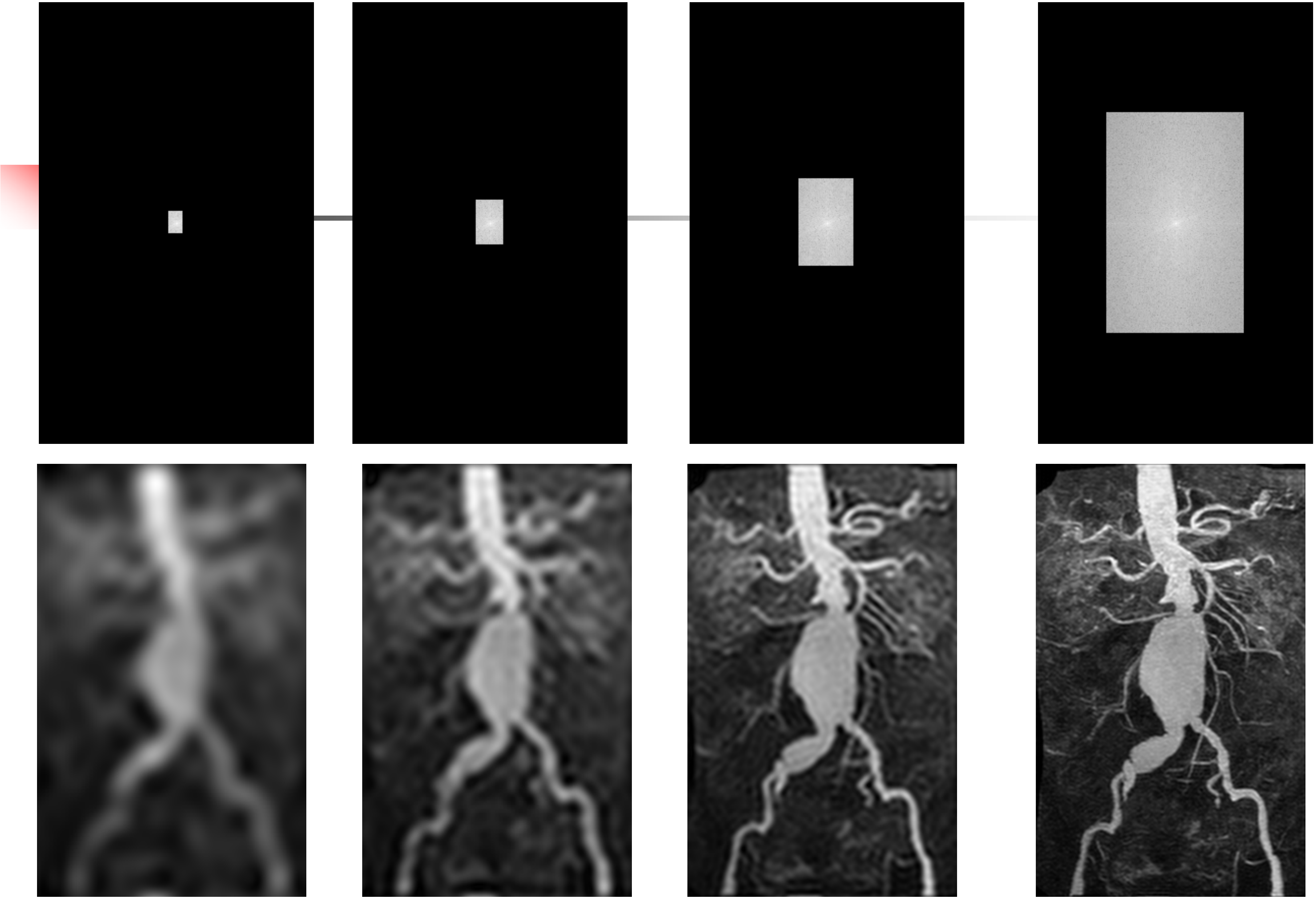


Image Filtering

- Image filtering techniques:
  - Spatial domain methods
  - Frequency domain methods
- Spatial (time) domain techniques are techniques that operate directly on pixels.
- Frequency domain techniques are based on modifying the Fourier transform of an image.

Frequency Domain Filtering

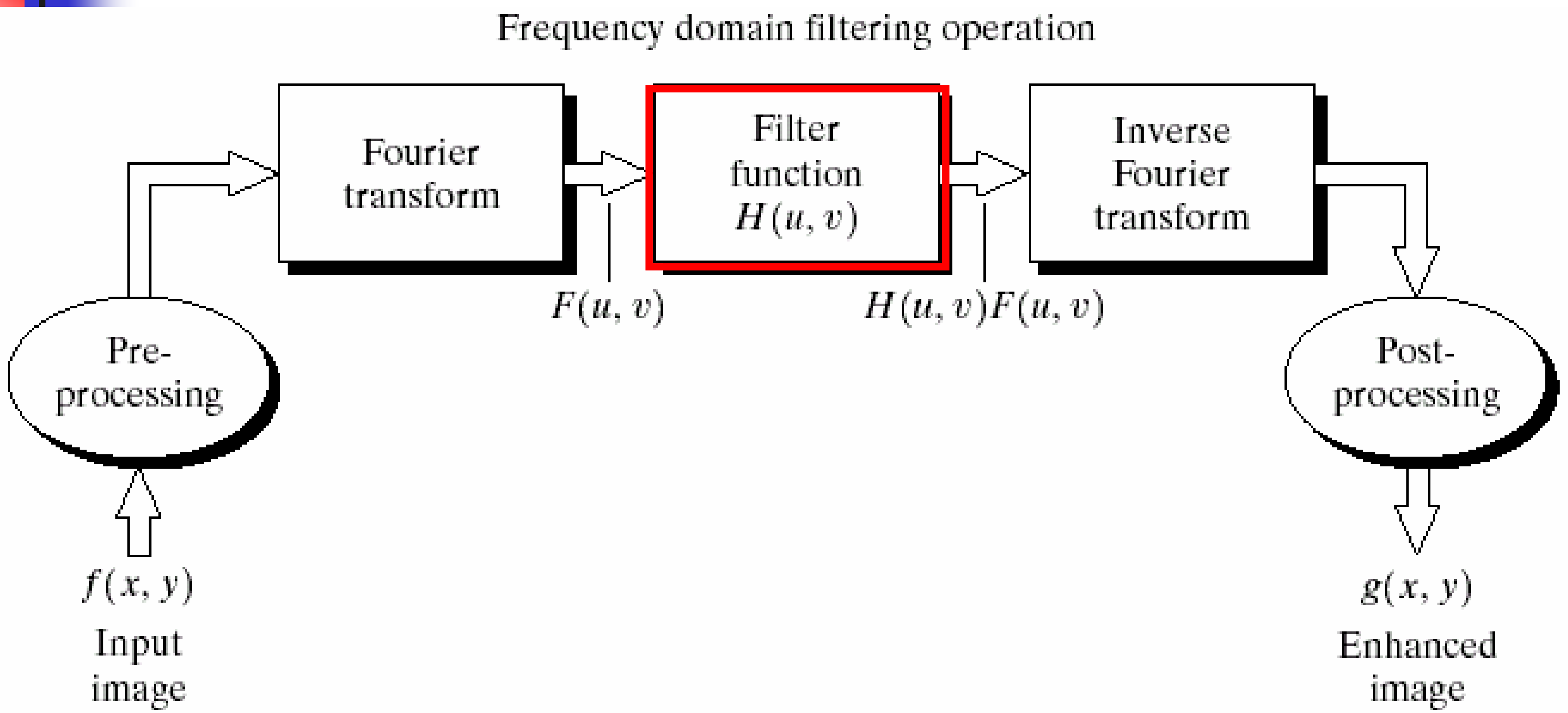


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Frequency Domain Filtering

- Edges and sharp transitions (e.g., noise) in an image contribute significantly to high-frequency content of FT.
- Low frequency contents in the FT are responsible to the general appearance of the image over smooth areas.
- Blurring (smoothing) is achieved by attenuating range of high frequency components of FT.

Convolution in Time Domain

$$g(x,y)=\sum_{x'=0}^{M-1}\sum_{y'=0}^{M-1}h(x',y')f(x-x',y-y')$$

$$\equiv f(x,y)*h(x,y)$$

- f(x,y) is the input image
- g(x,y) is the filtered image
- h(x,y): impulse response

Convolution Theorem

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

$$\Downarrow$$

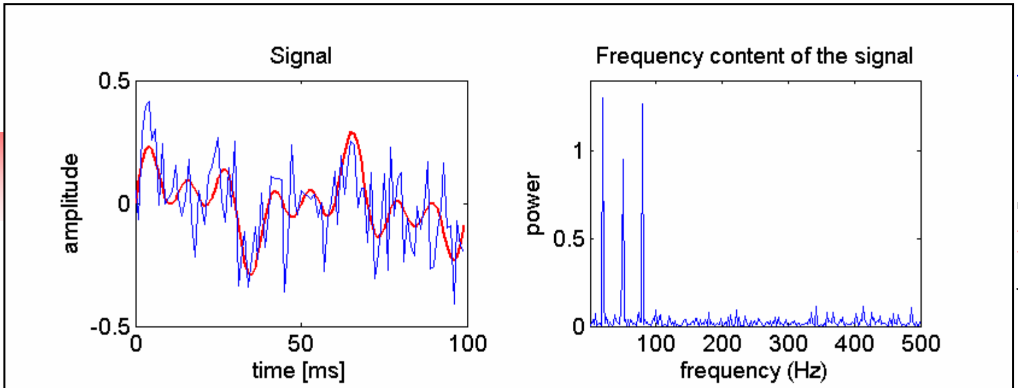
$$g(x,y)= f(x,y) \otimes h(x,y)$$

Multiplication in Frequency Domain

$\Updownarrow$

Convolution in Time Domain

- Filtering in Frequency Domain with H(u,v) is equivalent to filtering in Spatial Domain with h(x,y).



blue line = sum of 3 sinusoids (20, 50, and 80 Hz) + random noise

red line = sum of 3 sinusoids **without**t noise

Convolution Property of the Fourier Transform

Let functions $f(r,c)$ and $g(r,c)$ have Fourier Transforms $F(u,v)$ and $G(u,v)$.

$*$ = convolution
$\cdot$ = multiplication

Then,

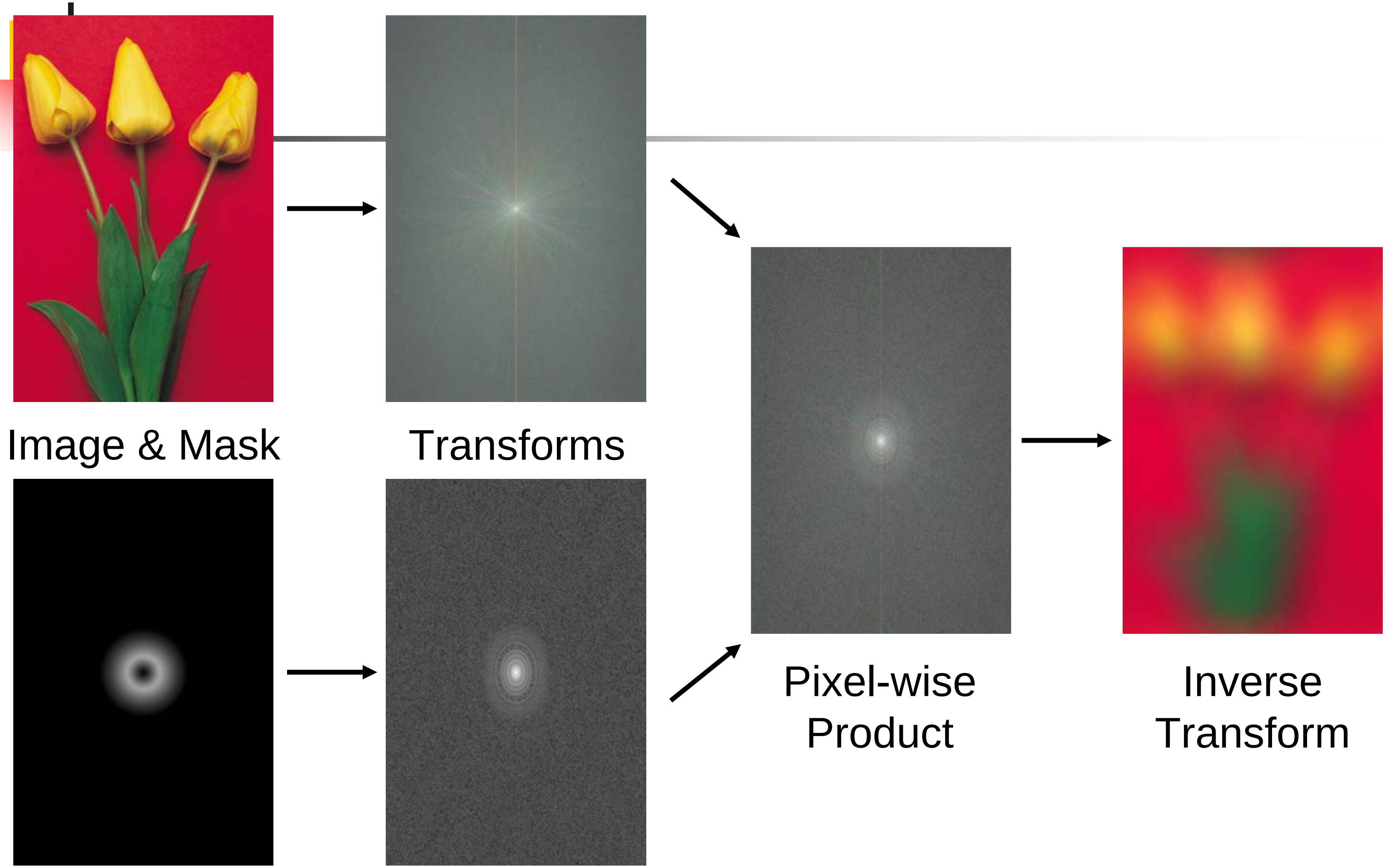
$$\mathbf{F}\{f * g\} = F \cdot G.$$

Moreover,

$$\mathbf{F}\{f \cdot g\} = F * G.$$

The Fourier Transform of a convolution equals the product of the Fourier Transforms. Similarly, the Fourier Transform of a multiplication is the convolution of the Fourier Transforms

Convolution via Fourier Transform



How to Convolve via FT in Matlab

1. Read the image from a file into a variable, say **`I`**.
2. Read in or create the convolution mask, **`h`**.
3. Compute the sum of the mask: **`s = sum(sum(h));`**
4. If **`s == 0,`** set **`s = 1;`**
5. Create: **`H = zeros(size(I));`**
6. Copy **`h`** into the middle of **`H`**.
7. Shift **`H`** into position: **`H = ifftshift(H);`**
8. Take the 2D FT of **`I`** and **`H`**: **`FI=fft2(I); FH=fft2(H);`**
9. Pointwise multiply the FTs: **`FJ=FI.*FH;`**
10. Compute the inverse FT: **`J = real(ifft2(FJ));`**
11. Normalize the result: **`J = J/s;`**

The mask is usually 1-band

For color images you may need to do each step for each band separately.

Coordinate Origin of the FFT

Center = $(\text{floor}(R/2)+1, \text{floor}(C/2)+1)$

Even

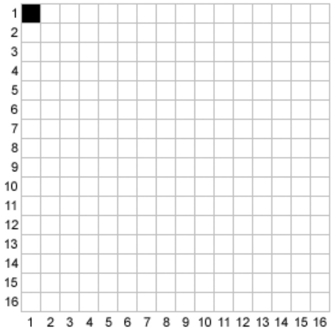
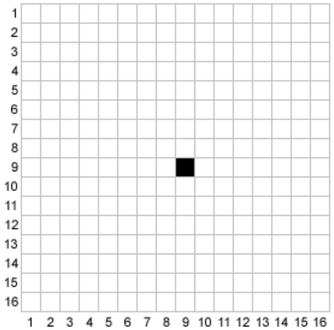


Image Origin



After FFT shift

Odd

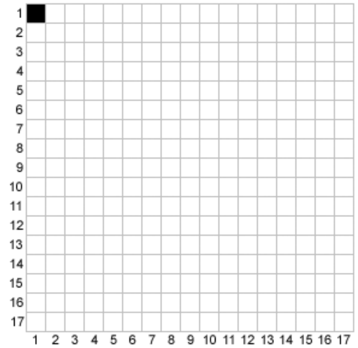
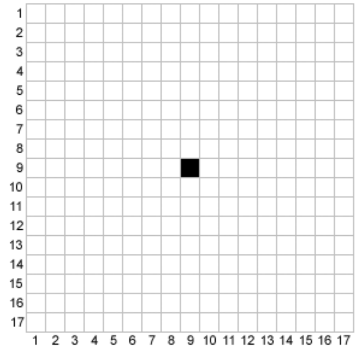


Image Origin

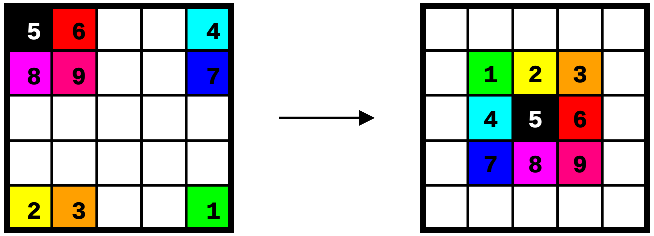


After FFT shift

Matlab's `fftshift` and `ifftshift`

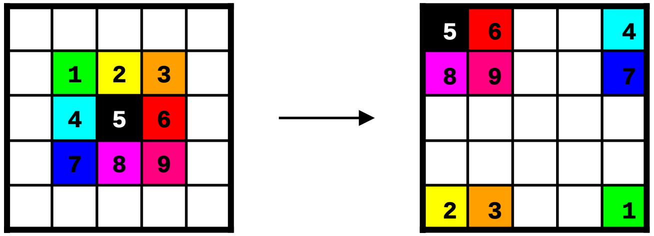
`J = fftshift(I):`

$$I\,(1,1) \rightarrow J\,(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1)$$



`I = ifftshift(J):`

$$J\,(\lfloor R/2 \rfloor + 1, \lfloor C/2 \rfloor + 1) \rightarrow I\,(1,1)$$



where $\lfloor x \rfloor$ `= floor(x)` = the largest integer smaller than x.

Algorithm Complexity

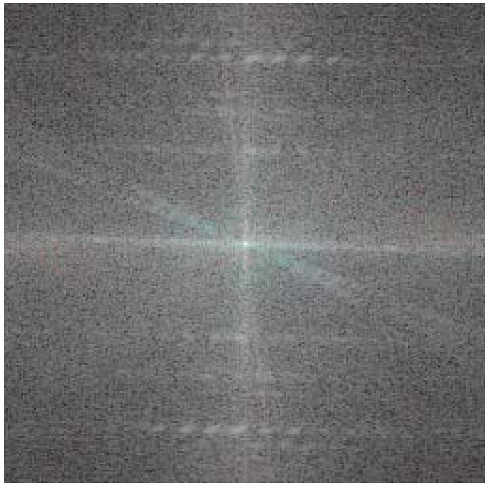
- We can compute the DFT directly using the formula
  - An N point DFT would require N^2 floating point multiplications per output point
  - Since there are N^2 output points , the computational complexity of the DFT is N^4
  - $N^4=4x10^9$ for N=256
  - Bad news! Many hours on a workstation

Algorithm Complexity

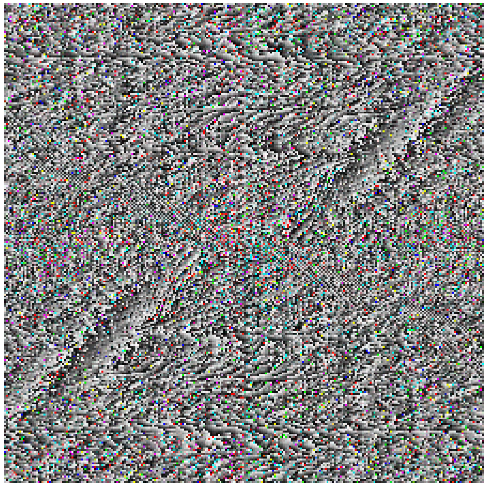
$$F(u,v) = F^*(-u,-v)$$

- The FFT algorithm was developed in the 60's for seismic exploration
- Reduced the DFT complexity to $2N^2\log_2 N$
  - $2N^2\log_2 N \sim 10^6$ for $N=256$
  - A few seconds on a workstation

original image



Fourier log magnitude

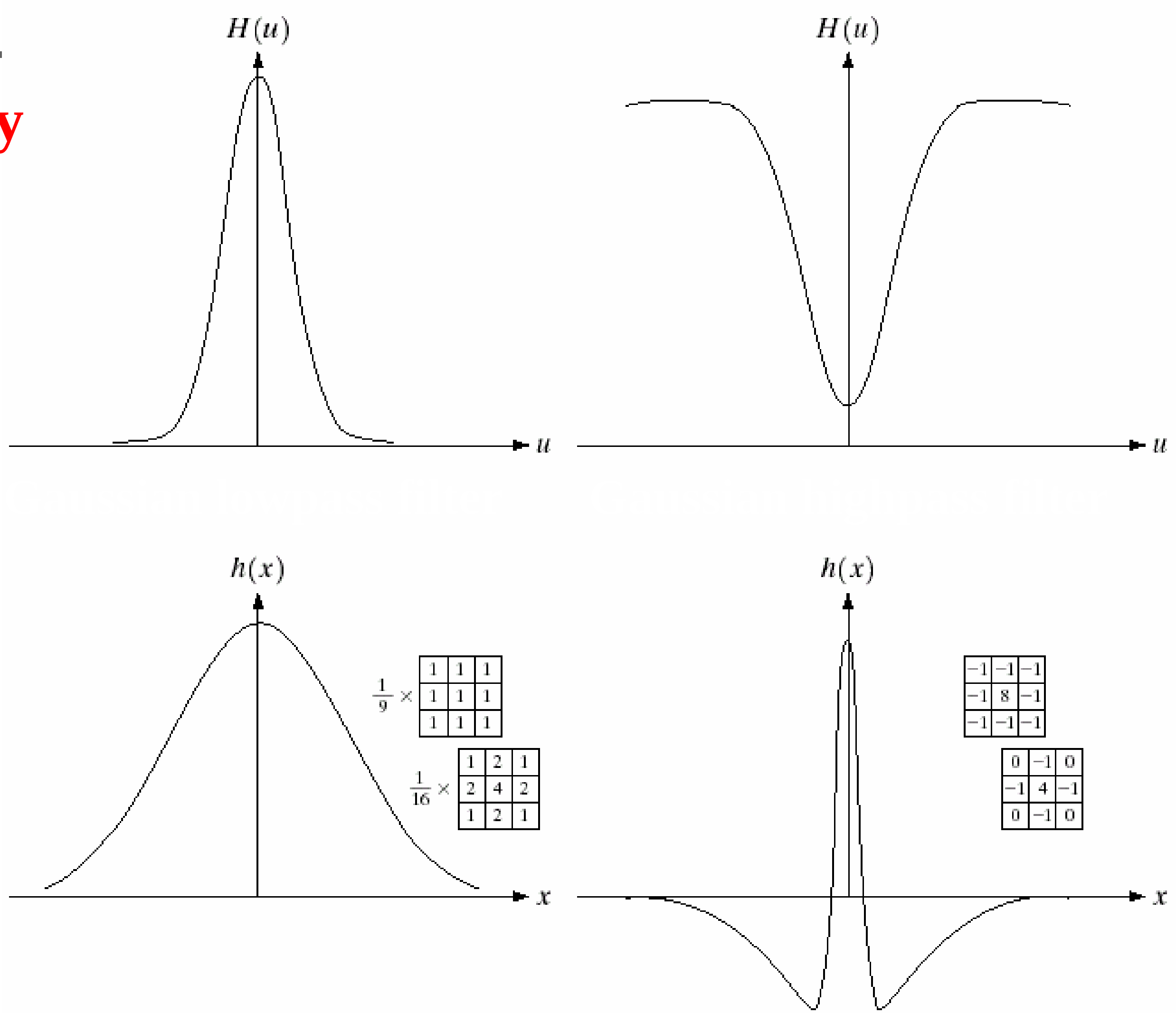


Fourier phase

Examples of Filters

Frequency

Spatial



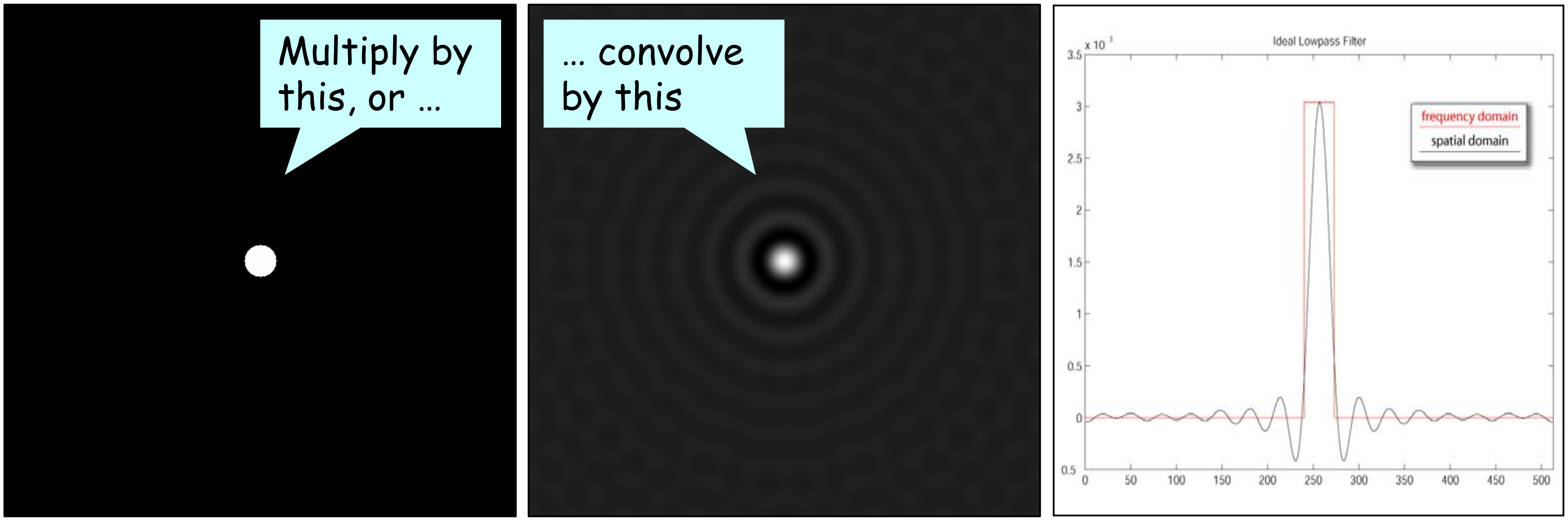
Blurring: Averaging / Lowpass Filtering

Blurring results from:

- Pixel averaging in the spatial domain:
  - Each pixel in the output is a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 1.
- Lowpass filtering in the frequency domain:
  - High frequencies are diminished or eliminated
  - Individual frequency components are multiplied by a nonincreasing function of ω such as $1/\omega = 1/\sqrt{(u^2+v^2)}$.

Ideal Lowpass Filter
理想低通濾波器

Image size: 512x512
FD filter radius: 16



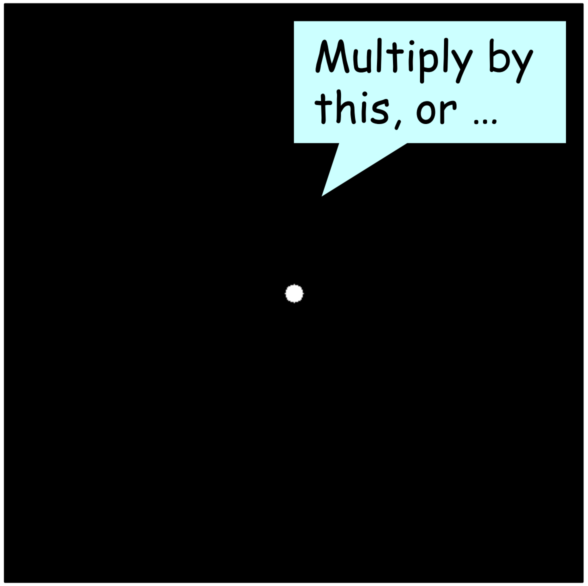
Fourier Domain Rep.

Spatial Representation

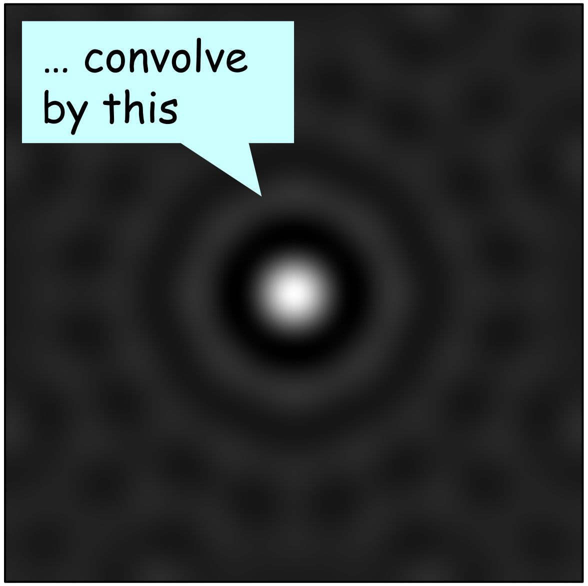
Central Profile

Ideal Lowpass Filter

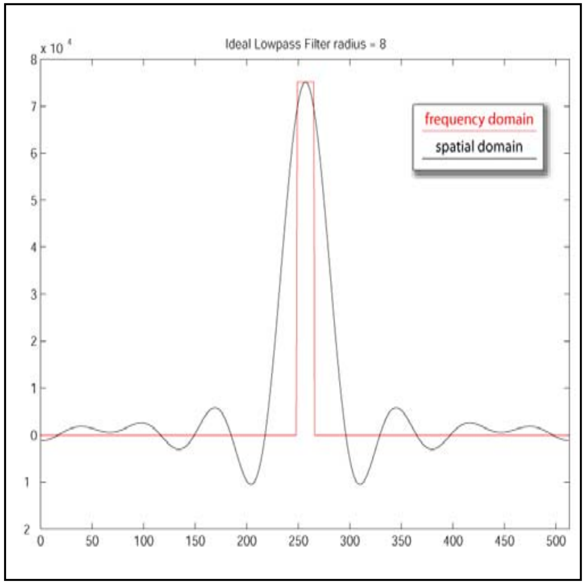
Image size: 512x512
FD filter radius: 8



Fourier Domain Rep.



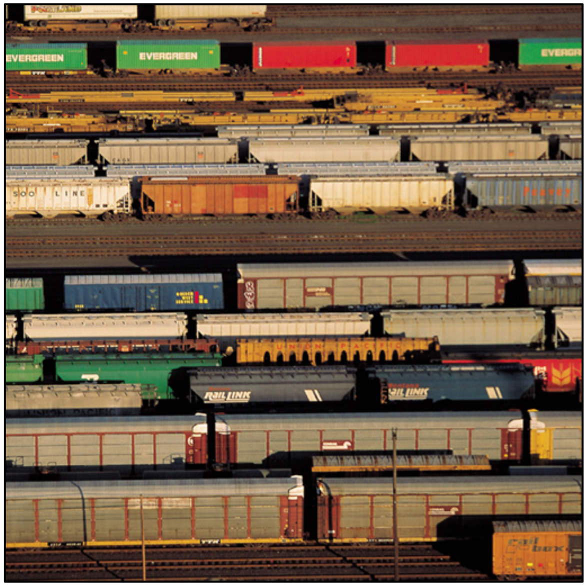
Spatial Representation



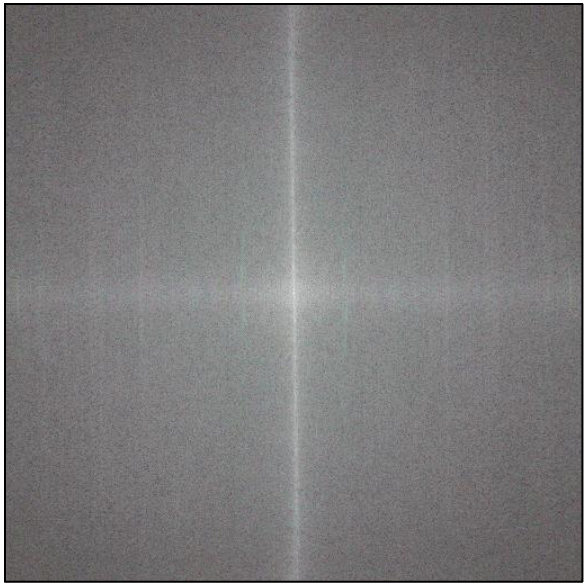
Central Profile

Power Spectrum and Phase of an Image

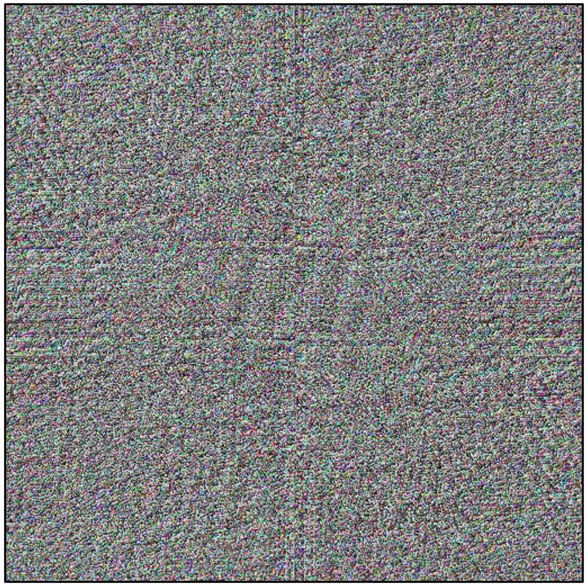
Consider the image below:



Original Image



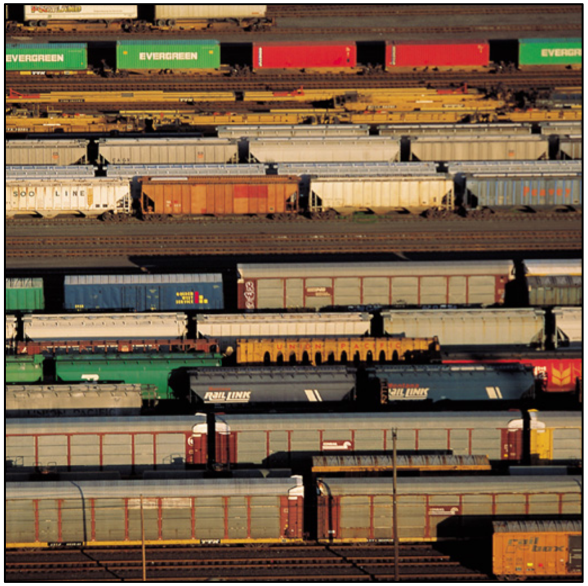
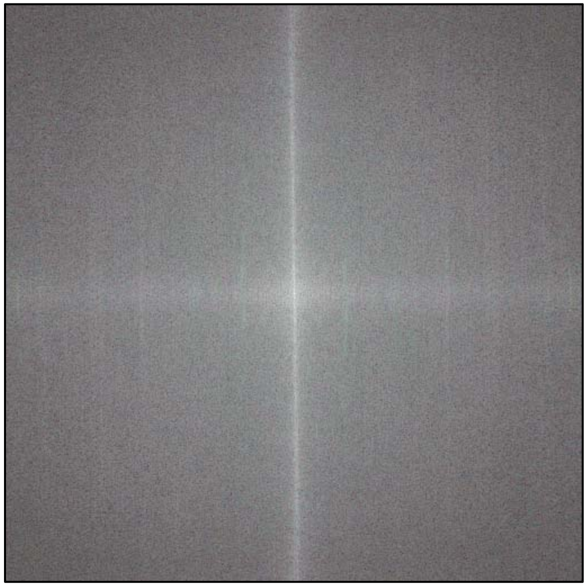
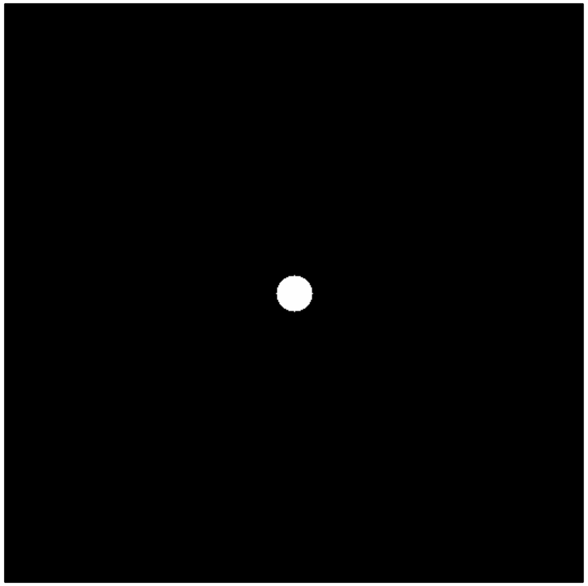
Power Spectrum



Phase

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 16

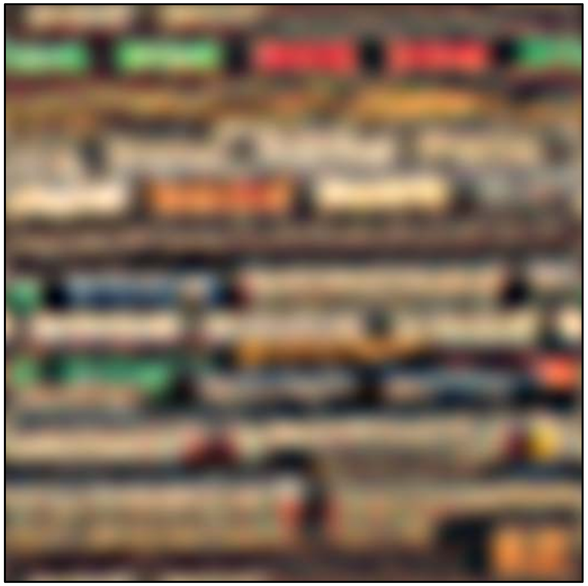
Original Image

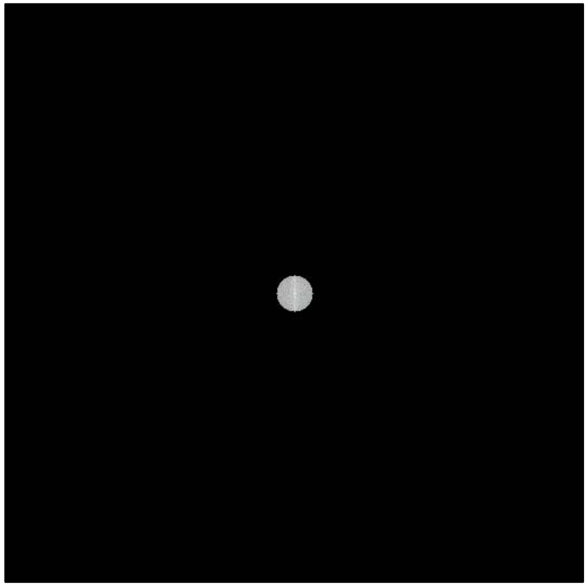
Power Spectrum

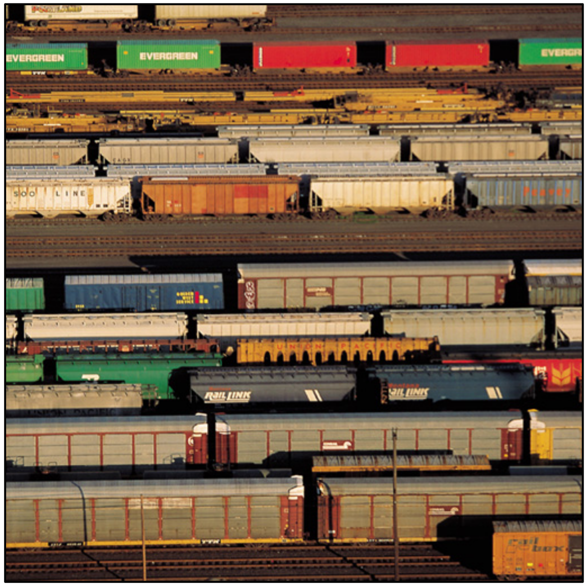
Ideal LPF in FD

Ideal Lowpass Filter

Image size: 512x512
FD filter radius: 16





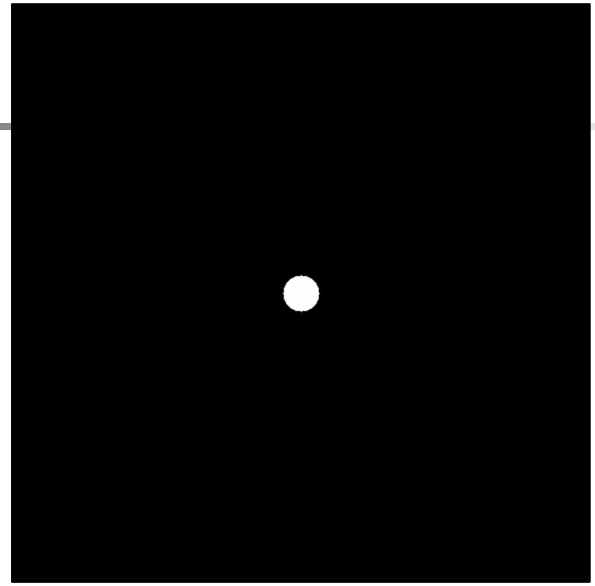


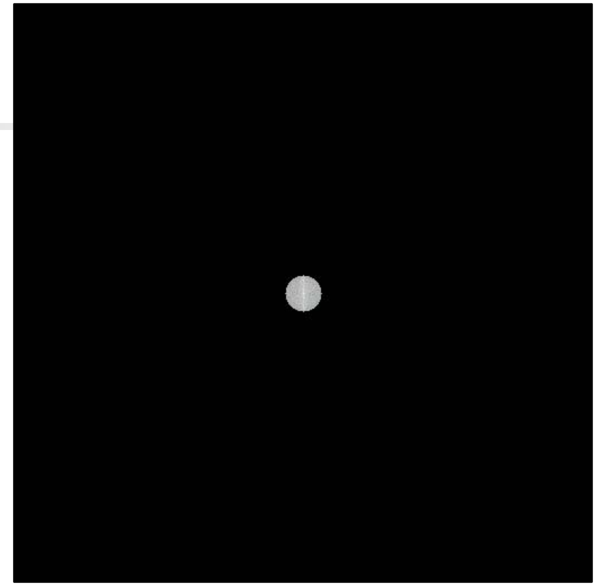
Filtered Image

Filtered Power Spectrum

Original Image

Ideal low-pass filter (ILPF)



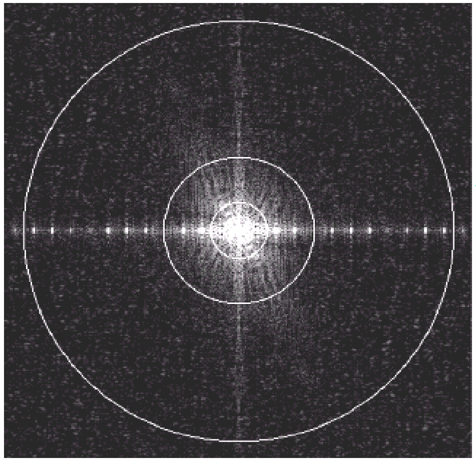


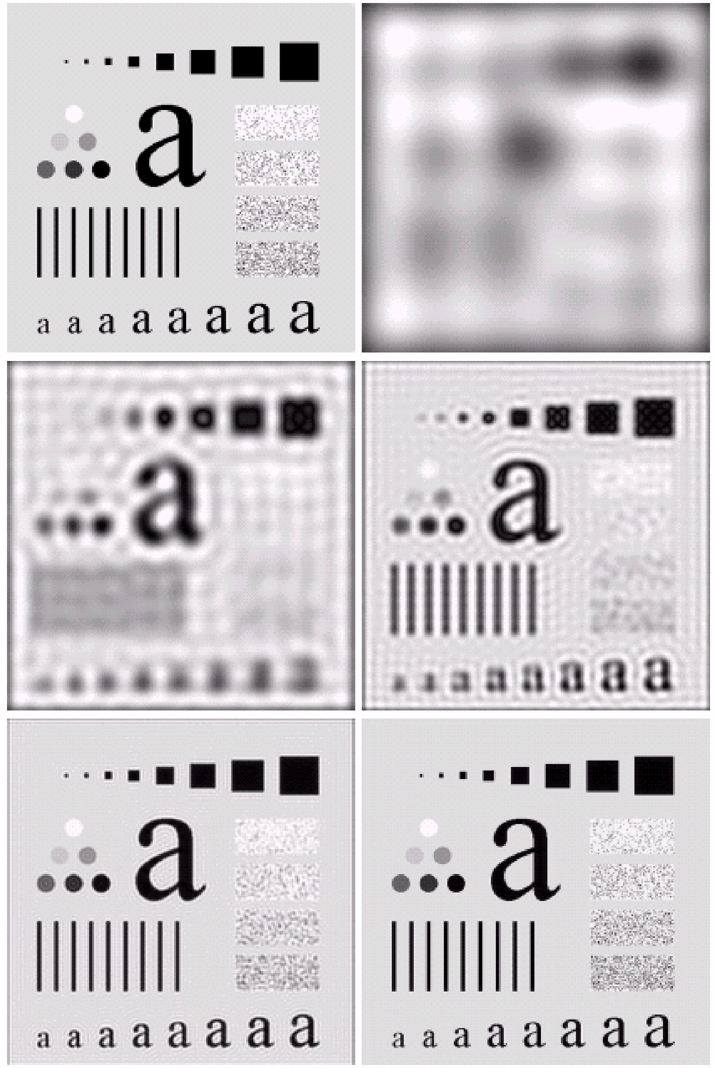
$$H(u,v)=\begin{cases}1 & D(u,v)\le D_0\\ 0 & D(u,v)>D_0\end{cases}$$

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

(M/2,N/2): center in frequency domain.

D_0 is called the ***cutoff*** frequency(截止频率).





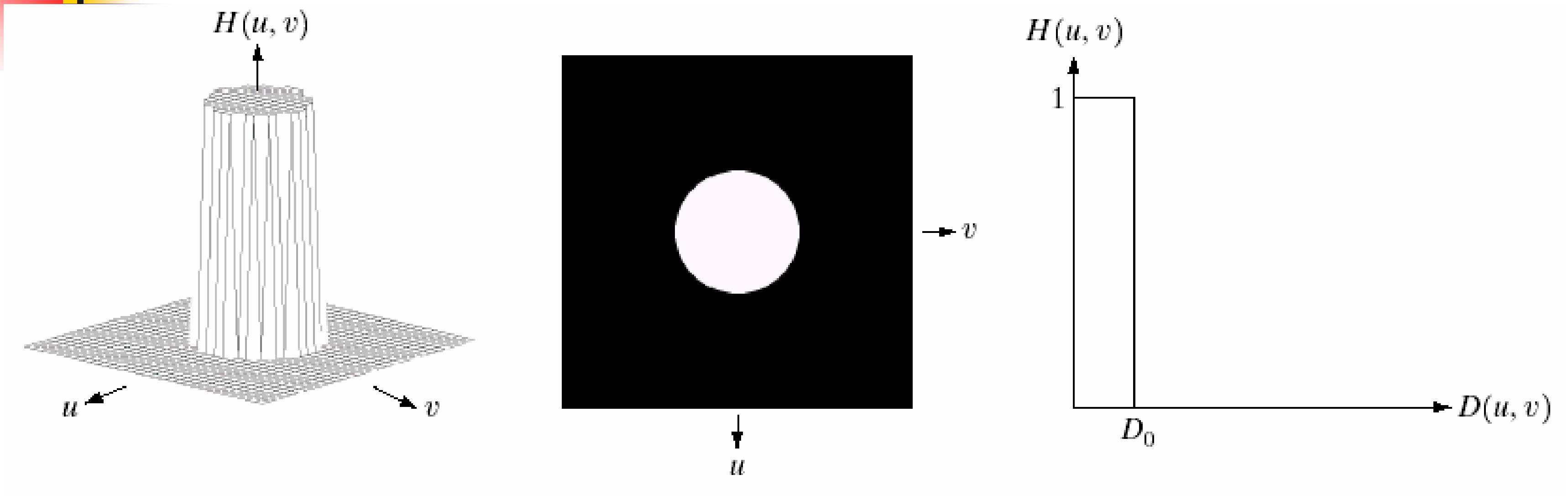
ringing and blurring

Ideal in frequency domain means non-ideal in spatial domain, vice versa.

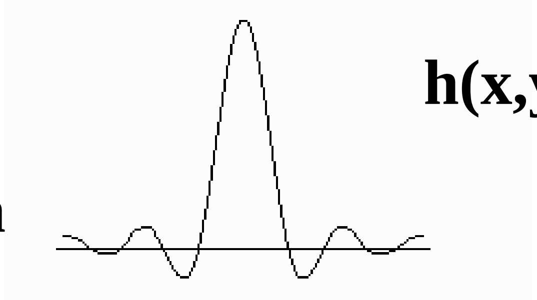
a	b
c	d
e	f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Shape of ILPF



Frequency domain



h(x,y)

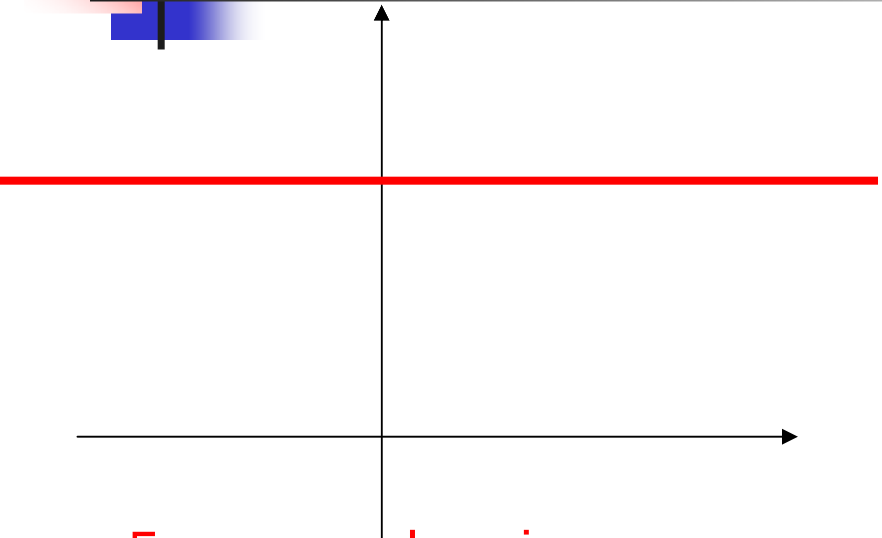
Spatial domain

Fourier transform basis functions

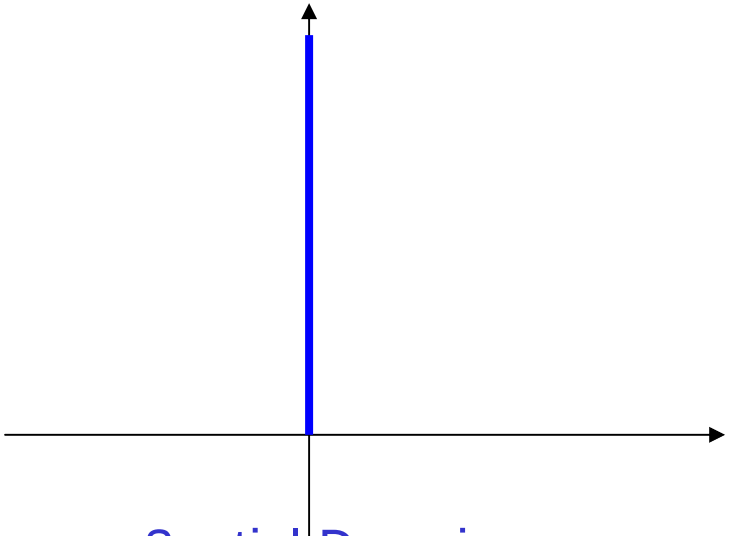
Approximating a square wave as the sum of sine waves.

$$F(u,v) \cdot H(u,v) \leftrightarrow f(x,y) \otimes h(x,y)$$

The limiting case…



Frequency domain



Spatial Domain

$H(u,v)=1$

$h(x,y)=$

0	0	0
0	1	0
0	0	0

And vice versa.

Butterworth Lowpass Filters (BLPF)

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

- Smooth transfer function, no sharp discontinuity, no clear cutoff frequency.

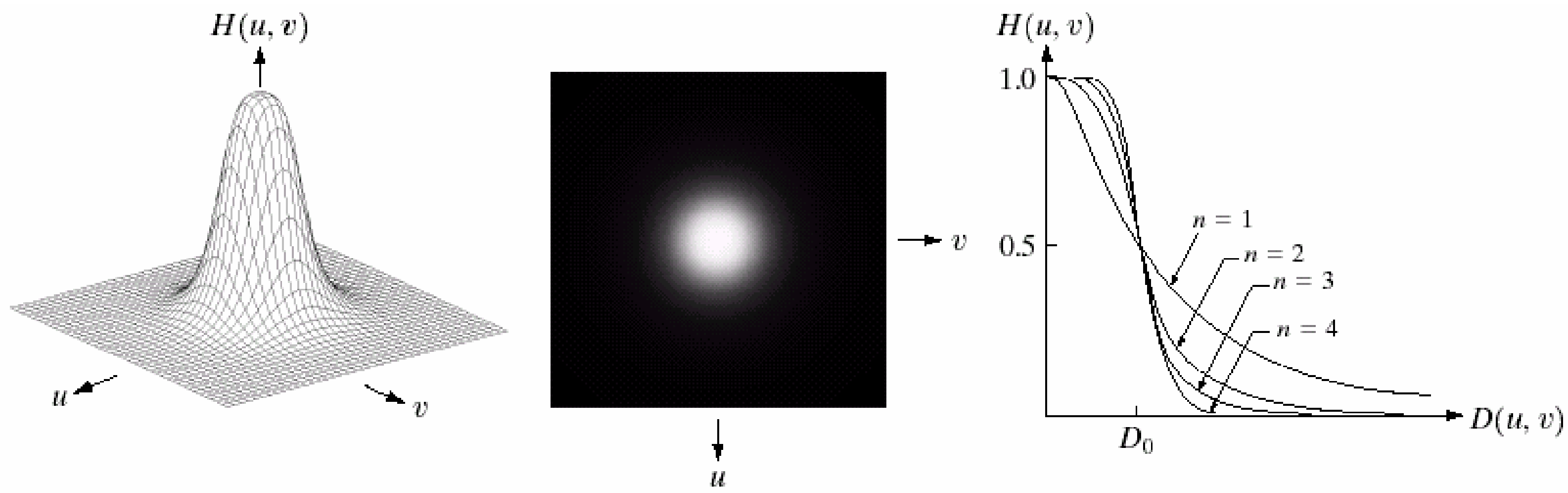
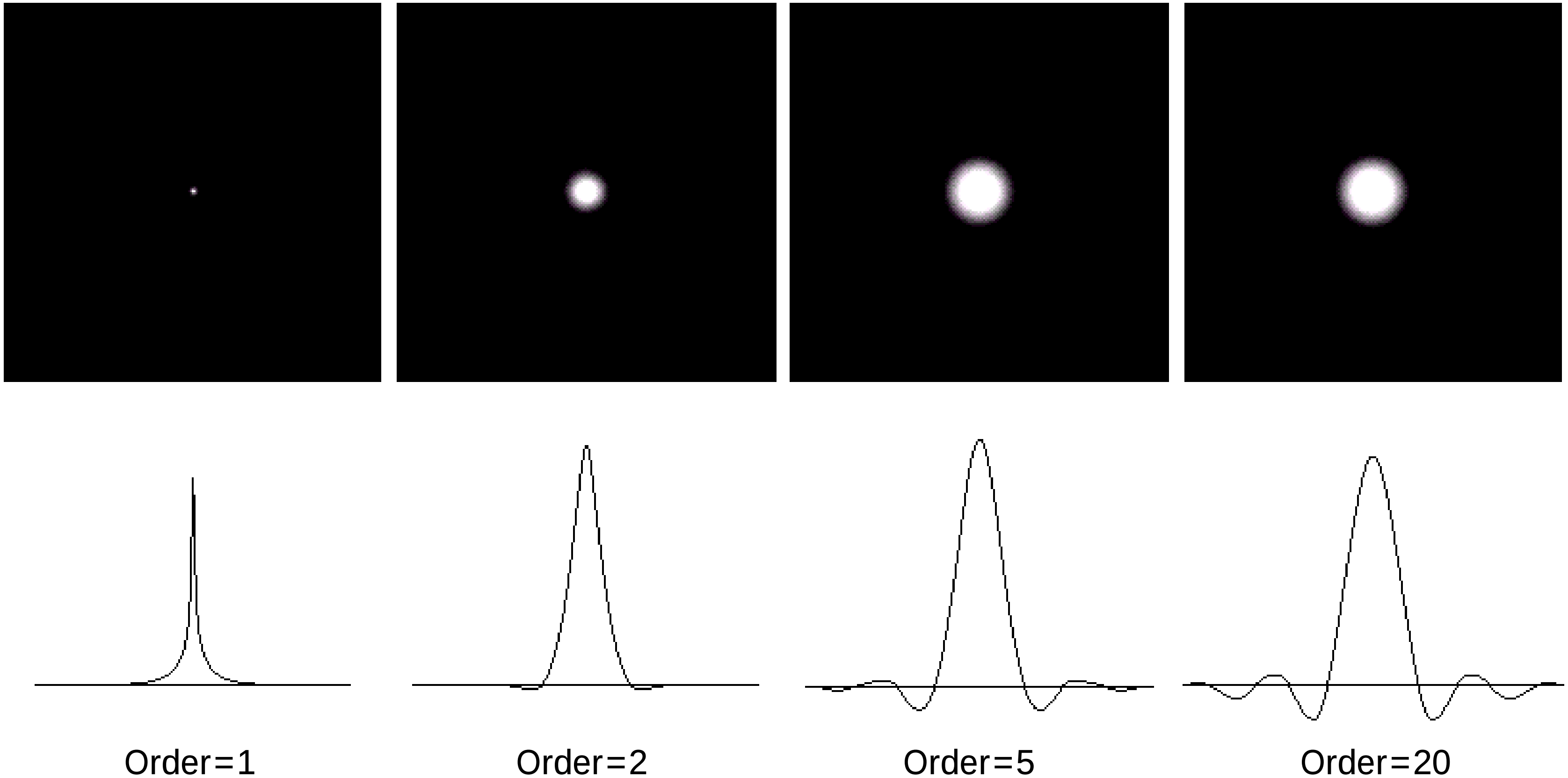


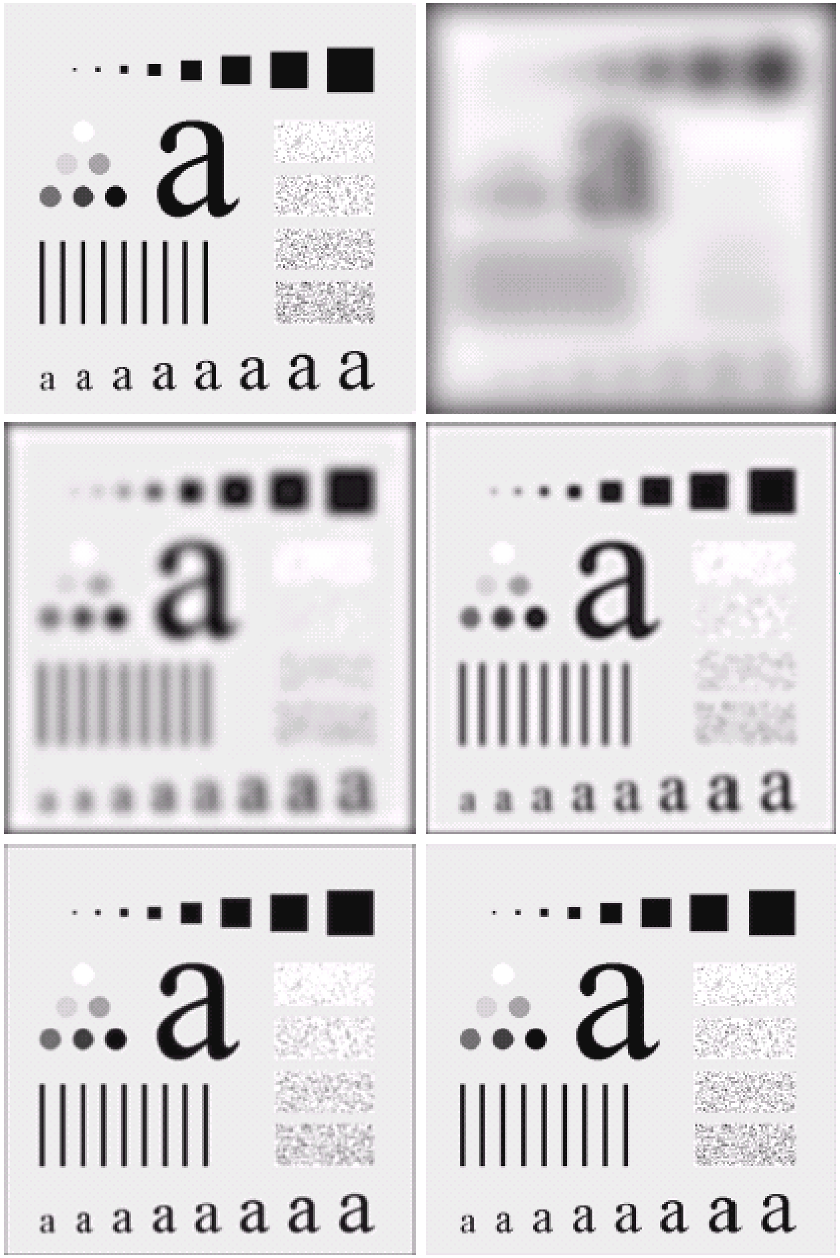
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth Lowpass Filters (BLPF)



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

- Spatial representations of BLPF with different orders
- Notice the ringing of different orders



No serious ringing artifacts

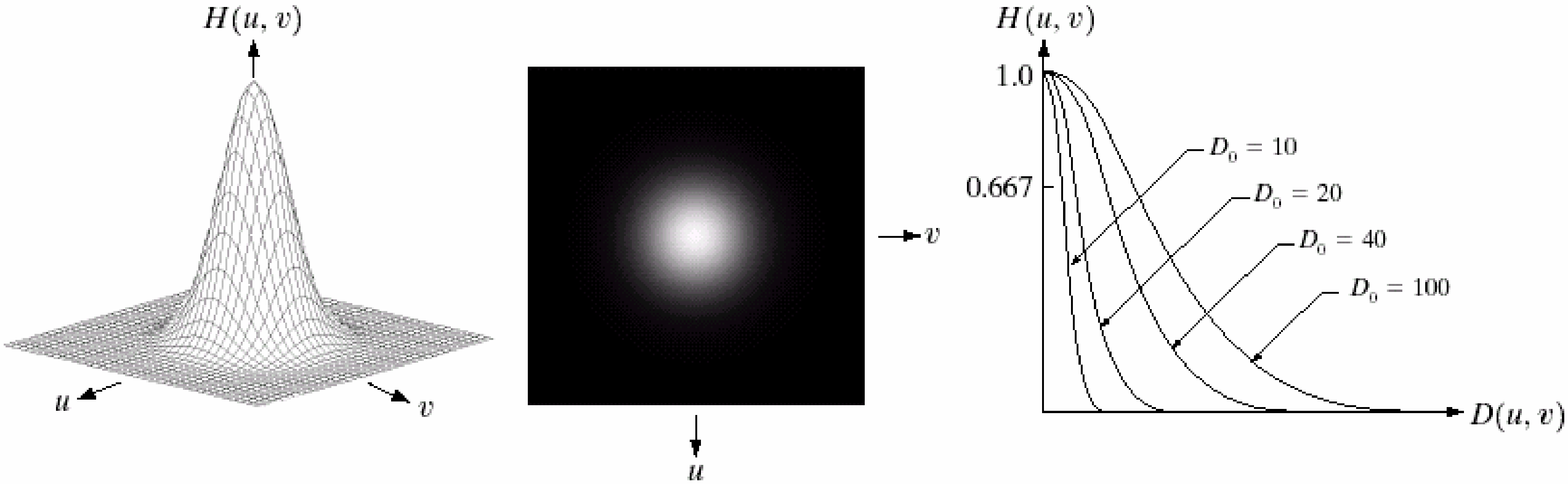
a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Gaussian Lowpass Filters (GLPF)

- Smooth transfer function, smooth impulse response, no ringing

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

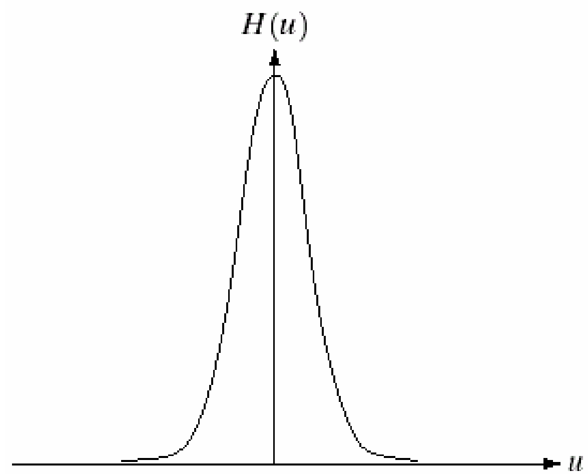


a b c

FIGURE 4.17 (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.

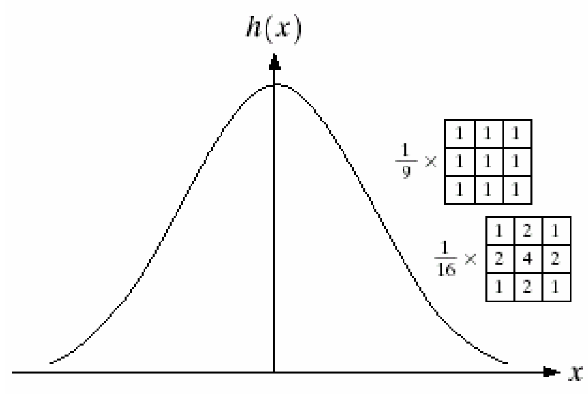
GLPF

Frequency domain



Gaussian lowpass filter

Spatial domain



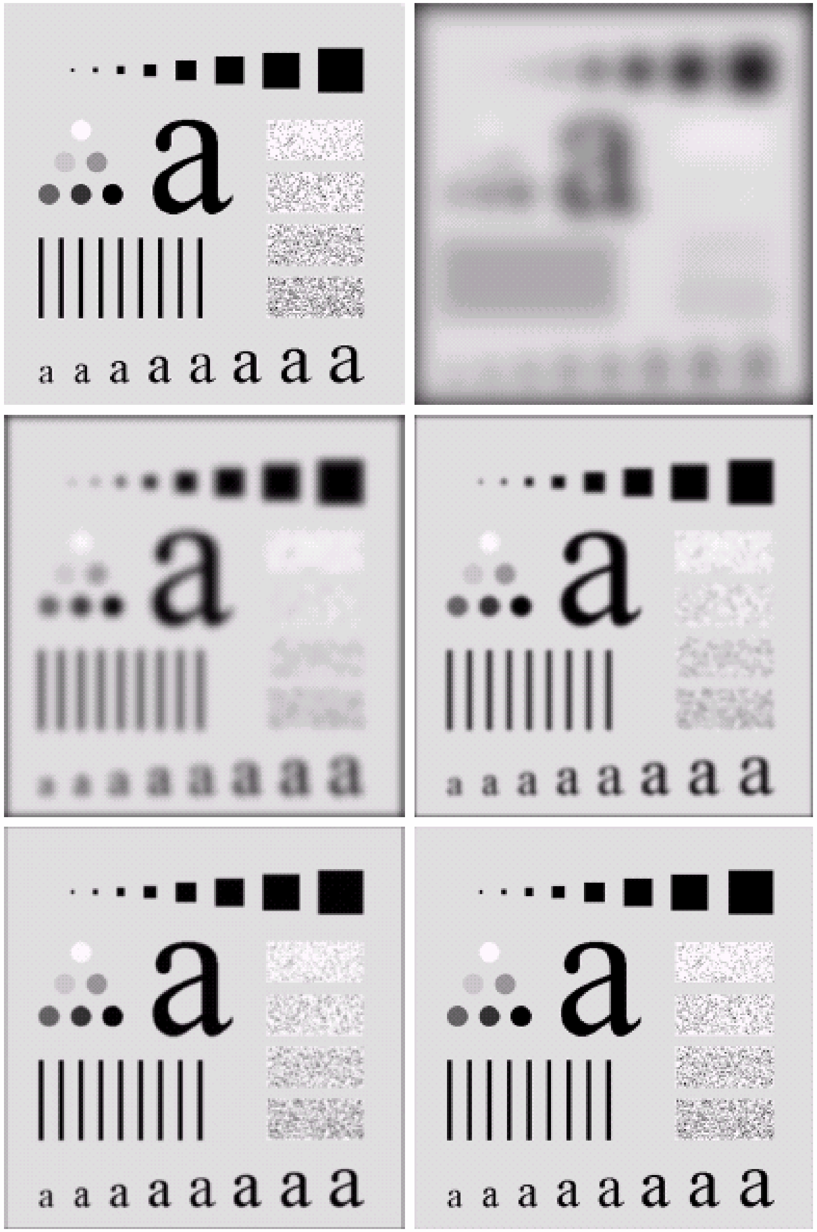


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

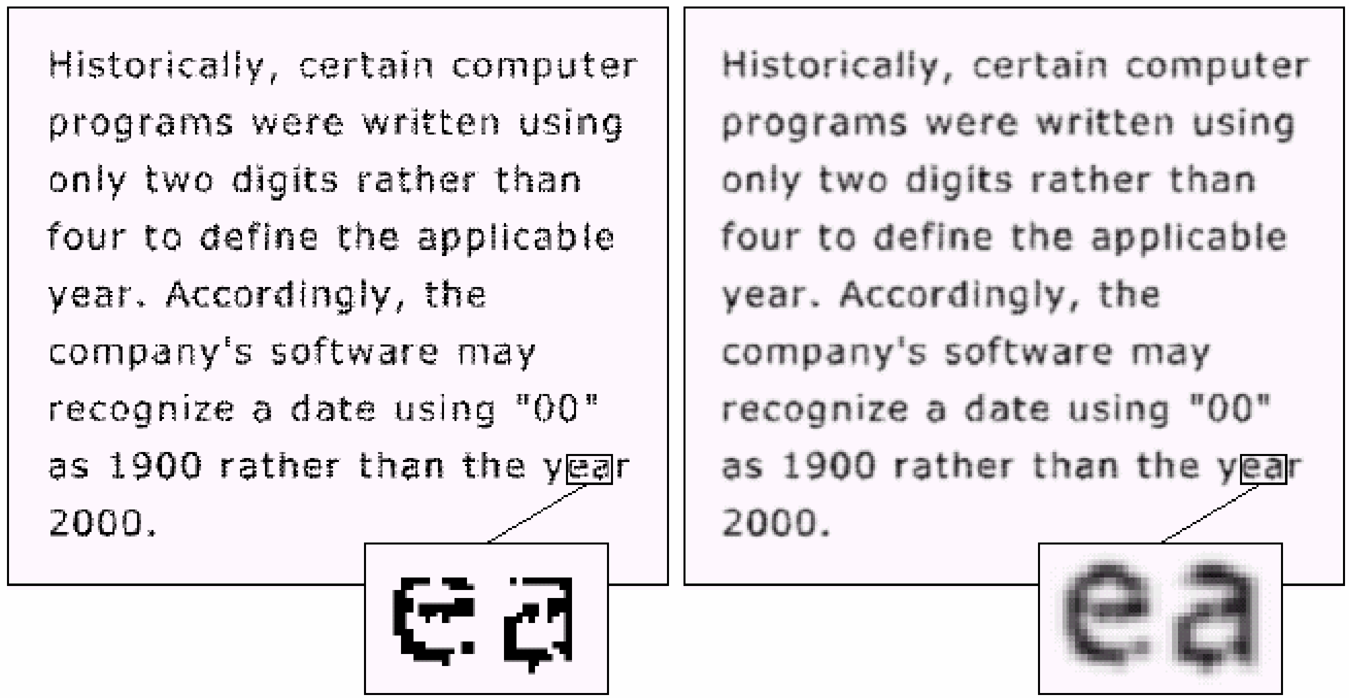
a b
c d
e f

No ringing artifacts

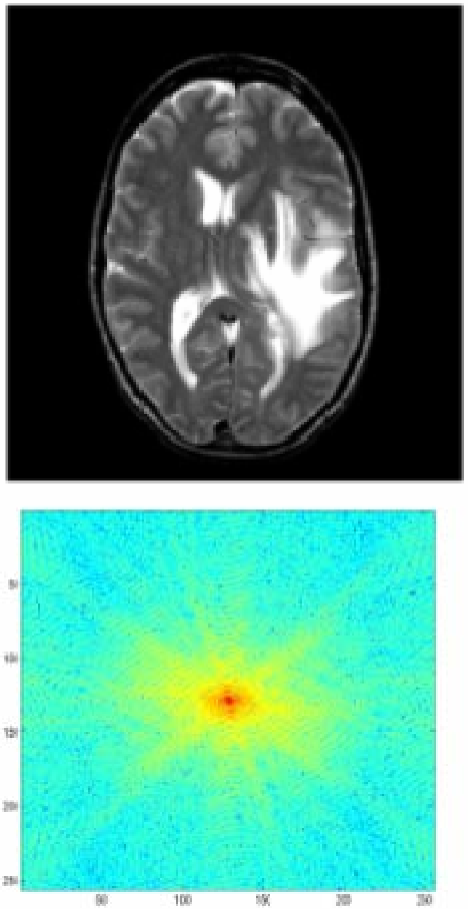
Examples of Lowpass Filtering

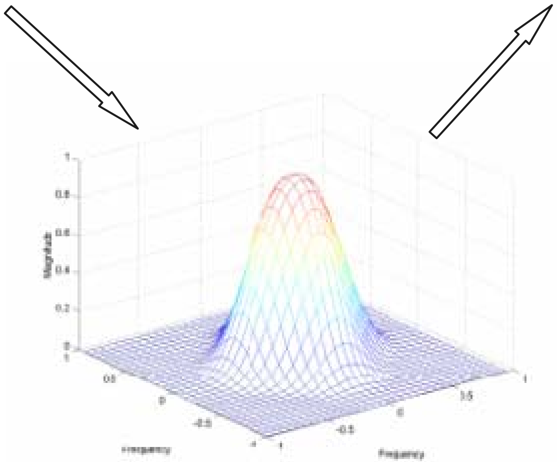
a b

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).

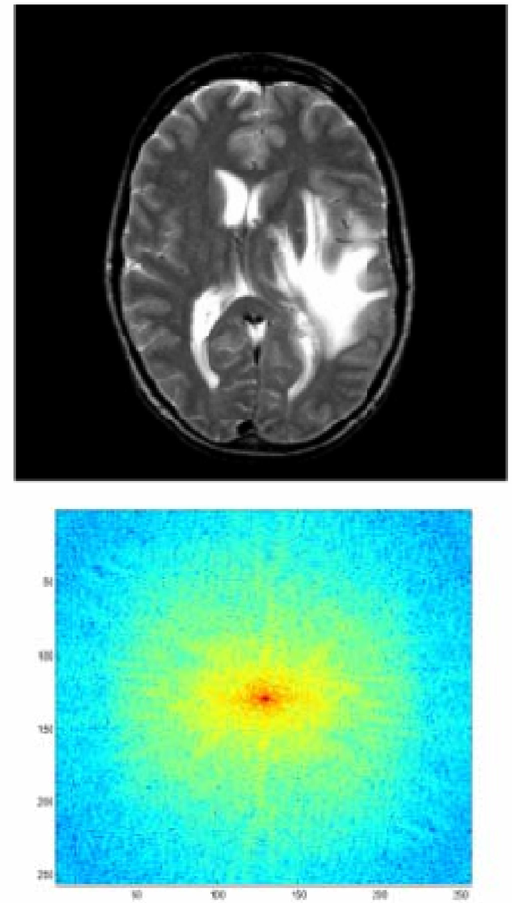


Examples of Lowpass Filtering





Low-pass filter H(u,v)



Original image and its FT

Filtered image and its FT

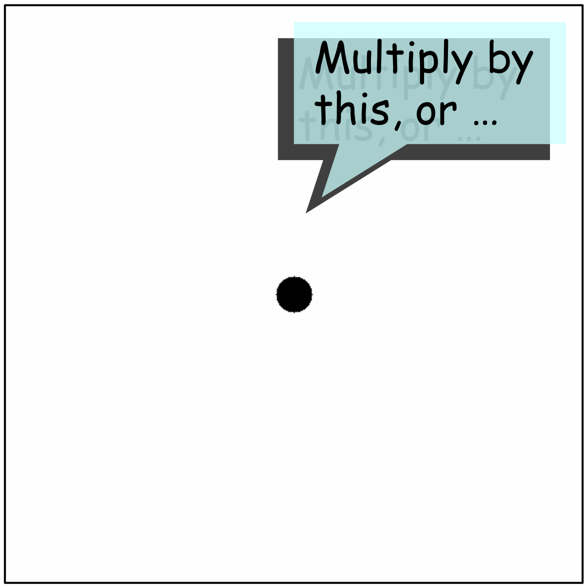
Sharpening: Differencing / Highpass Filtering

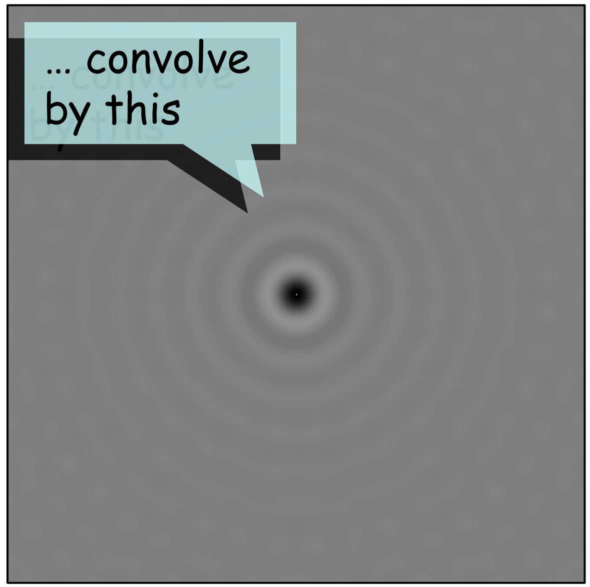
Sharpening results from adding to the image, a copy of itself that has been:

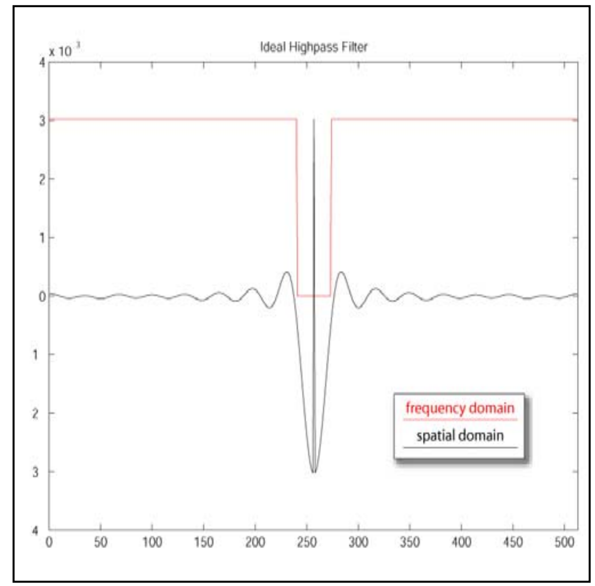
- Pixel-differenced in the spatial domain:
  - Each pixel in the output is a difference between itself and a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 0.
- Highpass filtered in the frequency domain:
  - High frequencies are enhanced or amplified.
  - Individual frequency components are multiplied by an increasing function of ω such as $\alpha\omega = \alpha\sqrt{(u^2+v^2)}$, where α is a constant.

Ideal Highpass Filter
理想高通滤波器

Image size: 512x512
FD notch radius: 16







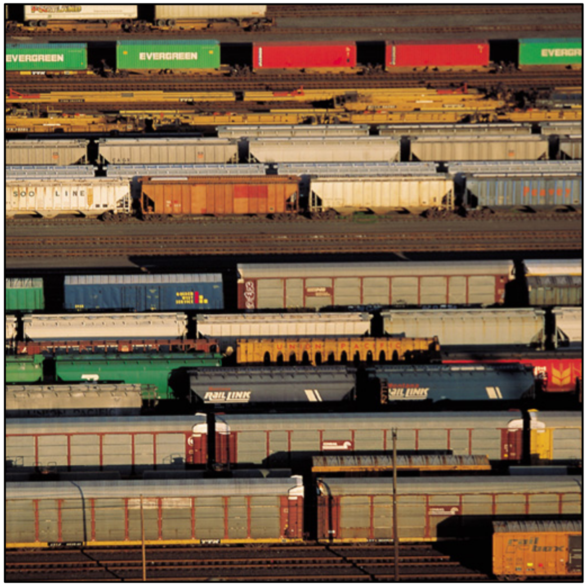
Fourier Domain Rep.

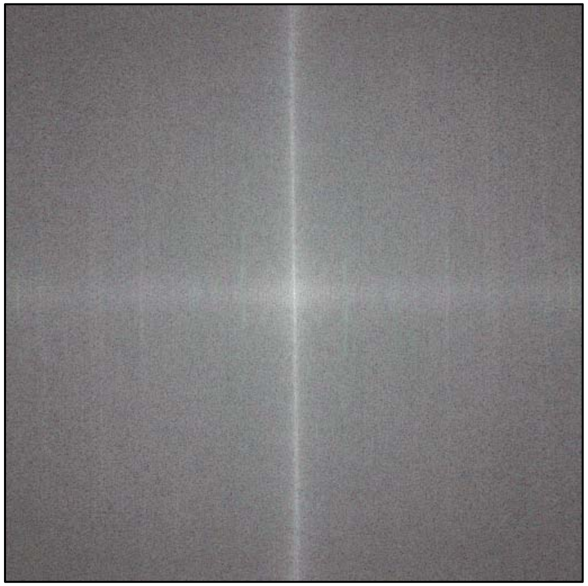
Spatial Representation

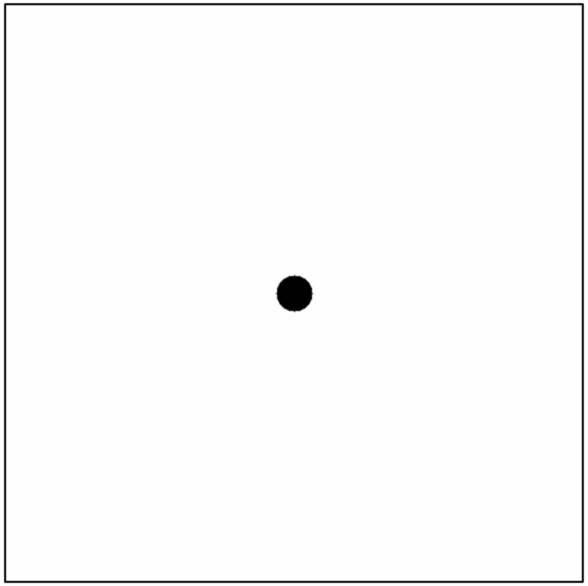
Central Profile

Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16







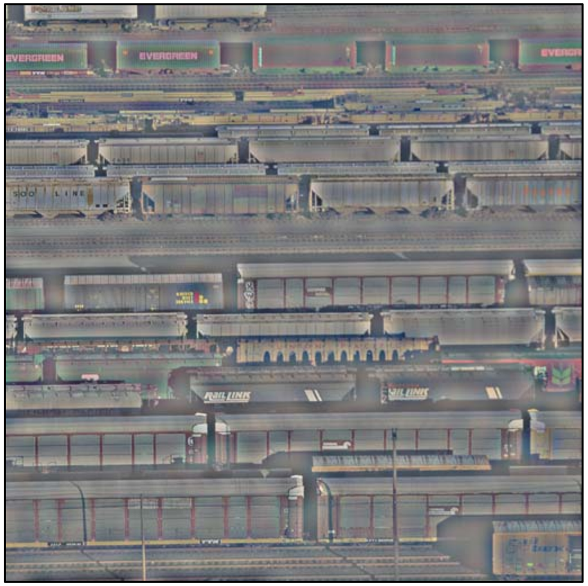
Original Image

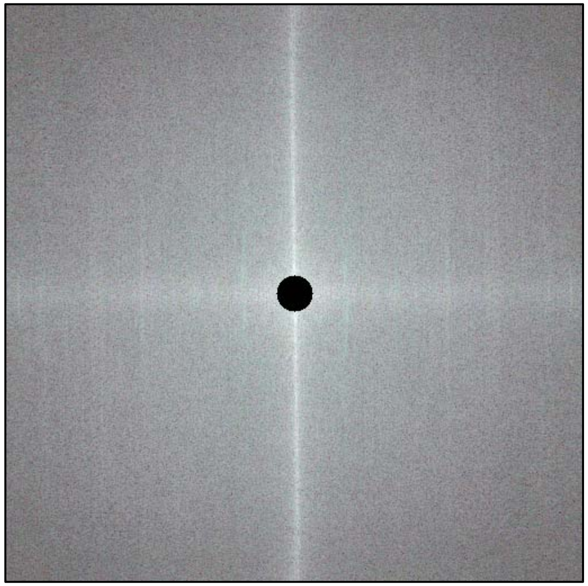
Power Spectrum

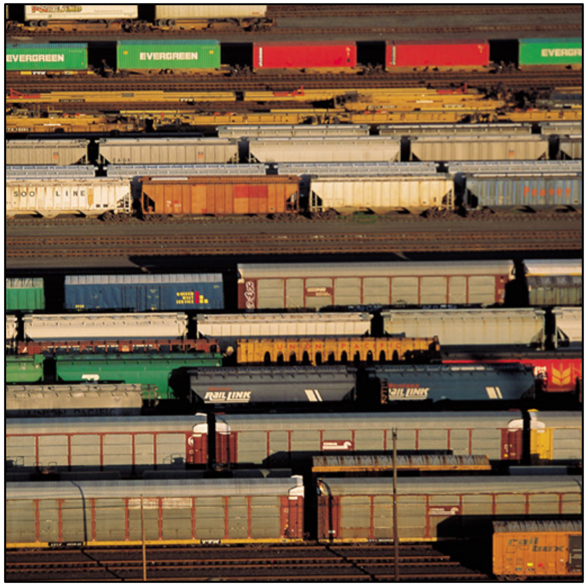
Ideal HPF in FD

Ideal Highpass Filter

Image size: 512x512
FD notch radius: 16







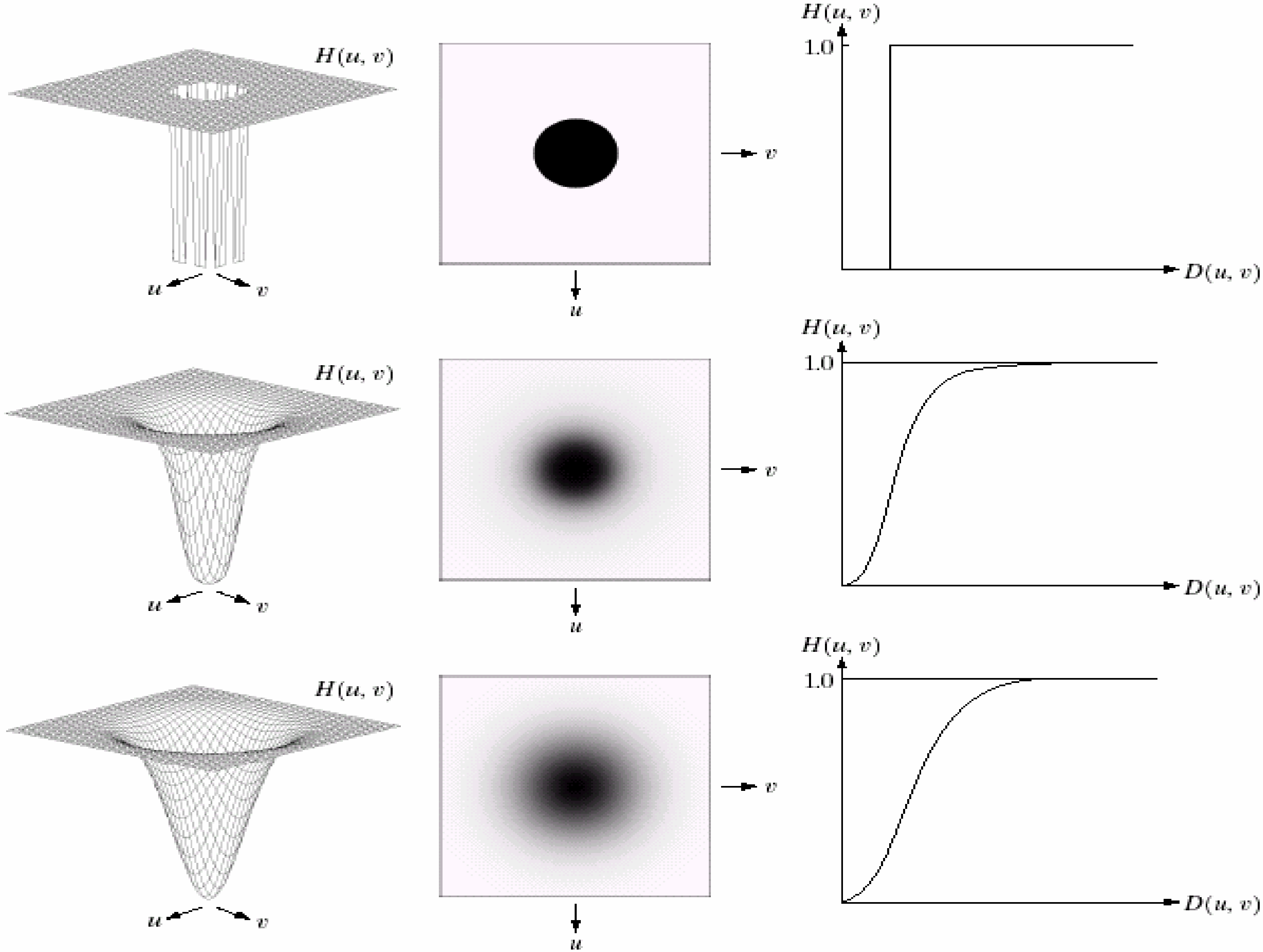
Filtered Image*

Filtered Power Spectrum

Original Image

High-pass Filters
高通滤波器

- $H_{hp}(u,v)=1-H_{lp}(u,v)$
- Ideal: $$H(u,v)=\begin{cases}1 & D(u,v)>D_0\\ 0 & D(u,v)\le D_0\end{cases}$$
- Butterworth: $$|H(u,v)|^2=\frac{1}{1+\left[\frac{D_0}{D(u,v)}\right]^{2n}}$$
- Gaussian: $$H(u,v)=1-e^{-D^2(u,v)/2D_0^2}$$



a	b	c
d	e	f
g	h	i

FIGURE 4.22 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.

High-pass Filters

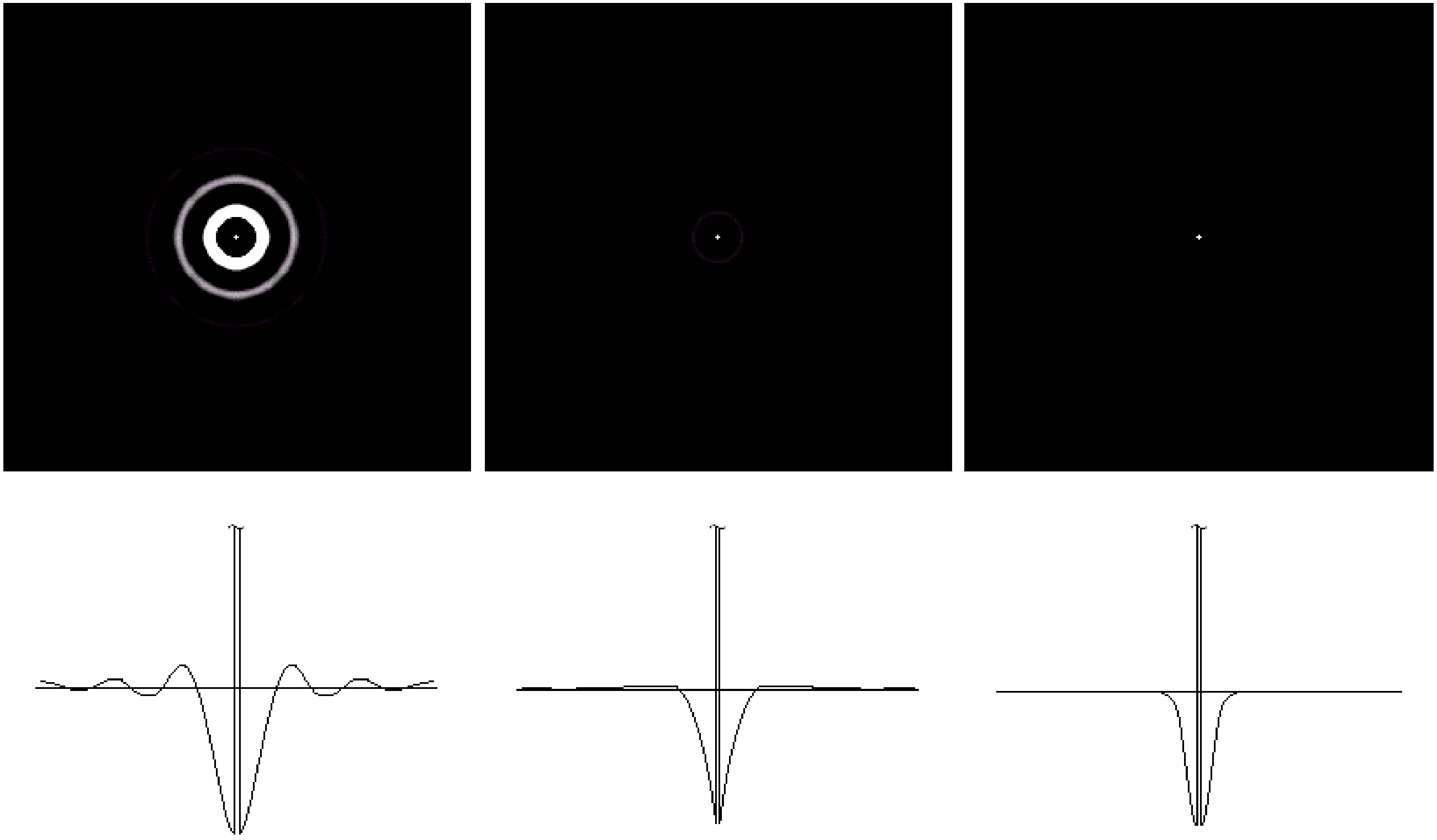
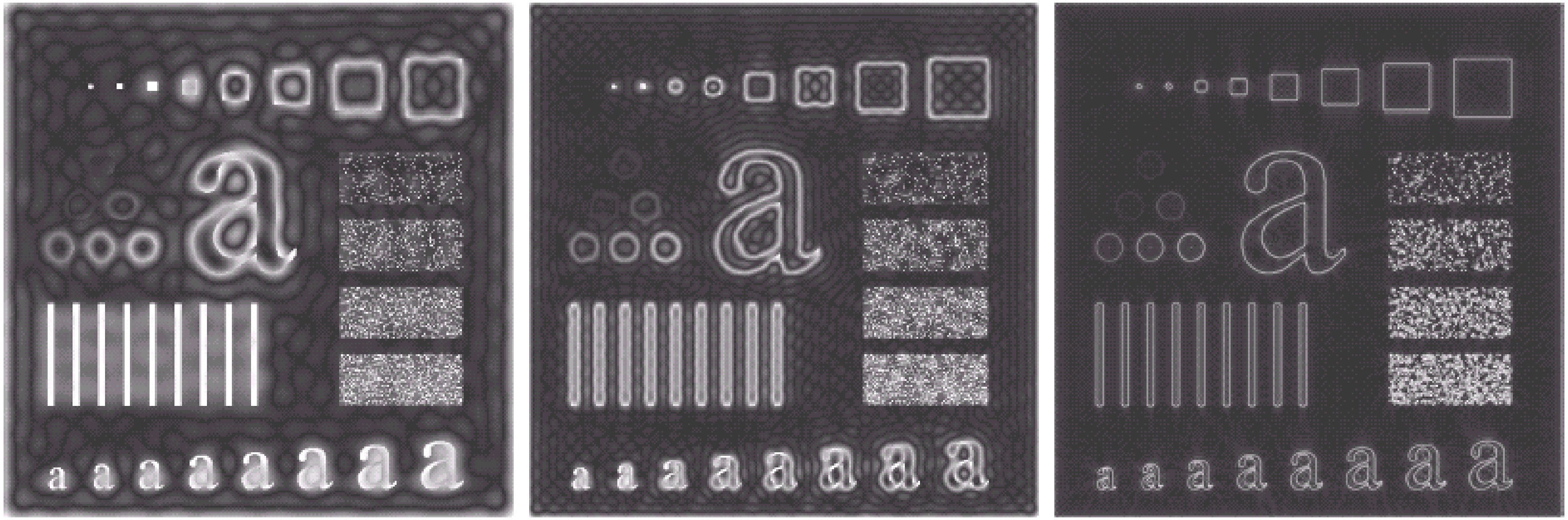


FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

Ideal High-pass Filtering

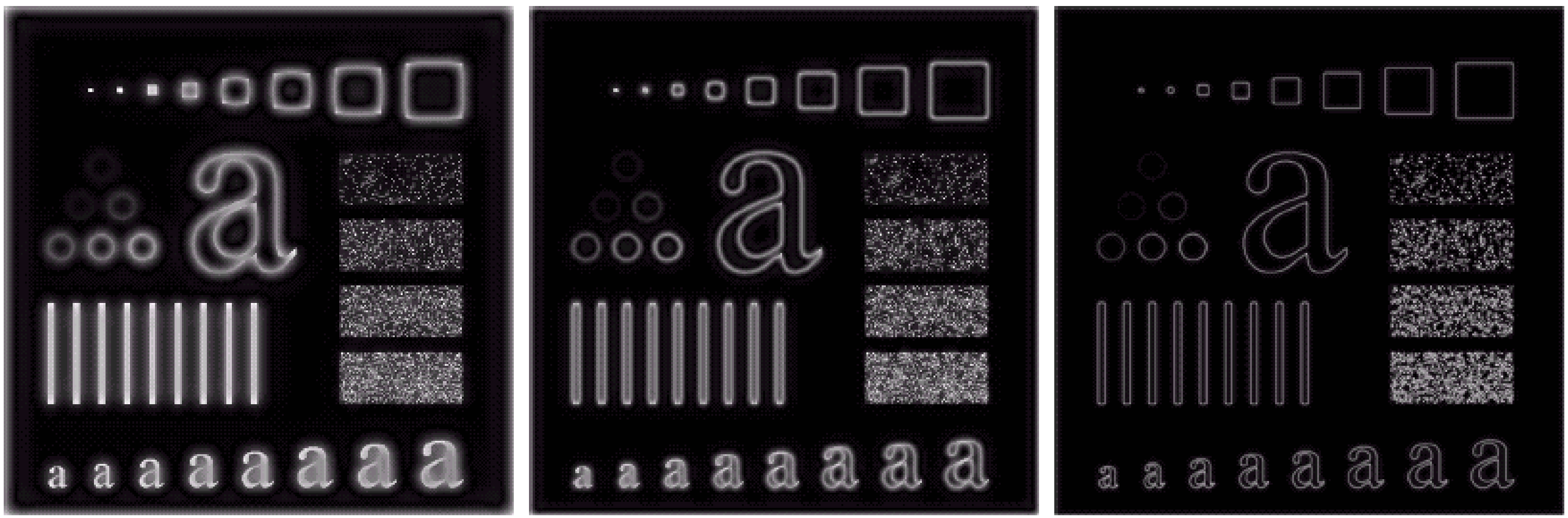
ringing artifacts



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

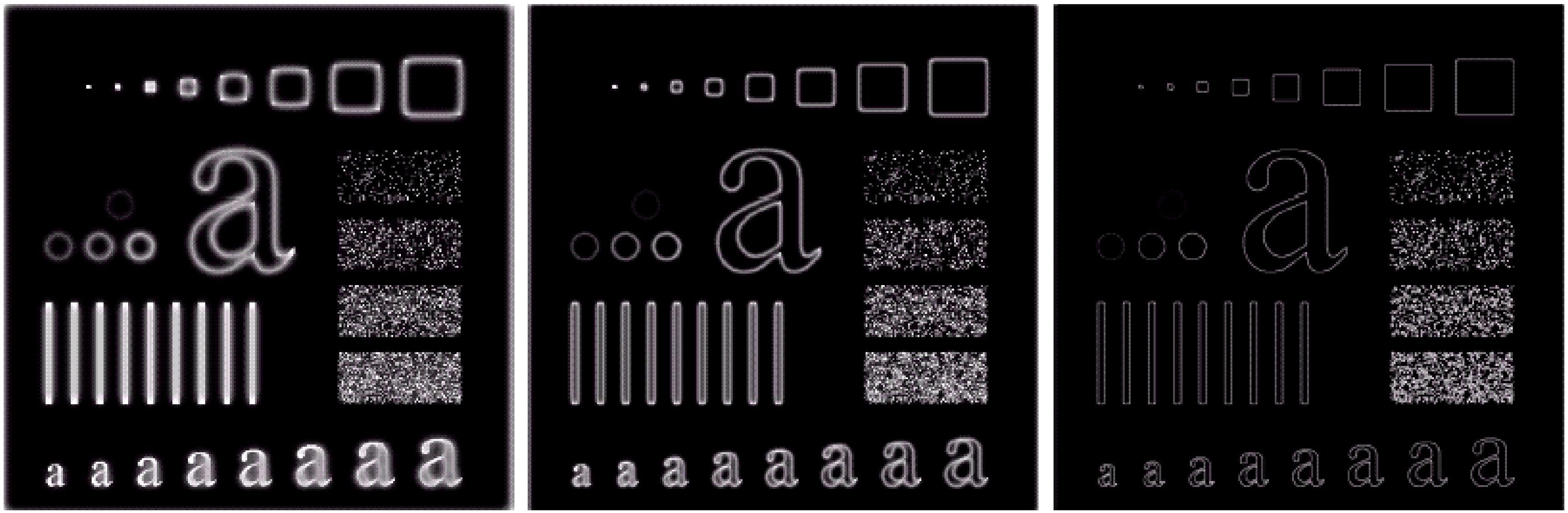
Butterworth High-pass Filtering



a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

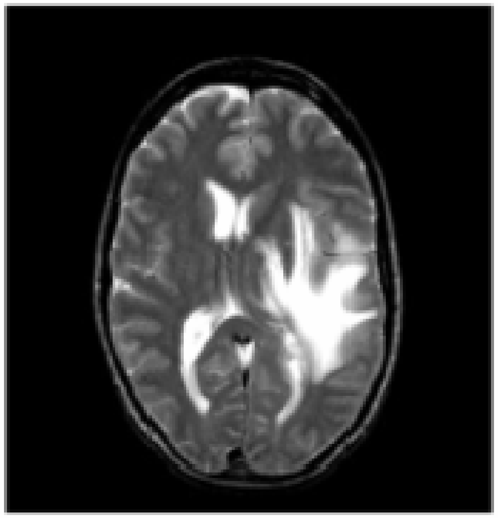
Gaussian High-pass Filtering



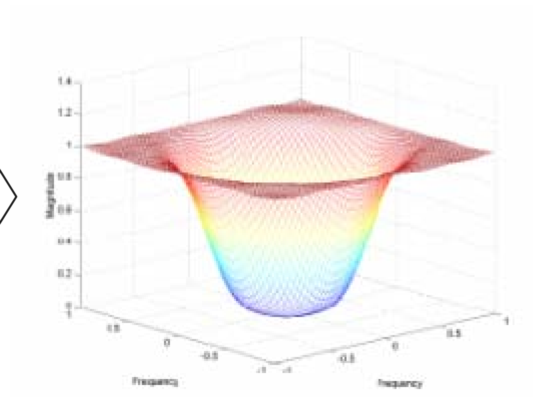
a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

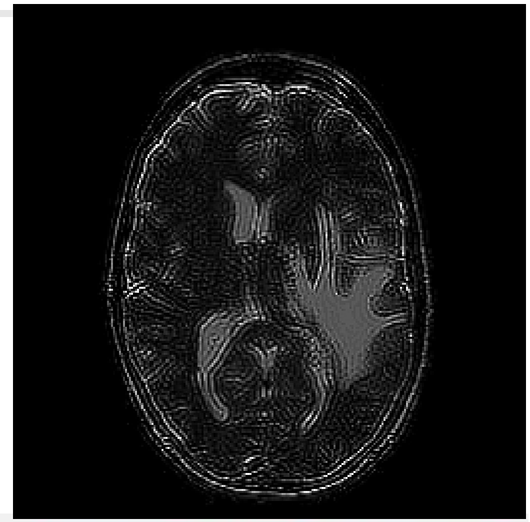
Gaussian High-pass Filtering

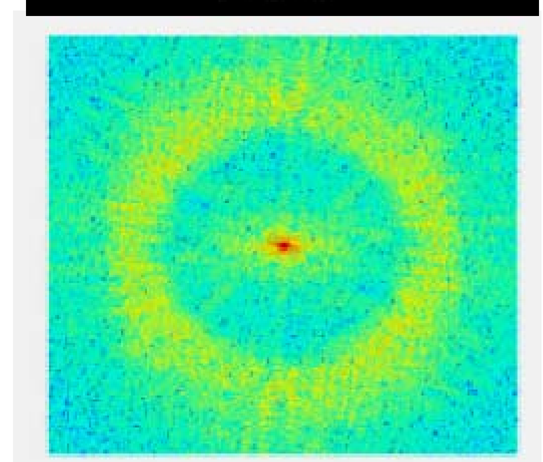


Original image



Gaussian filter H(u,v)





Filtered image and its FT

Laplacian in Frequency Domain

$$\Im\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] = \boxed{-(u^2+v^2)}F(u,v)$$

$$H_1(u,v) = -(u^2+v^2)$$

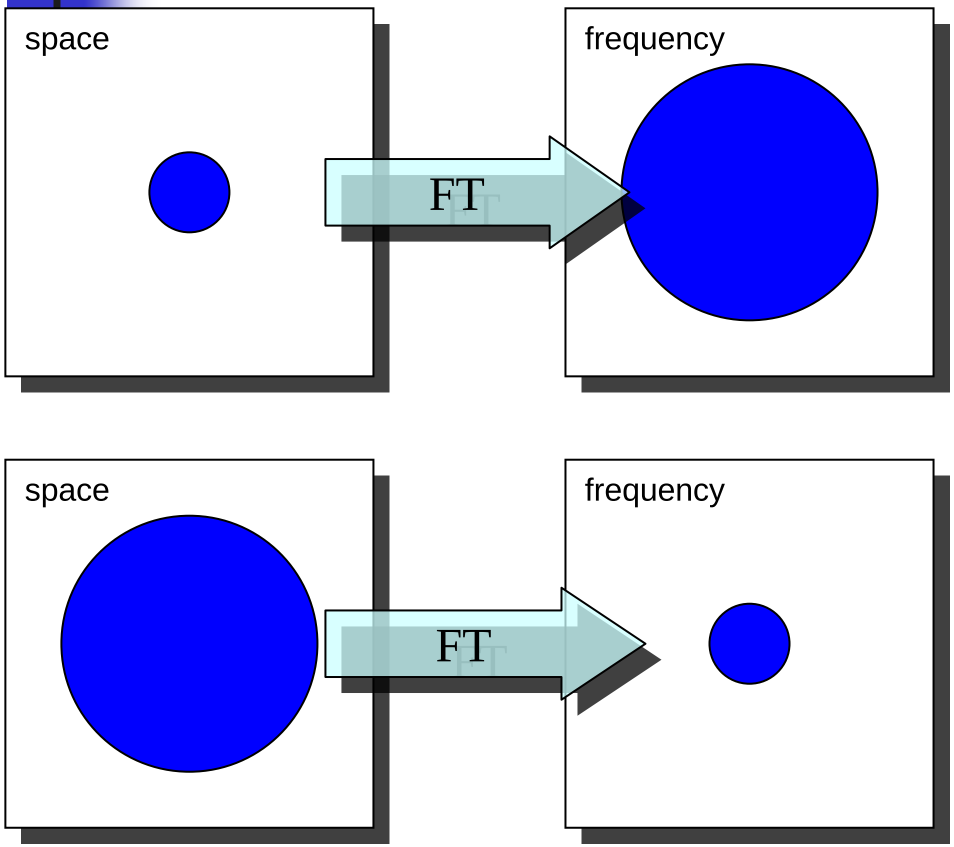
Frequency domain

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Spatial domain

Laplacian operator

The Uncertainty Relation
不确定性关系

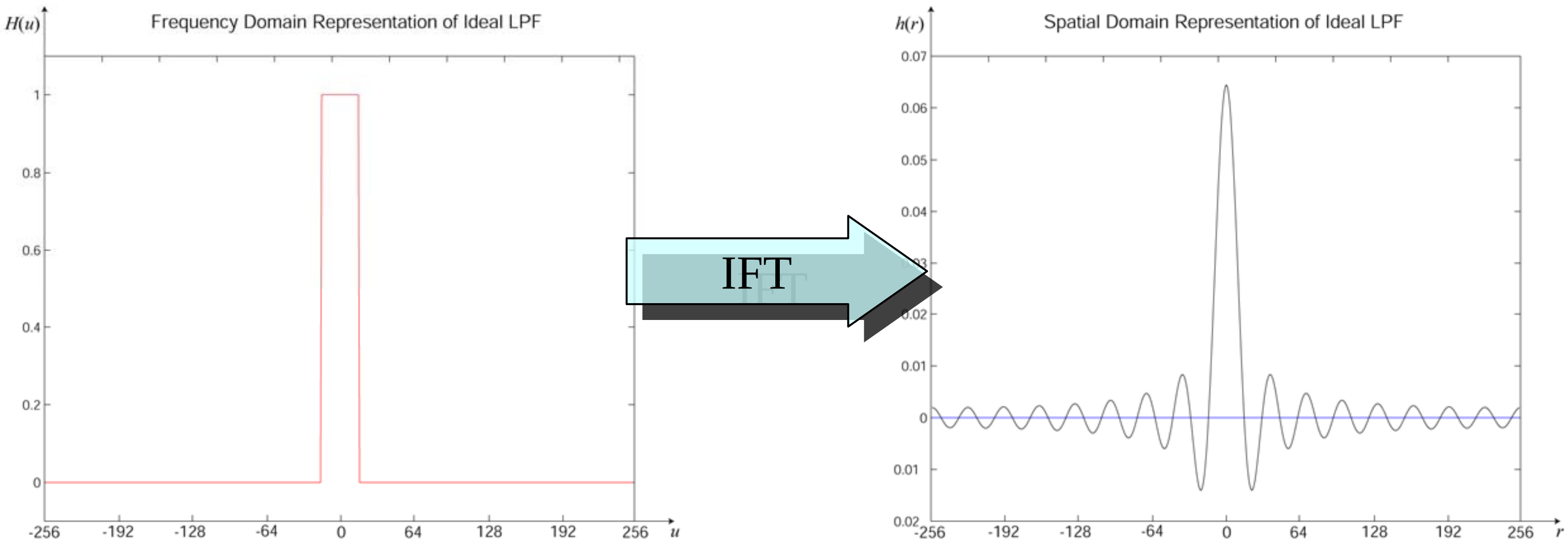


If $\Delta x \, \Delta y$ is the extent of the object in space and if $\Delta u \, \Delta v$ is its extent in frequency then,

$$\Delta x \, \Delta y \cdot \Delta u \, \Delta v \geq \frac{1}{16\pi^2}$$

A small object in space has a large frequency extent and vice-versa.

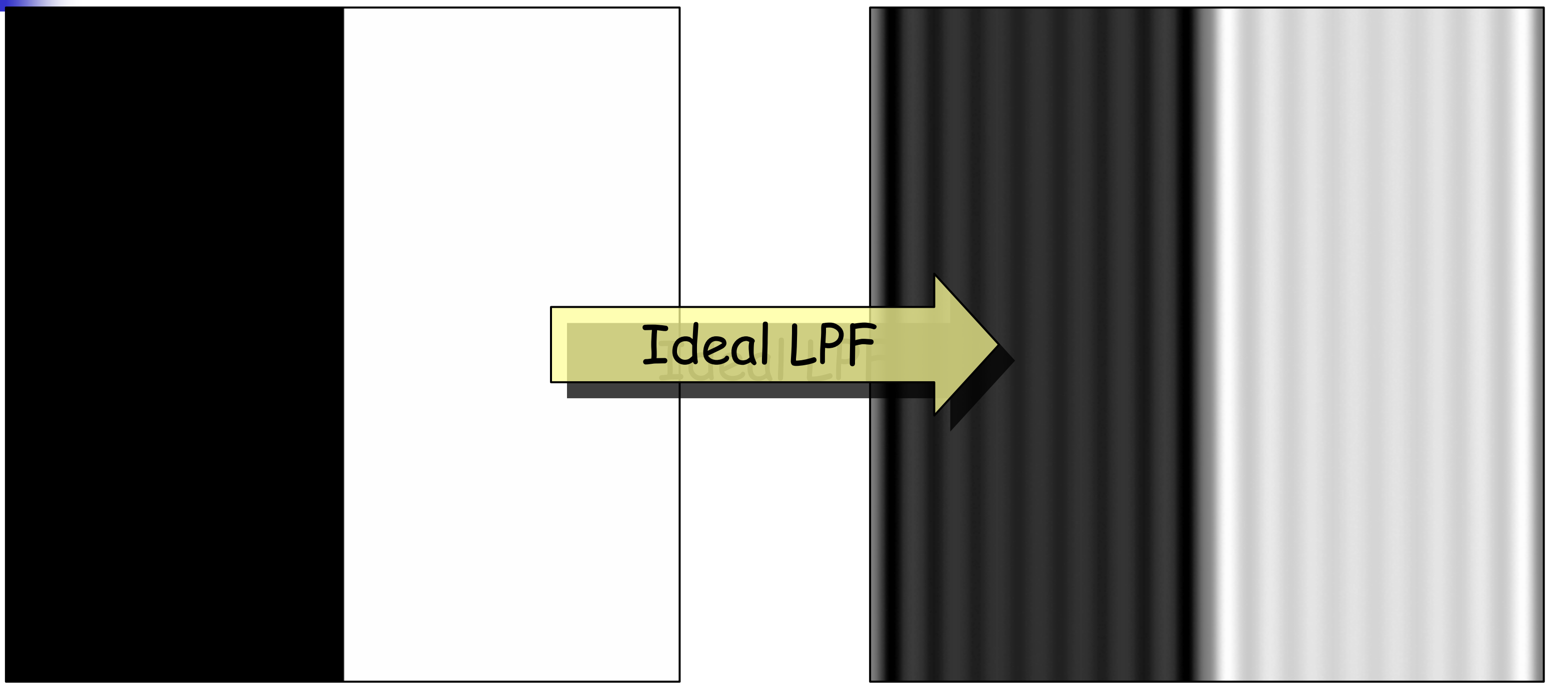
Ideal Filters Do Not Produce Ideal Results



A sharp cutoff in the frequency domain…

…causes ringing in the spatial domain.

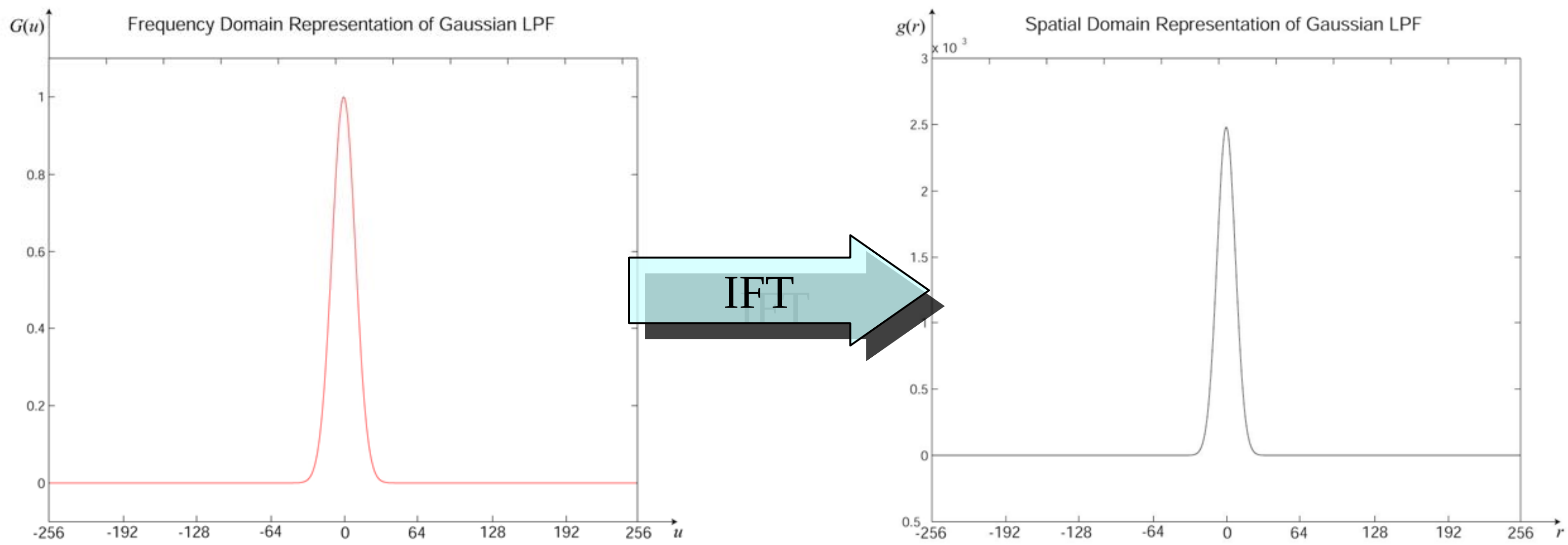
Ideal Filters Do Not Produce Ideal Results



Blurring the image above with an ideal lowpass filter…

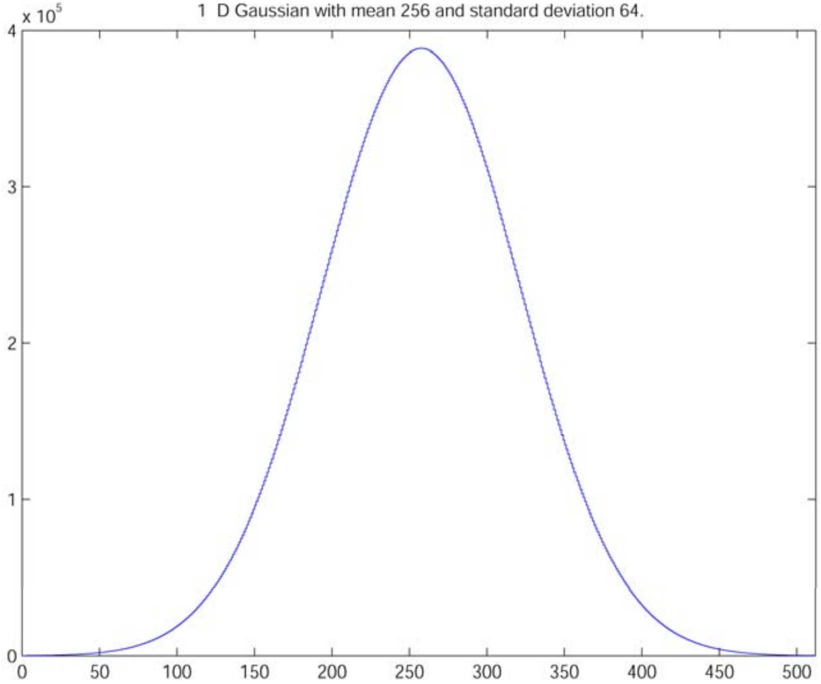
…distorts the results with ringing or ghosting.

Optimal Filter: The Gaussian



The Gaussian filter optimizes the uncertainty relation.
It provides the sharpest cutoff with the least ringing.

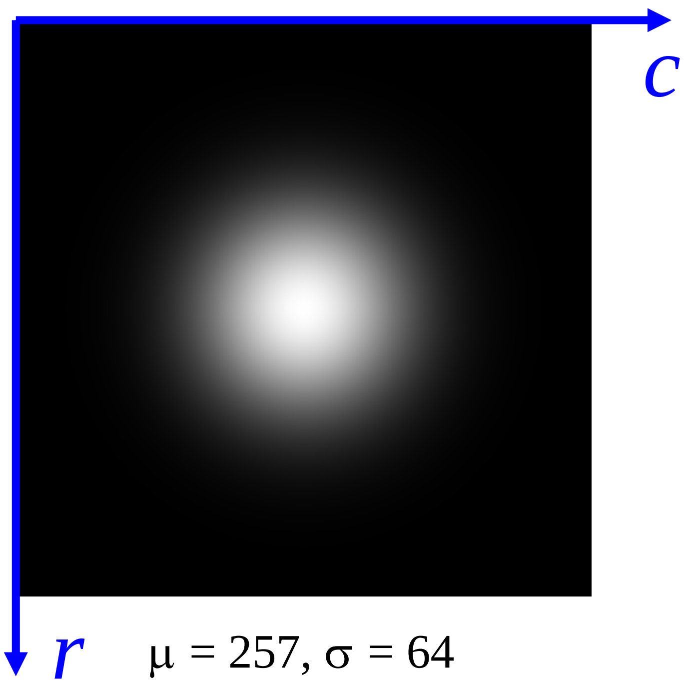
One-Dimensional Gaussian



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

Two-Dimensional Gaussian

$R = 512$, $C = 512$



$\mu = 257$, $\sigma = 64$

If μ and σ are different for r & c…

$$g(r,c) = g(r)g(c)$$

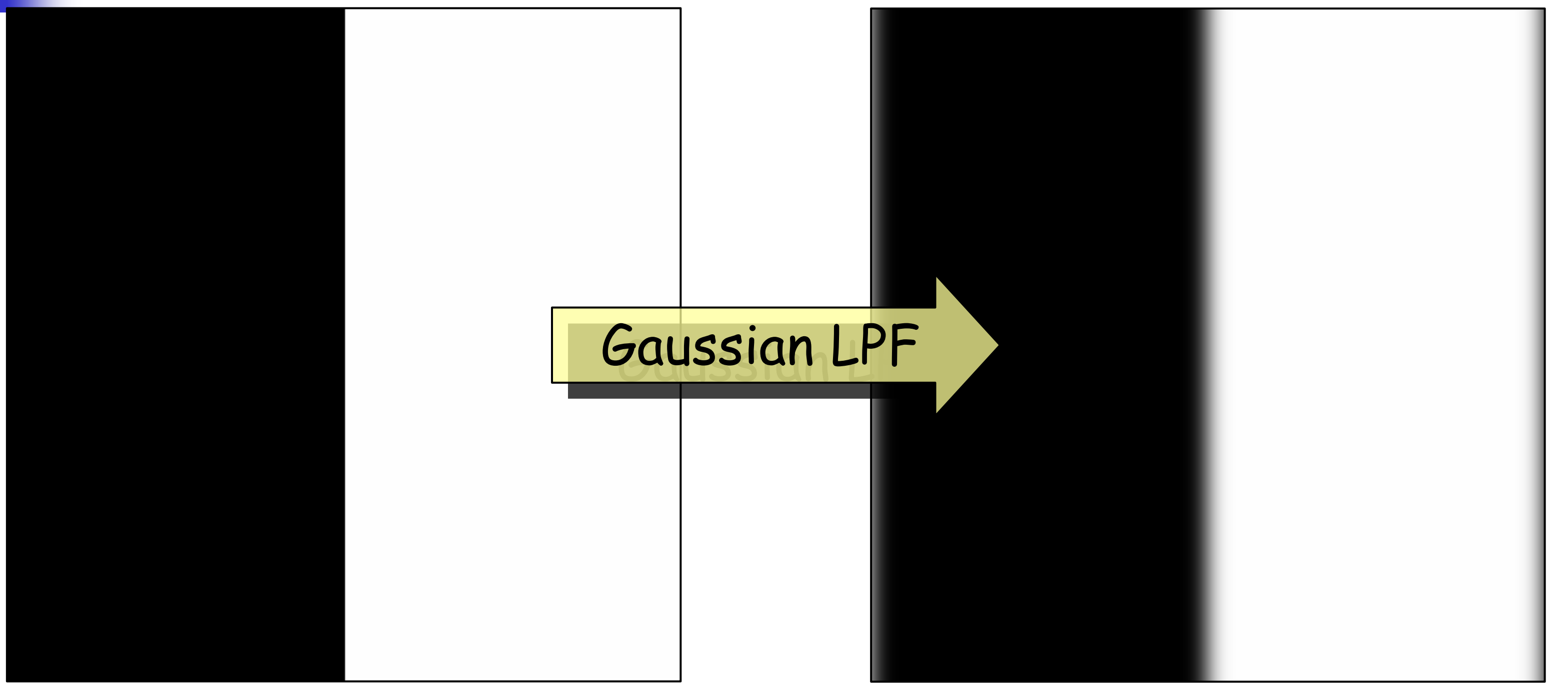
$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{(r-\mu_r)^2}{2\sigma_r^2} - \frac{(y-\mu_c)^2}{2\sigma_c^2}}$$

$$= \frac{1}{\sigma_r \sigma_c 2\pi} e^{-\frac{\sigma_c^2 (x-\mu_r)^2 + \sigma_r^2 (y-\mu_c)^2}{2\sigma_r^2 \sigma_c^2}}$$

…or if μ and σ are the same for r & c.

$$g(r,c) = \frac{1}{\sigma^2 2\pi} e^{-\frac{(r-\mu)^2 + (c-\mu)^2}{2\sigma^2}}$$

Optimal Filter: The Gaussian

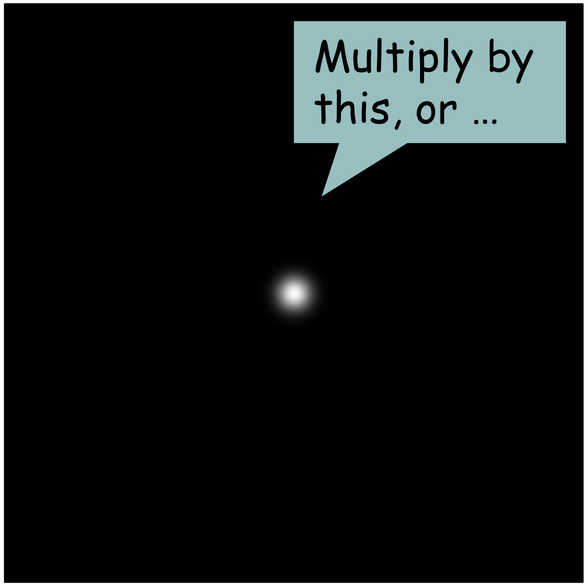


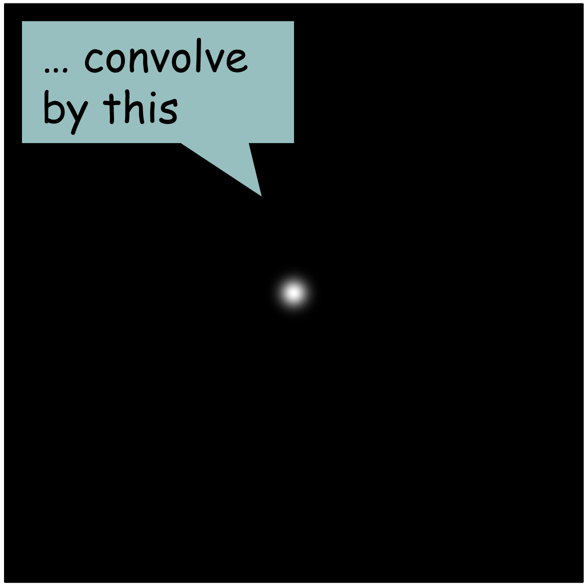
With a gaussian lowpass filter, the image above …

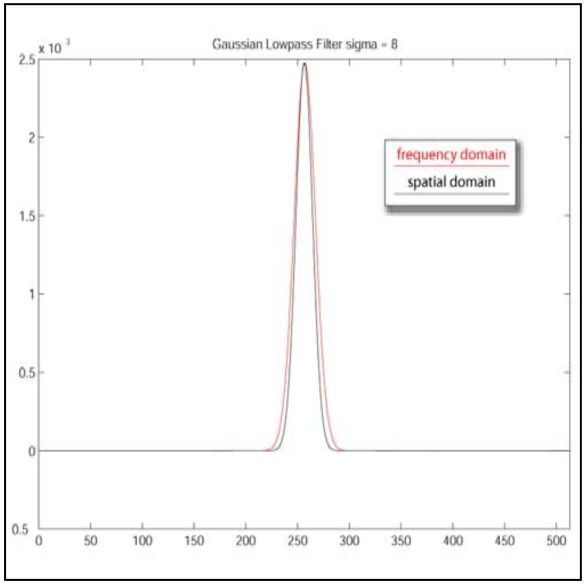
… is blurred without ringing or ghosting.

Gaussian Lowpass Filter
高斯低通滤波器

Image size: 512x512
SD filter sigma = 8







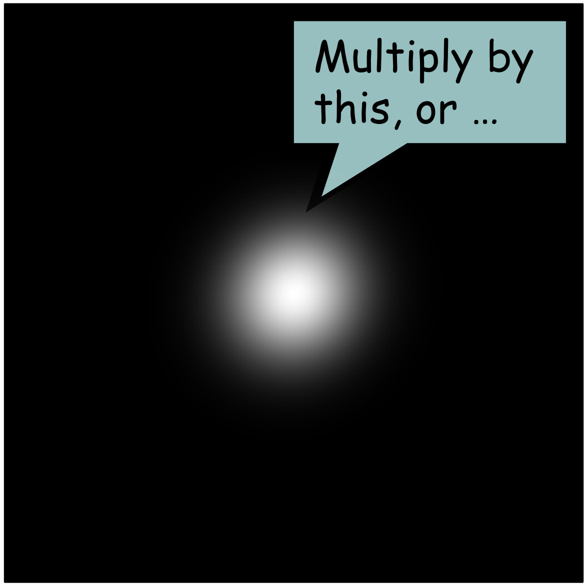
Fourier Domain Rep.

Spatial Representation

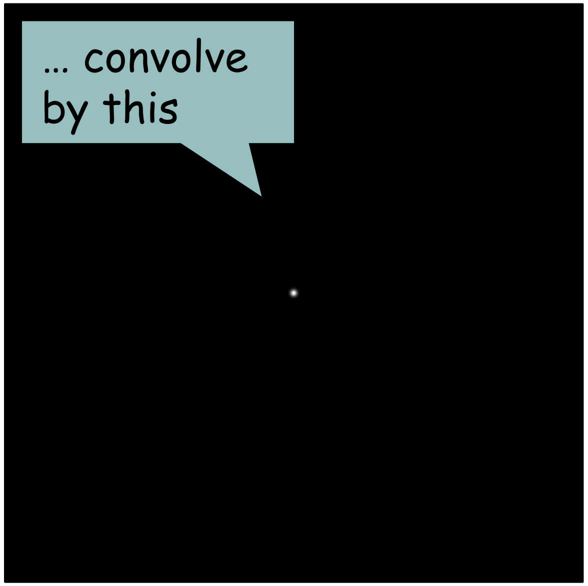
Central Profile

Gaussian Lowpass Filter

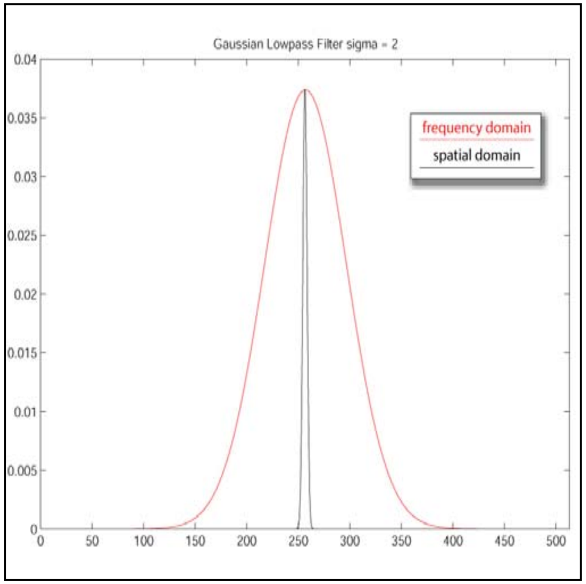
Image size: 512x512
SD filter sigma = 2



Fourier Domain Rep.



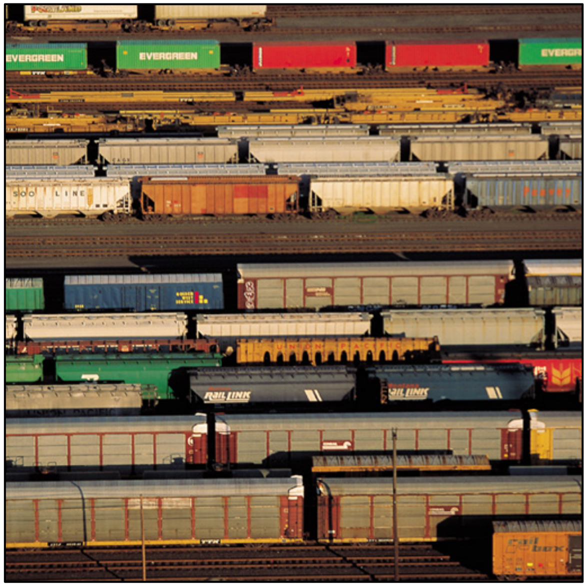
Spatial Representation

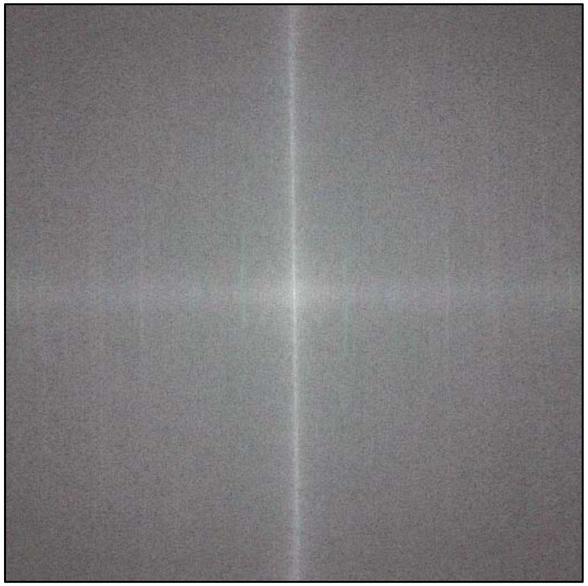


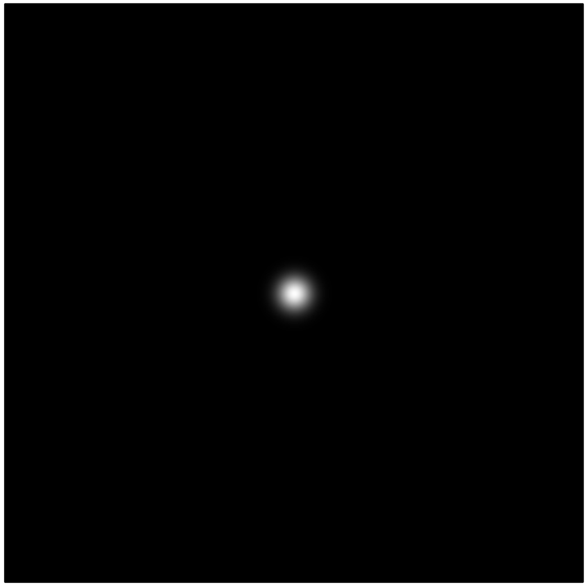
Central Profile

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8







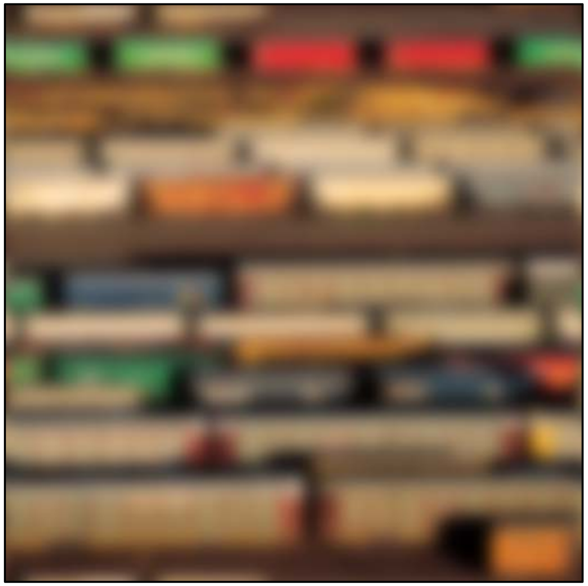
Original Image

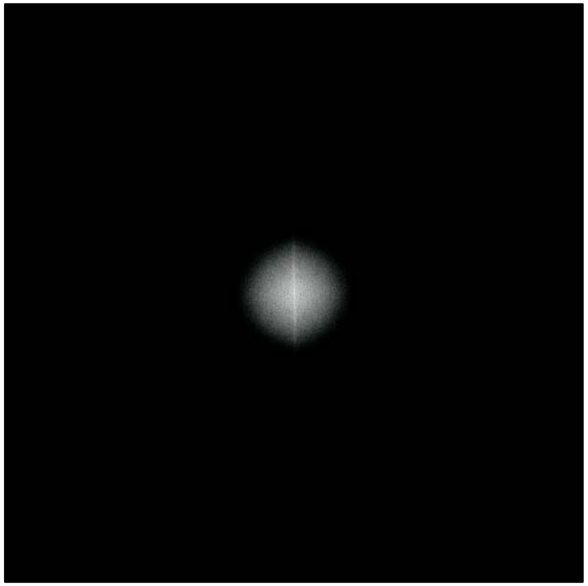
Power Spectrum

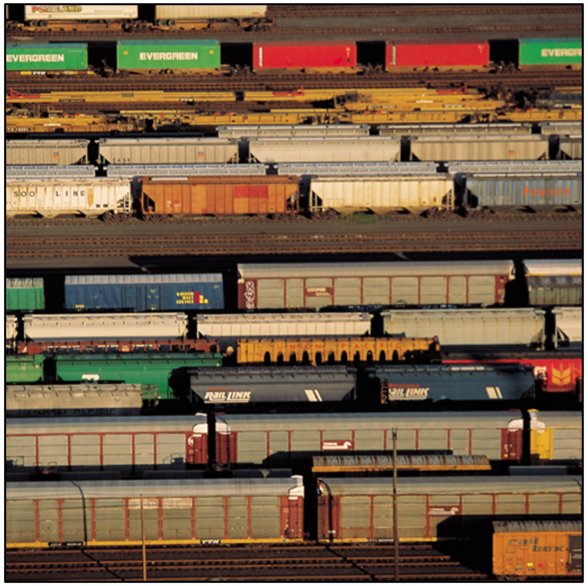
Gaussian LPF in FD

Gaussian Lowpass Filter

Image size: 512x512
SD filter sigma = 8





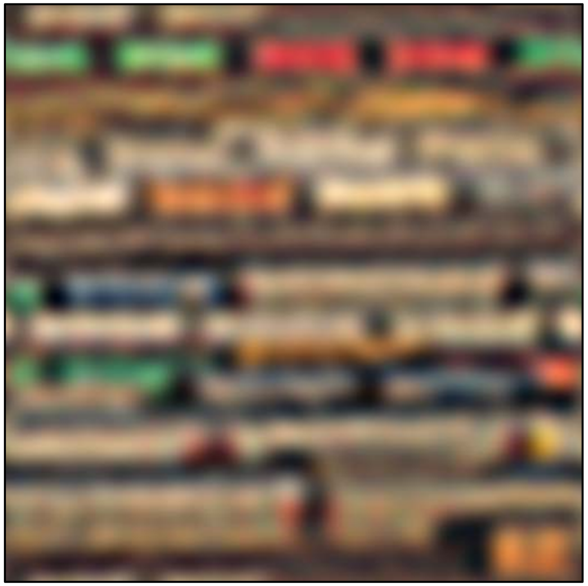


Filtered Image

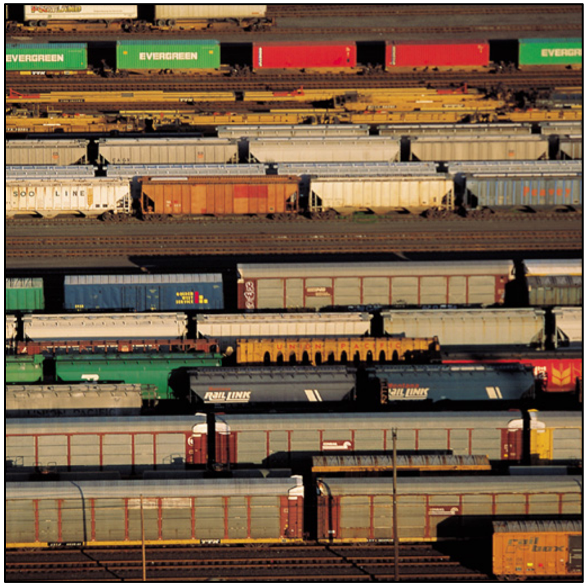
Filtered Power Spectrum

Original Image

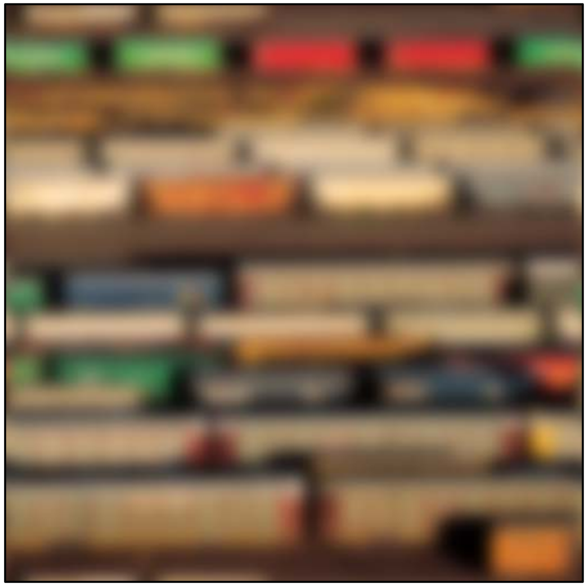
Comparison of Ideal and Gaussian Lowpass Filters



Ideal LPF



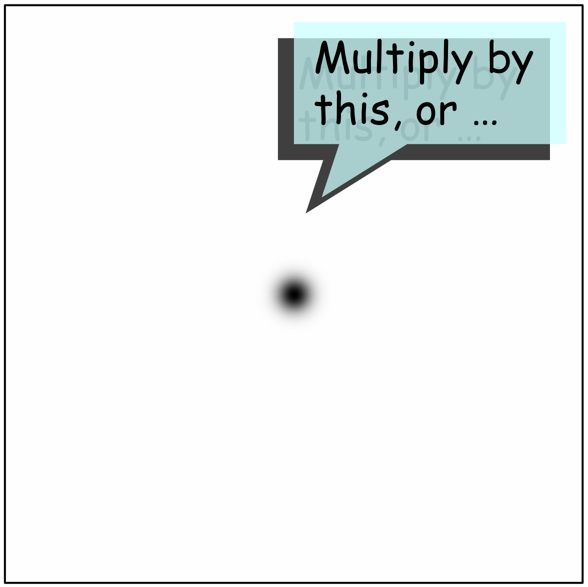
Original Image

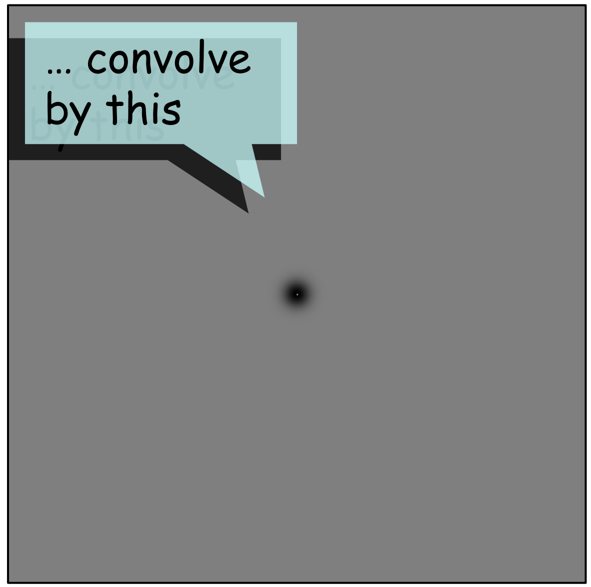


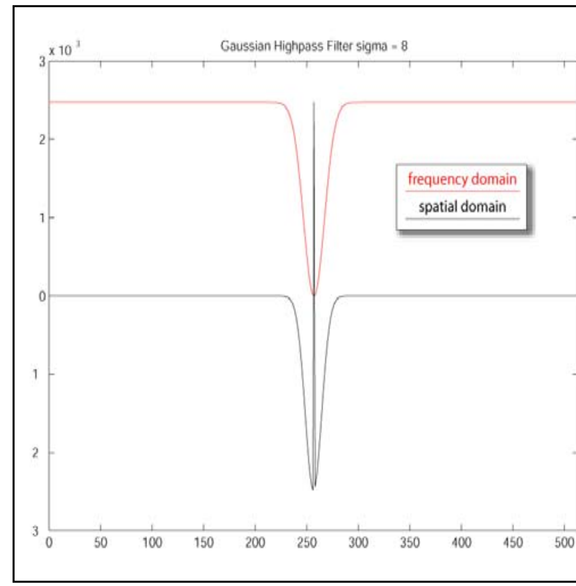
Gaussian LPF

Gaussian Highpass Filter
高斯高通滤波器

Image size: 512x512
FD notch sigma = 8







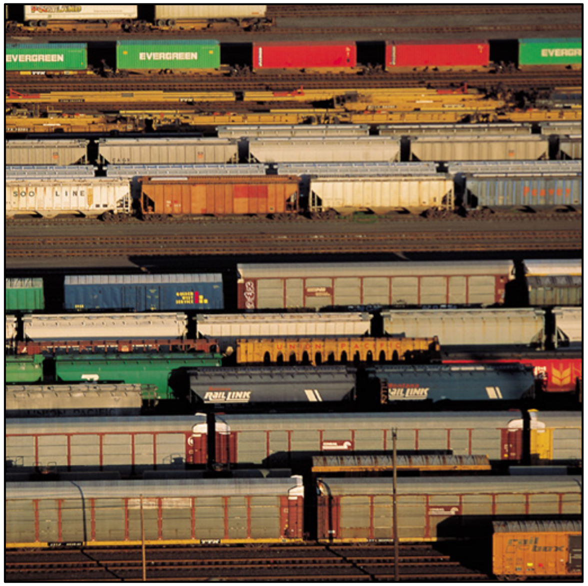
Fourier Domain Rep.

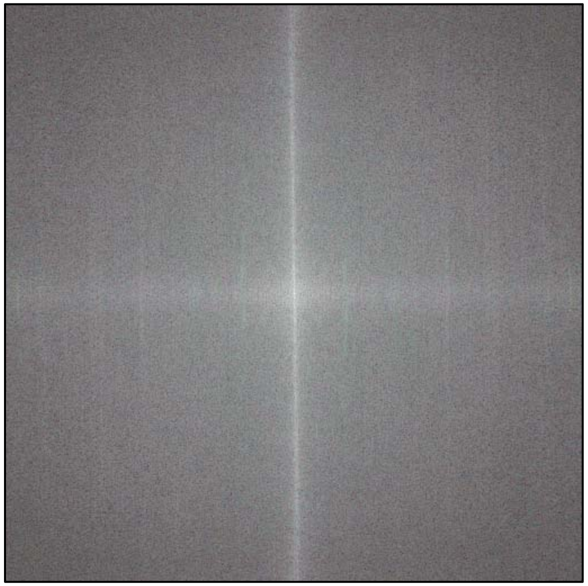
Spatial Representation

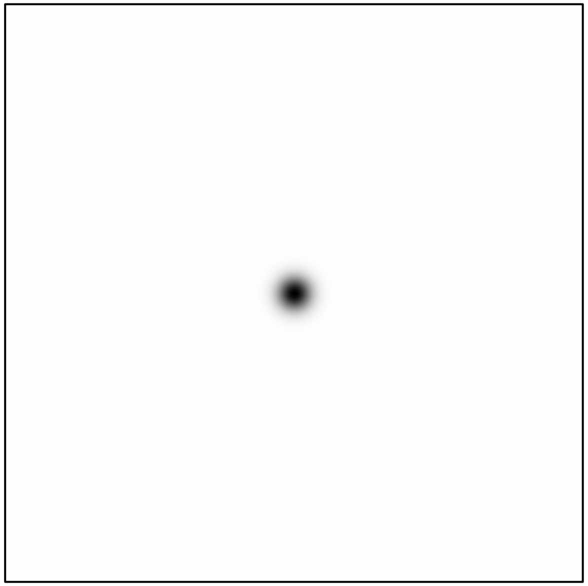
Central Profile

Gaussian Highpass Filter

Image size: 512x512
FD notch sigma = 8







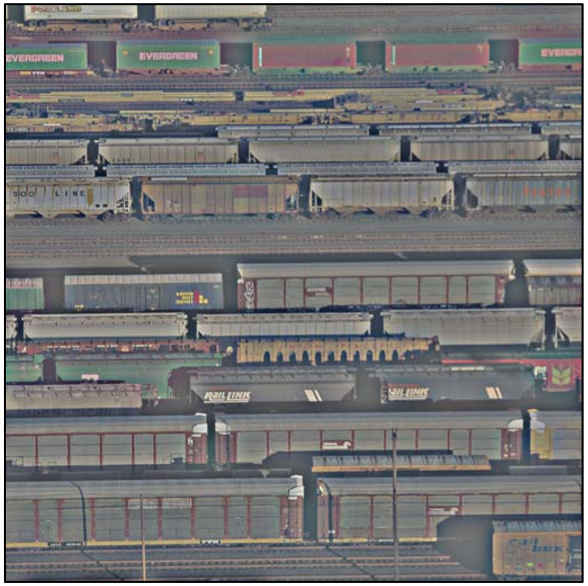
Original Image

Power Spectrum

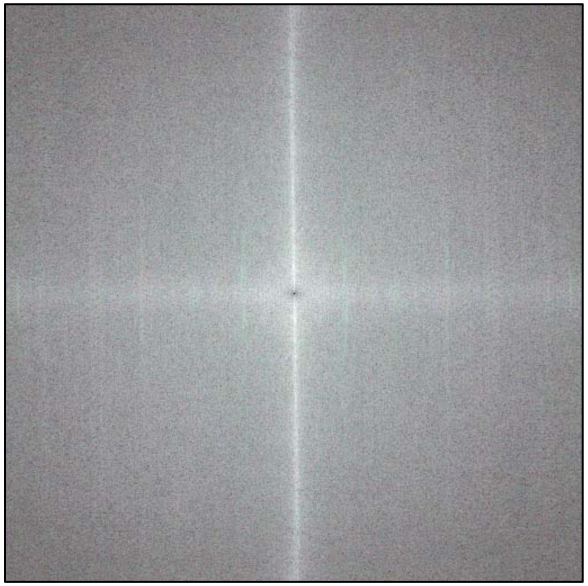
Gaussian HPF in FD

Gaussian Highpass Filter

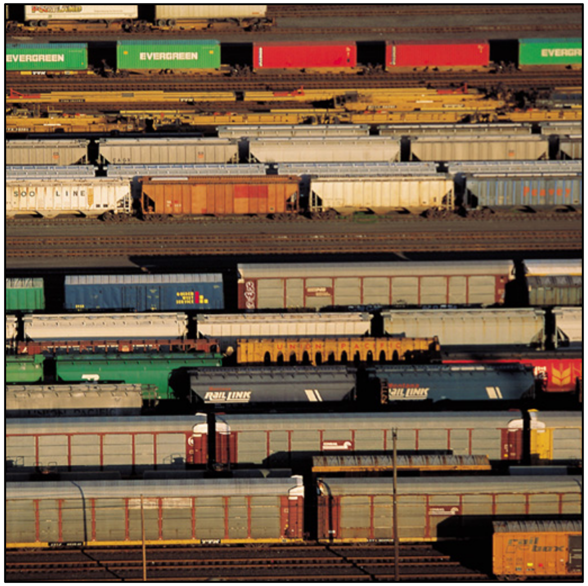
Image size: 512x512
FD notch sigma = 8



Filtered Image*



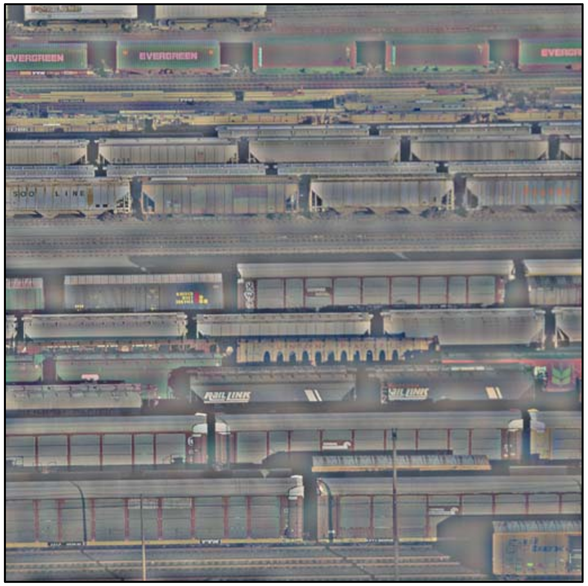
Filtered Power Spectrum



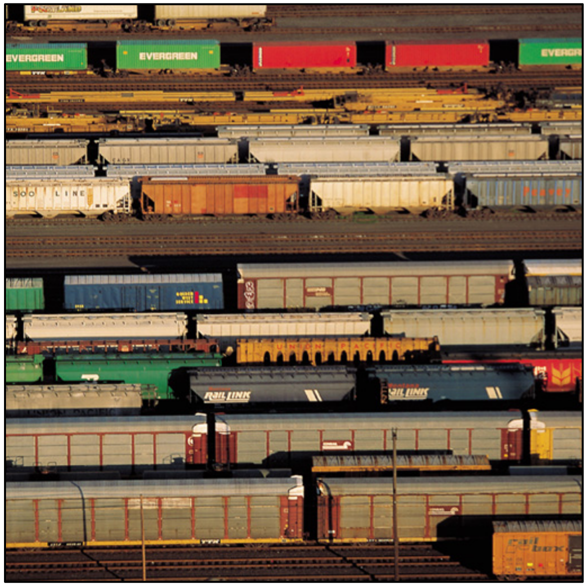
Original Image

Comparison of Ideal and Gaussian Highpass Filters

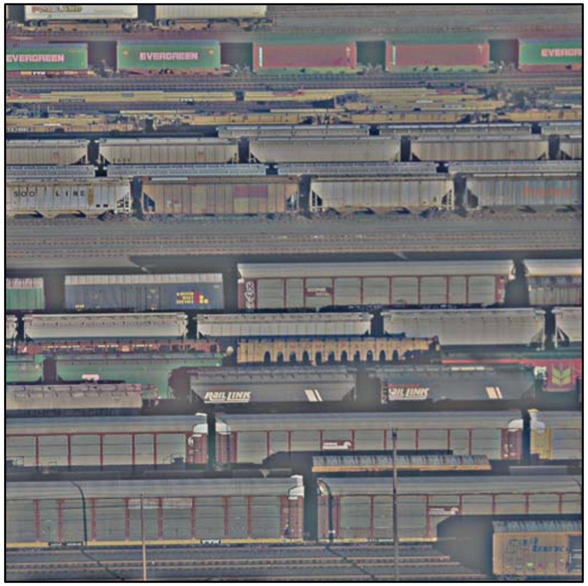
*signed image; 0 mapped to 128



Ideal HPF*



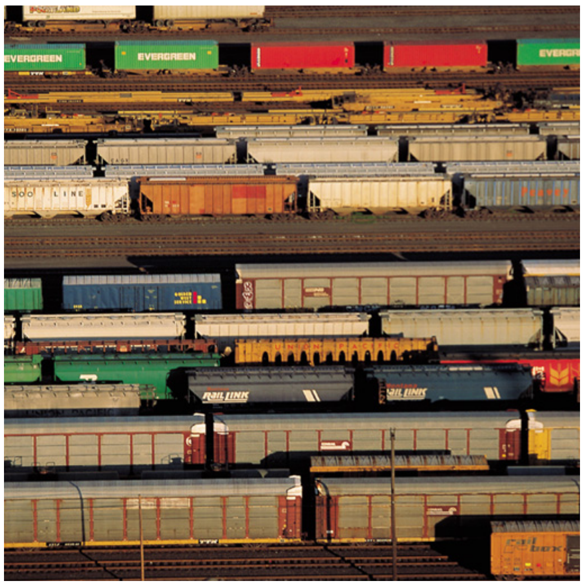
Original Image



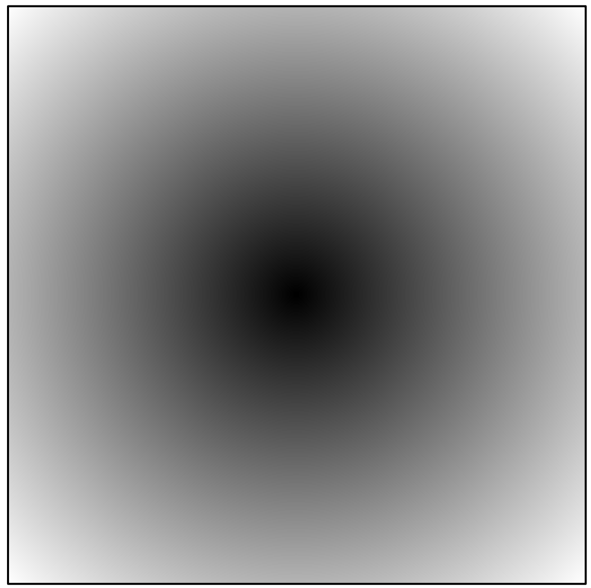
Gaussian HPF*

Another Highpass Filter

*signed image; 0 mapped to 128



original image

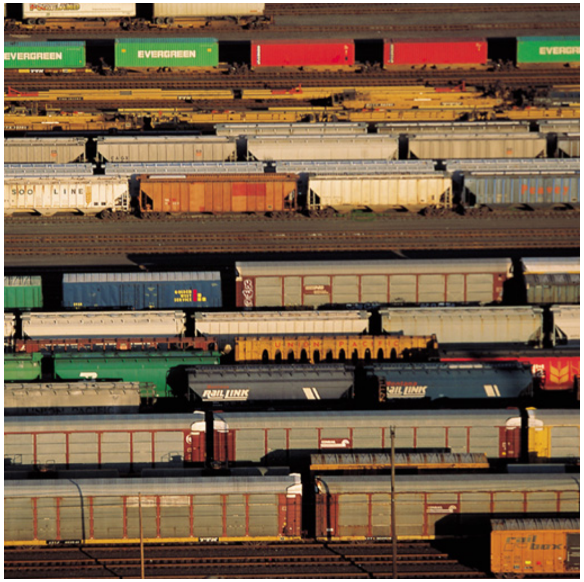


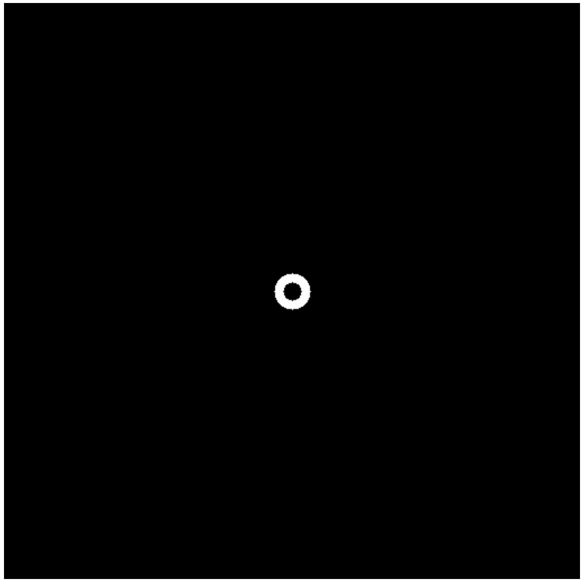
filter power spectrum

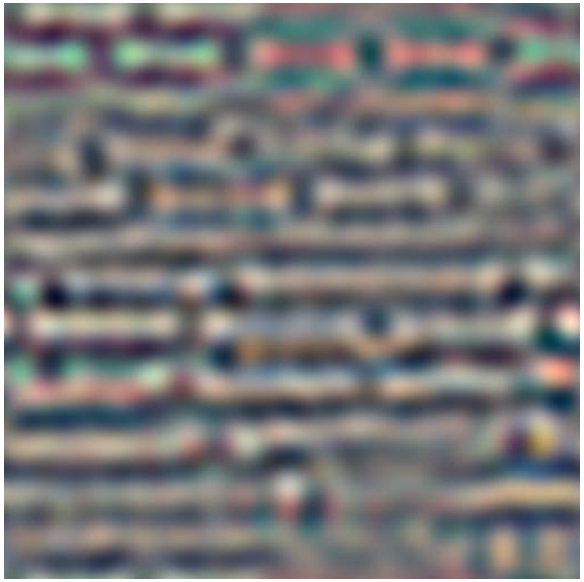
filtered image*

Ideal Bandpass Filter
理想带通滤波器

*signed image; 0 mapped to 128







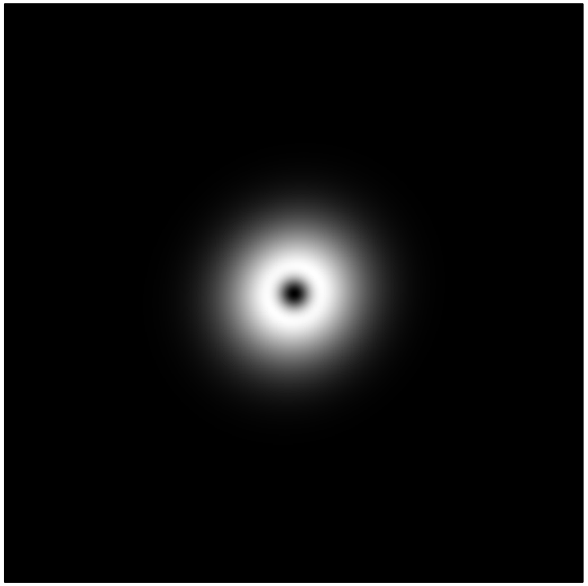
original image

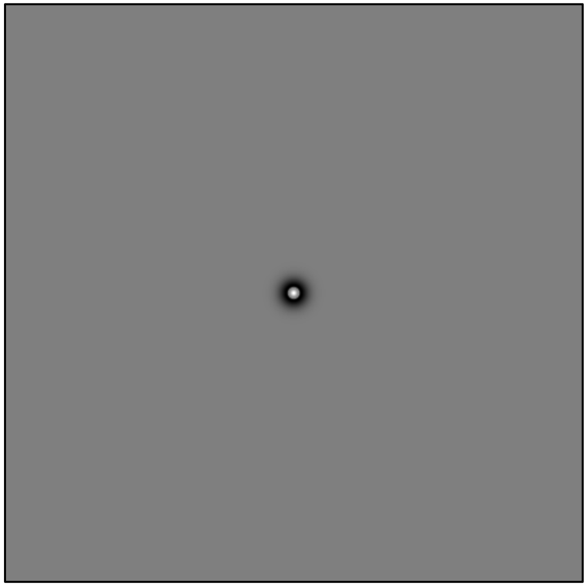
filter power spectrum

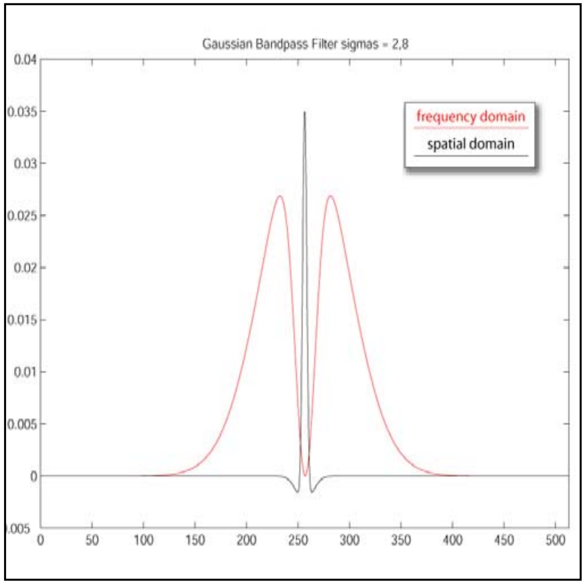
filtered image*

Gaussian Bandpass Filter
高斯带通滤波器

Image size: 512x512
sigma = 2 - sigma = 8







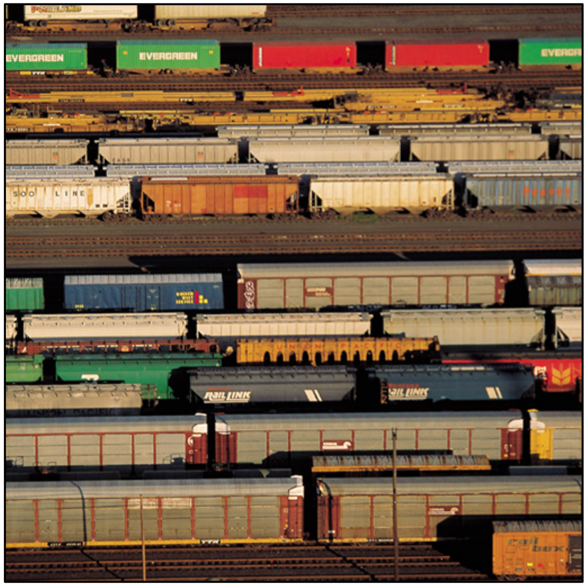
Fourier Domain Rep.

Spatial Representation

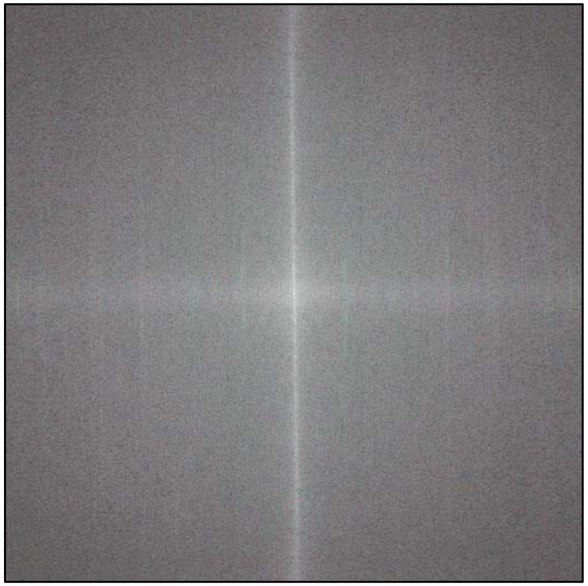
Central Profile

Gaussian Bandpass Filter

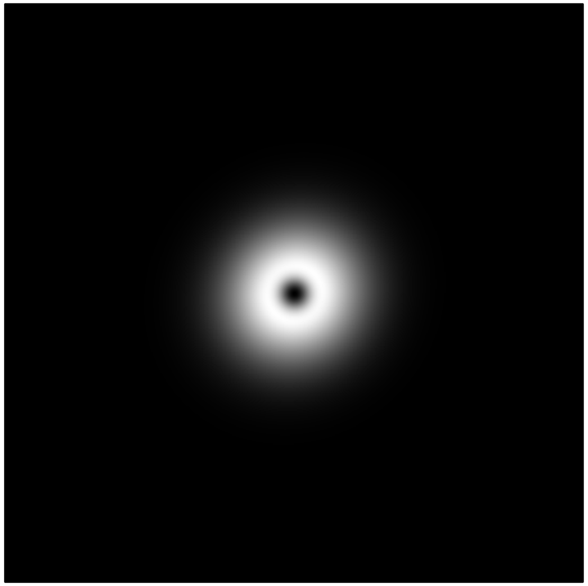
Image size: 512x512
sigma = 2 - sigma = 8



Original Image



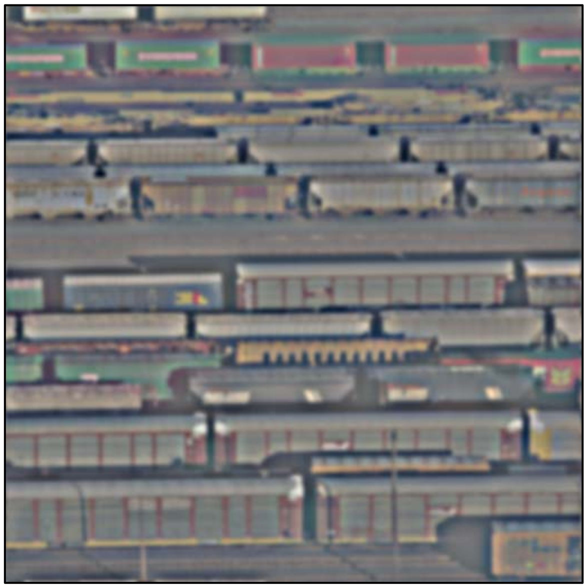
Power Spectrum



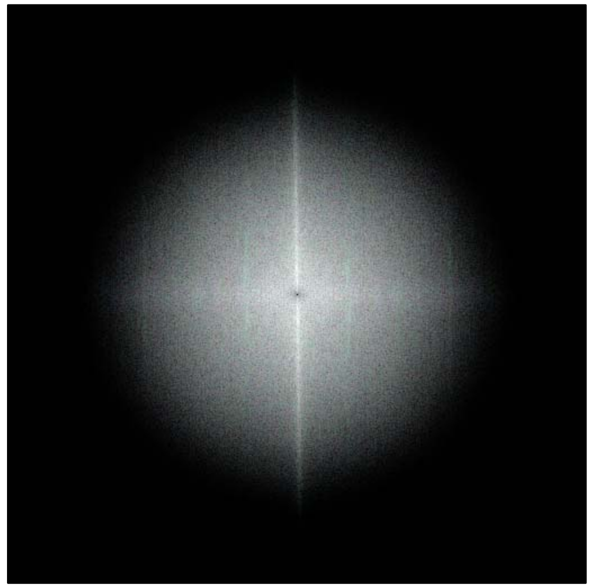
Gaussian BPF in FD

Gaussian Bandpass Filter

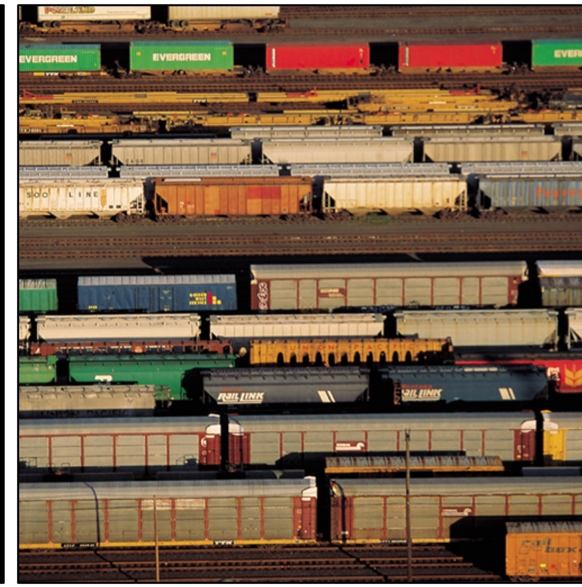
Image size: 512x512
sigma = 2 - sigma = 8



Filtered Image*



Filtered Power Spectrum

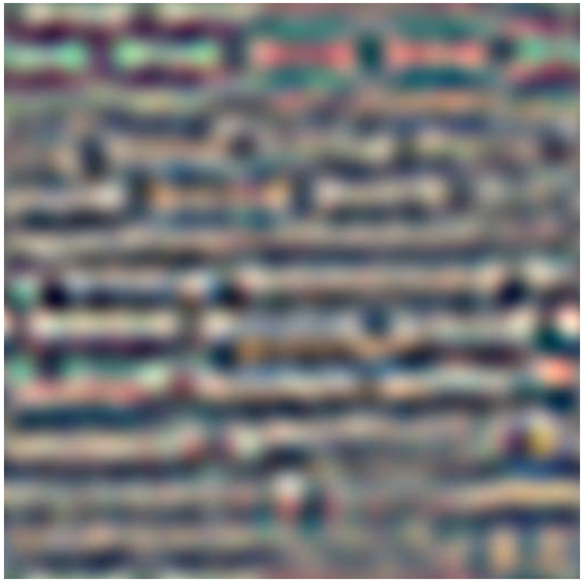


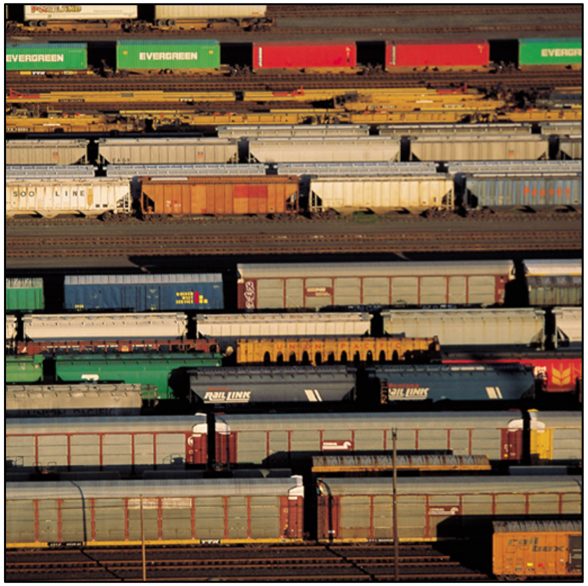
Original Image

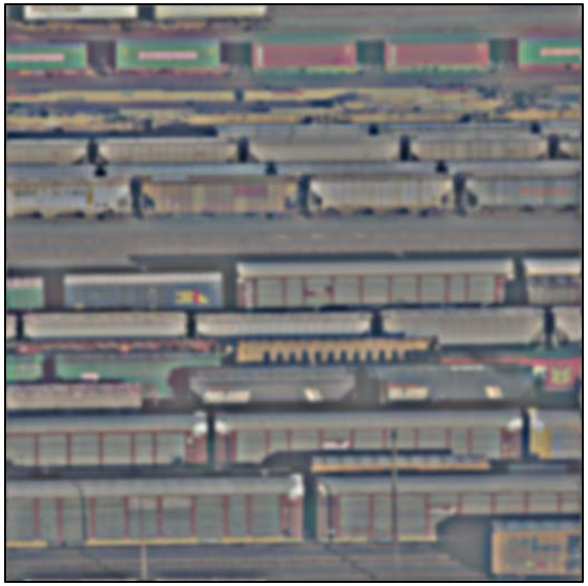
*signed image; 0 mapped to 128

Comparison of Ideal and Gaussian Bandpass Filters

*signed image; 0 mapped to 128





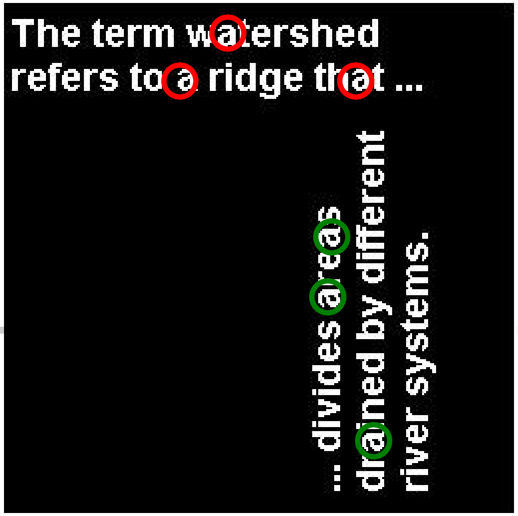


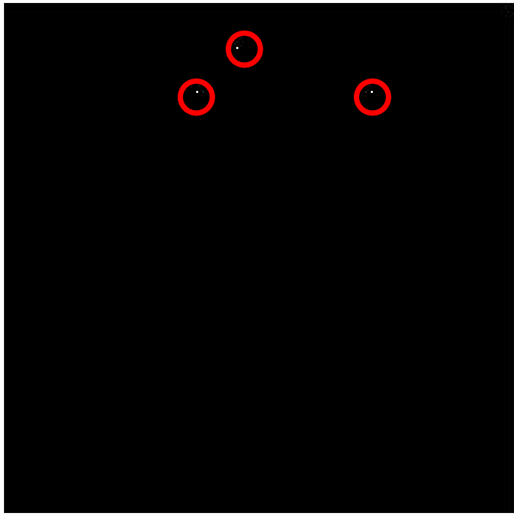
Ideal BPF*

Original Image

Gaussian BPF*

DFT – Matlab demo





```
clear all;close all;
iptsetpref('ImshowBorder','tight');
bw = imread('text.png');
imshow(bw);
a = bw(32:45,88:98);
figure, imshow(a);
C = real(ifft2(fft2(bw) .* fft2(rot90(a,2),256,256)));
figure, imshow(C,[])
max(C(:));  %cys this call output 68
thresh = 60; % Set a threshold a little less than max.
figure, imshow(C > thresh); % highlight pixels over threshold.
```

Homework VII

- Design your own DFT exploration experimence. It can be either of the following
  - DFT and reconstruction
  - DFT Real/Imagenary part, magnitude/phase
  - Frequency domain filtering
  - DFT based image analysis (Detection/Editing…)
- Submit your test images, codes, experiment results and documents.

End of Lecture