Image & Video Processing

Frequency Domain Filtering

Contents

In this lecture we will look at DFT and image enhancement in the frequency domain

- Discrete Fourier Transform (2D)
- DFT and Images
- DFT Properties
- Image Processing in the frequency domain
- Frequency domain filters for enhancement
    - Image smoothing
    - Image sharpening

The Discrete Fourier Transform in 2D

The *Discrete Fourier Transform* of $f(x, y)$, for x = 0, 1, 2…M-1 and y = 0,1,2…N-1, denoted by $F(u, v)$, is given by the equation:

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

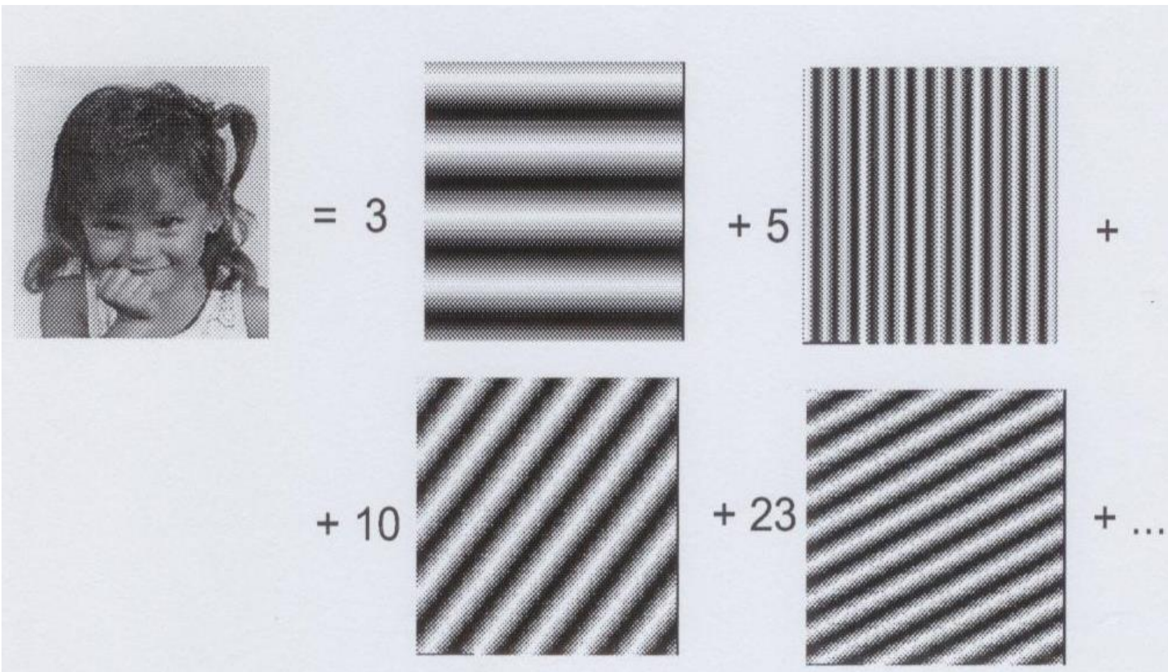
for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

Inverse Discrete Fourier Transform is given by equation:

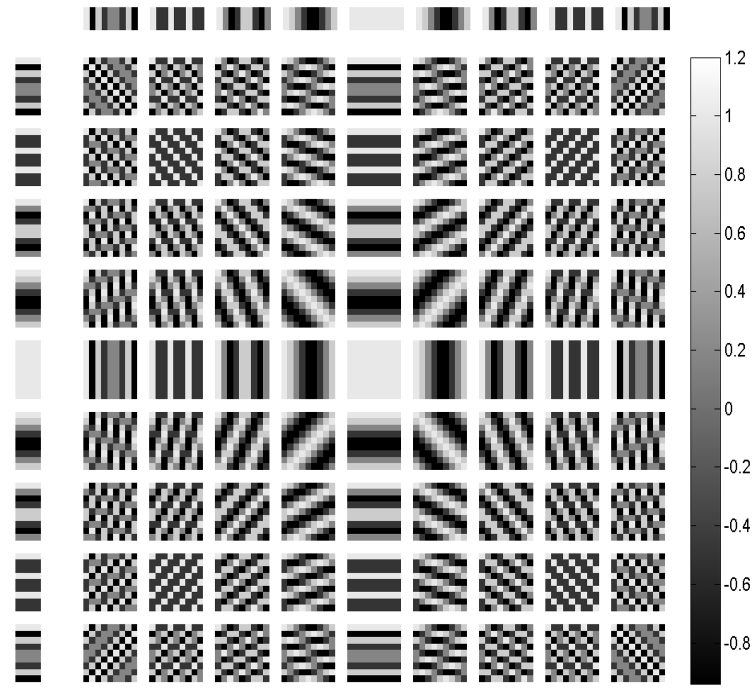
$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

DFT & Images

Interpretation: 2D cos/sin functions

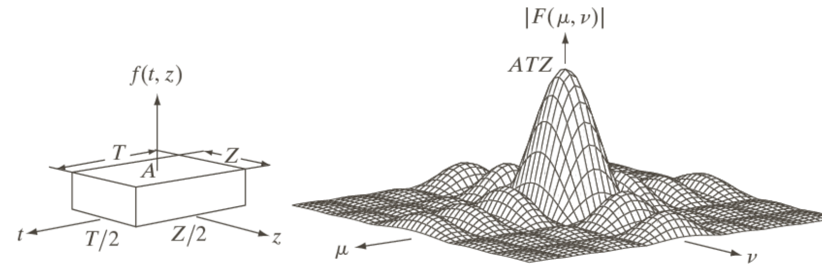


2-D Fourier basis

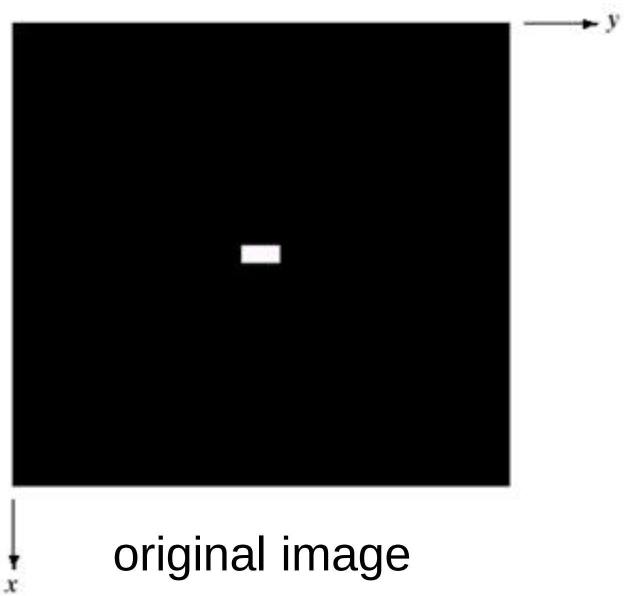


Visualizing DFT

- Typically, we visualize |F(u,v)|
- The dynamic range of |F(u,v)| is typically very large

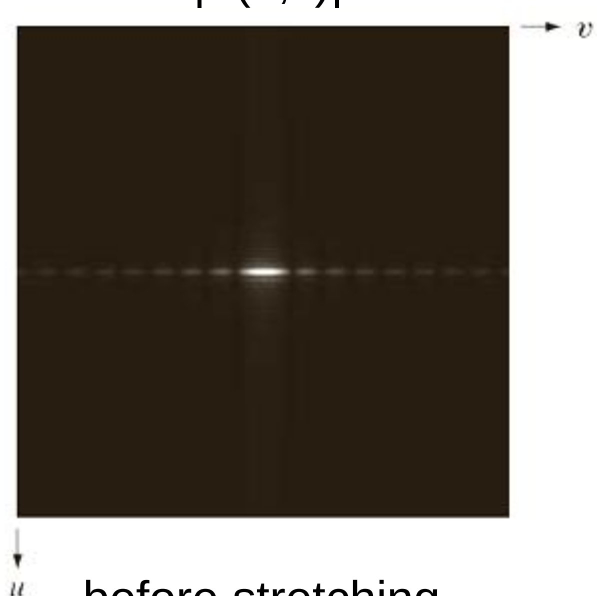


- Apply log transformation: $D(u, v) = c \ log(1 + |F(u, v)|)$



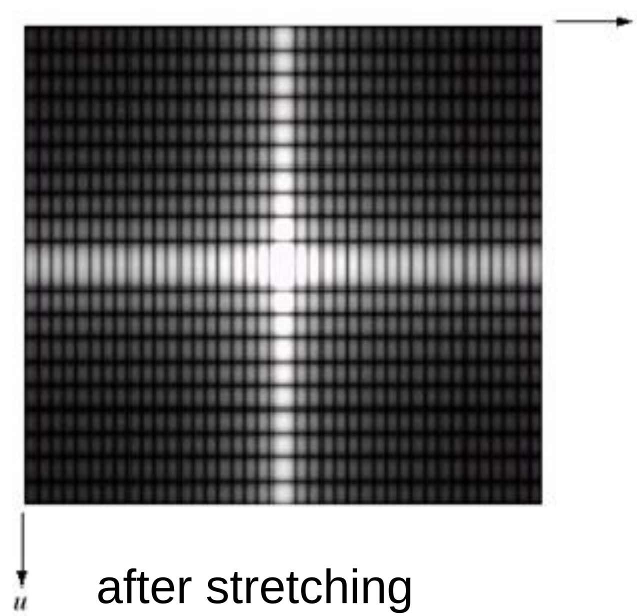
original image

|F(u,v)|



before stretching

|D(u,v)|



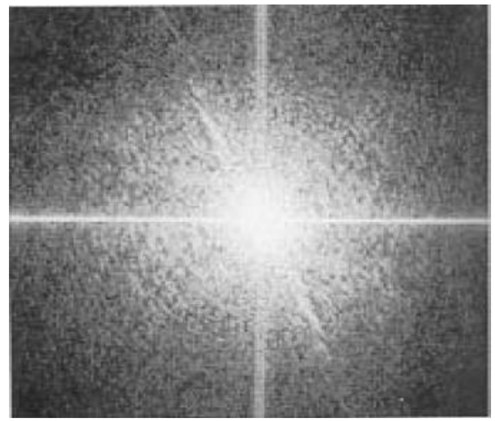
after stretching

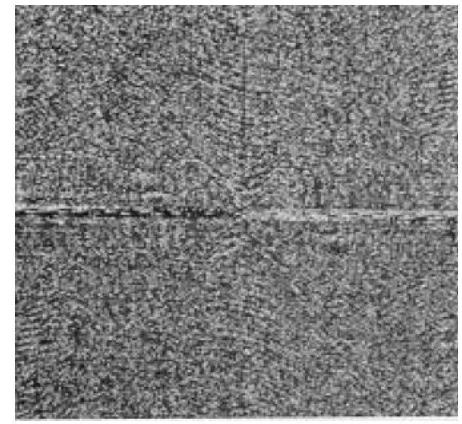
Magnitude and Phase of DFT

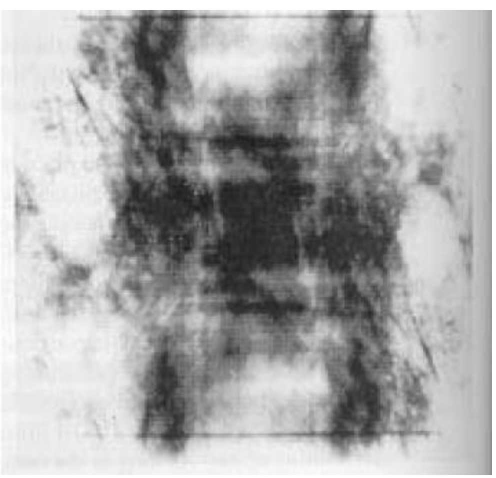
- What is more important?

magnitude

phase







Reconstructed image using magnitude only

Reconstructed image using Phase only

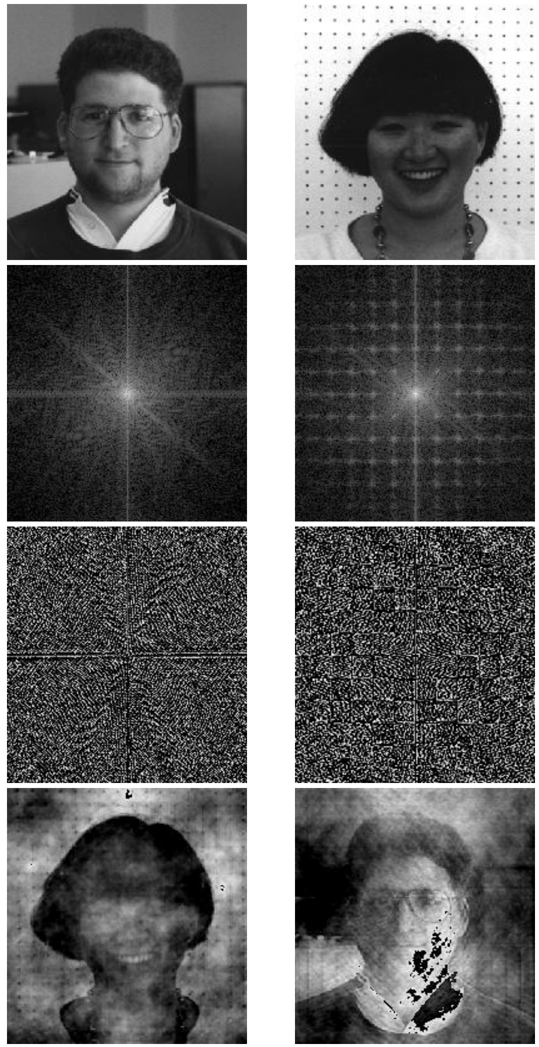
another example: amplitude vs. phase

A = “Aron”

FA = fft2(A)

P = “Phyllis”

FP = fft2(P)



log(abs(FA))

log(abs(FP))

angle(FA)

angle(FP)

ifft2(abs(FA), angle(FP))

ifft2(abs(FP), angle(FA))

DFT Properties: Separability

- The 2D DFT can be computed using 1D transforms **only**:

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

kernel is separable: $$e^{-j2\pi(\frac{ux+vy}{N})}=e^{-j2\pi(\frac{ux}{N})}e^{-j2\pi(\frac{vy}{N})}$$

- Let's set: $$\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{vy}{N})}=F(x,v)$$

- Then: $$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1} e^{-j2\pi(\frac{ux}{N})}F(x,v)$$

Fast Fourier Transform

- The development of the *Fast Fourier Transform (FFT)* algorithm allows the Fourier transform to be carried out in a reasonable amount of time

- Based on Divide and conquer strategy and exploits property (like Separability) of the 2D transform

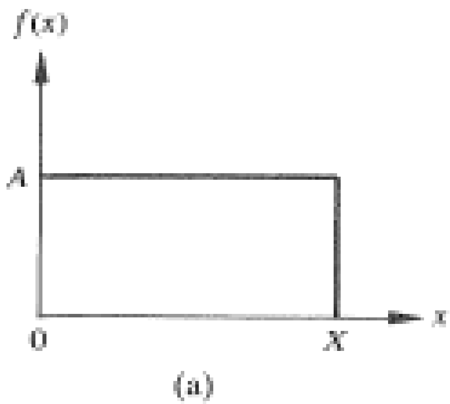
- Reduces the amount of time required to perform a Fourier transform by a factor of ***MN log MN*** times

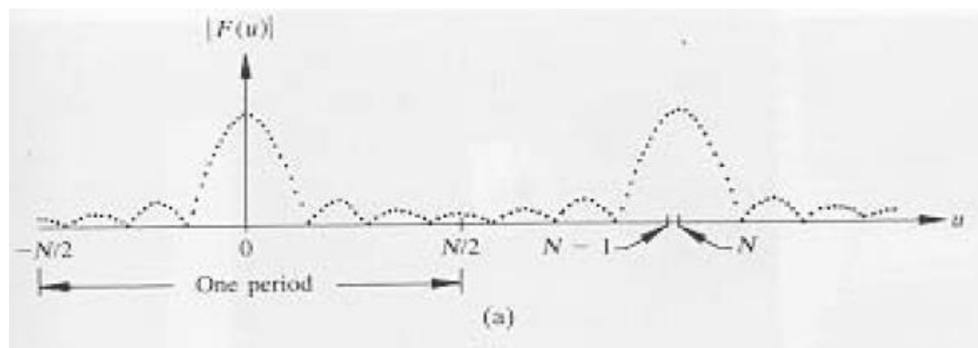
- Discrete, 2-D Fourier & inverse Fast Fourier transforms are implemented in Matlab as `fft2` and `ifft2`

DFT Properties: Periodicity

- The DFT and its inverse are periodic with period N

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





DFT Properties: Translation

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

- Translation in spatial domain:

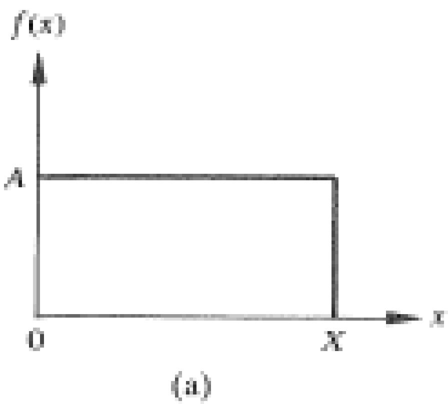
$$f(x-x_0, y-y_0) \longleftrightarrow F(u,v)e^{-j2\pi(\frac{ux_0+vy_0}{N})}$$

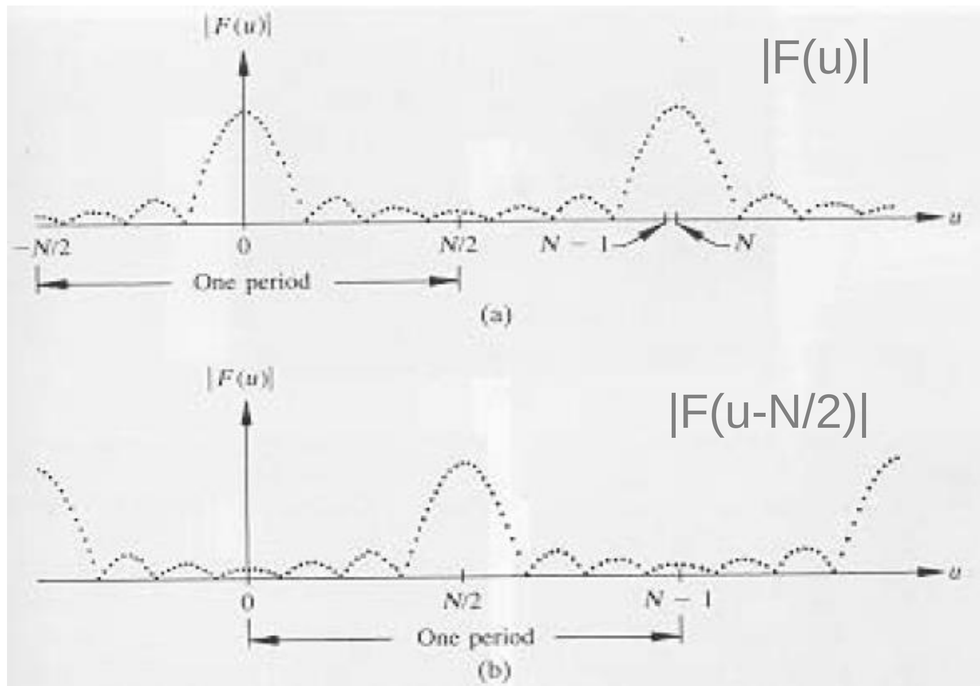
- Translation in frequency domain:

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

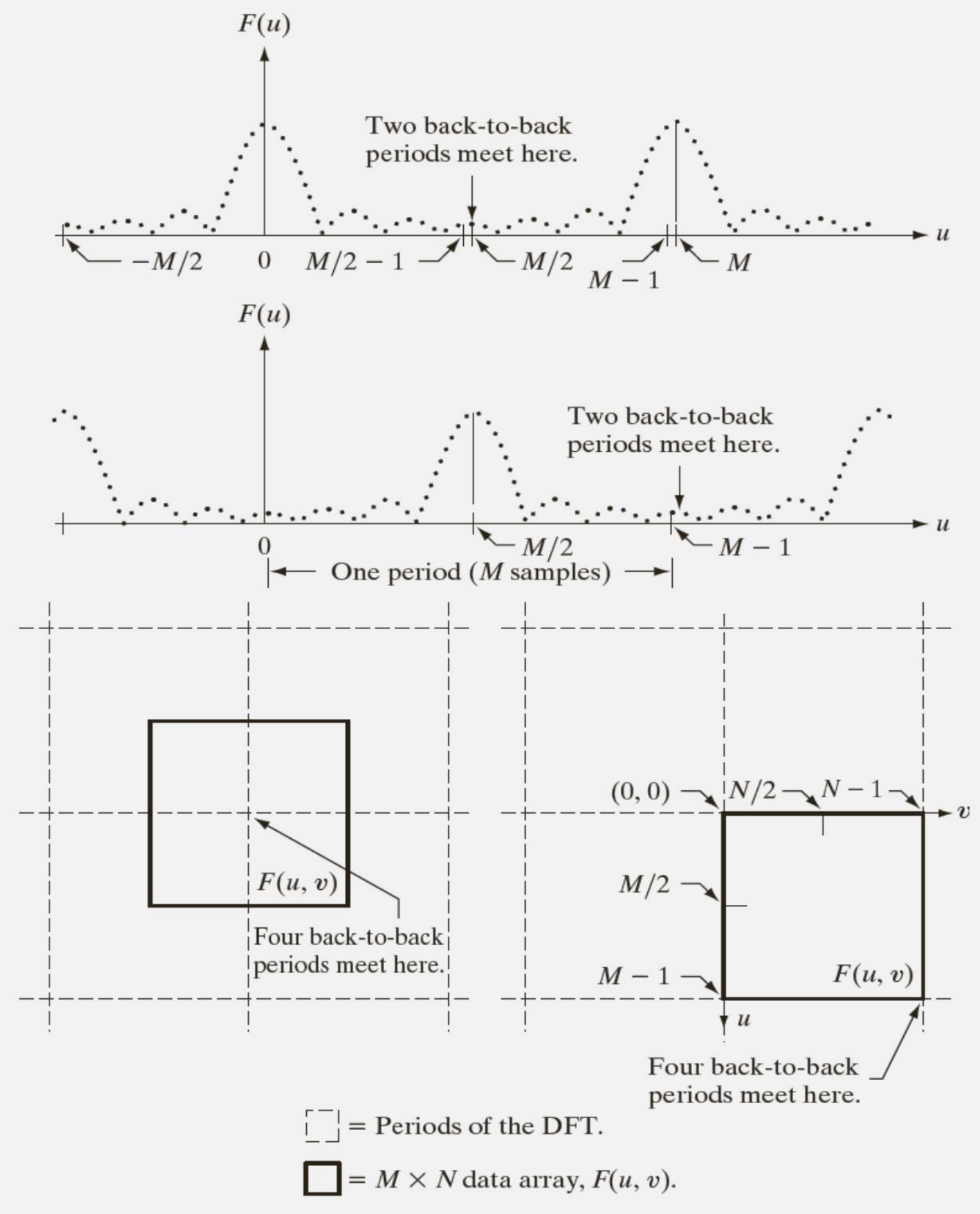
DFT Properties: Translation (cont'd)

- To show a full period, we need to translate the origin of the transform at **u=N/2** (or at **(N/2,N/2)** in 2D)





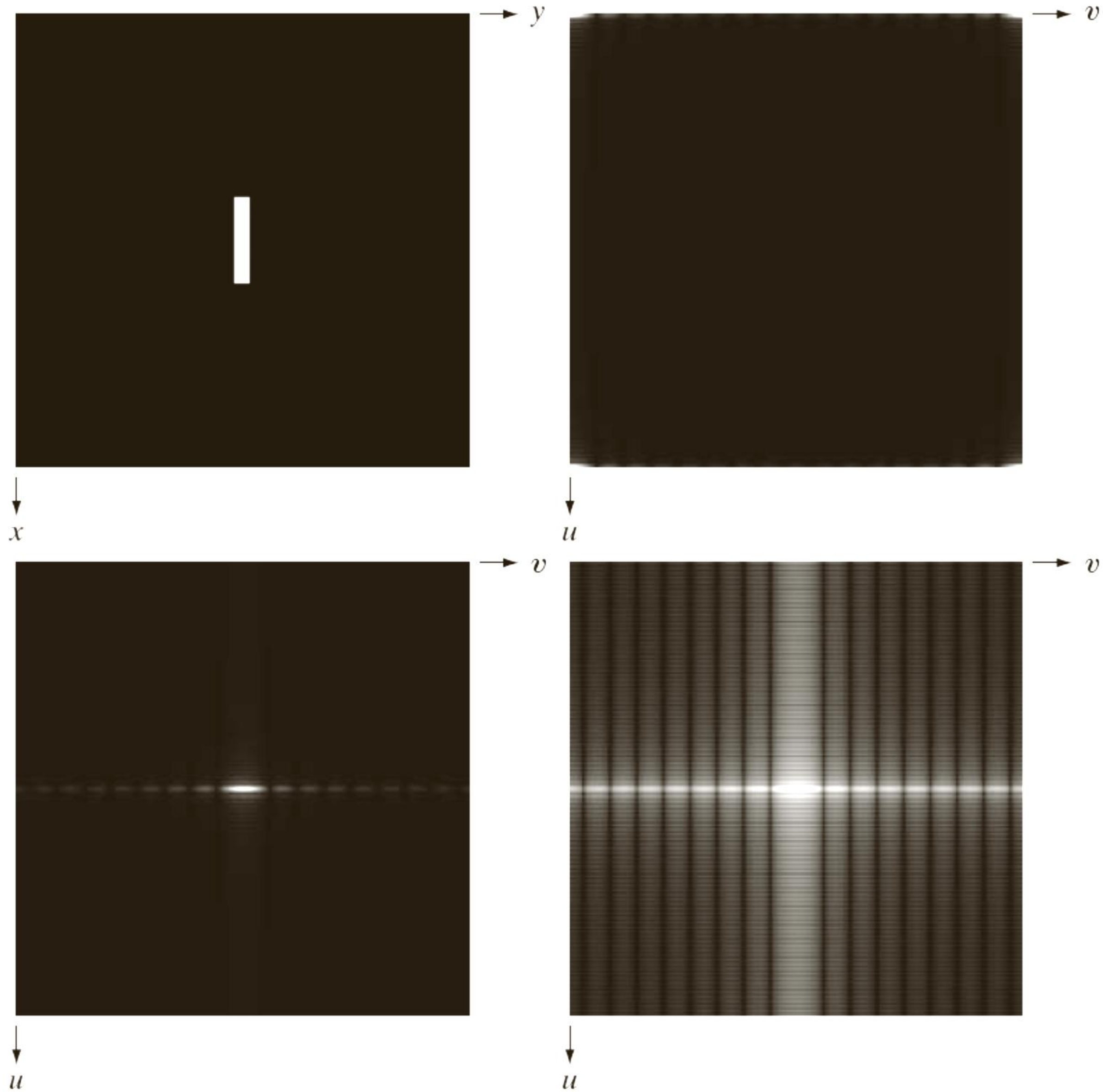
Periodicity & Translation of the transform



a
b
c d

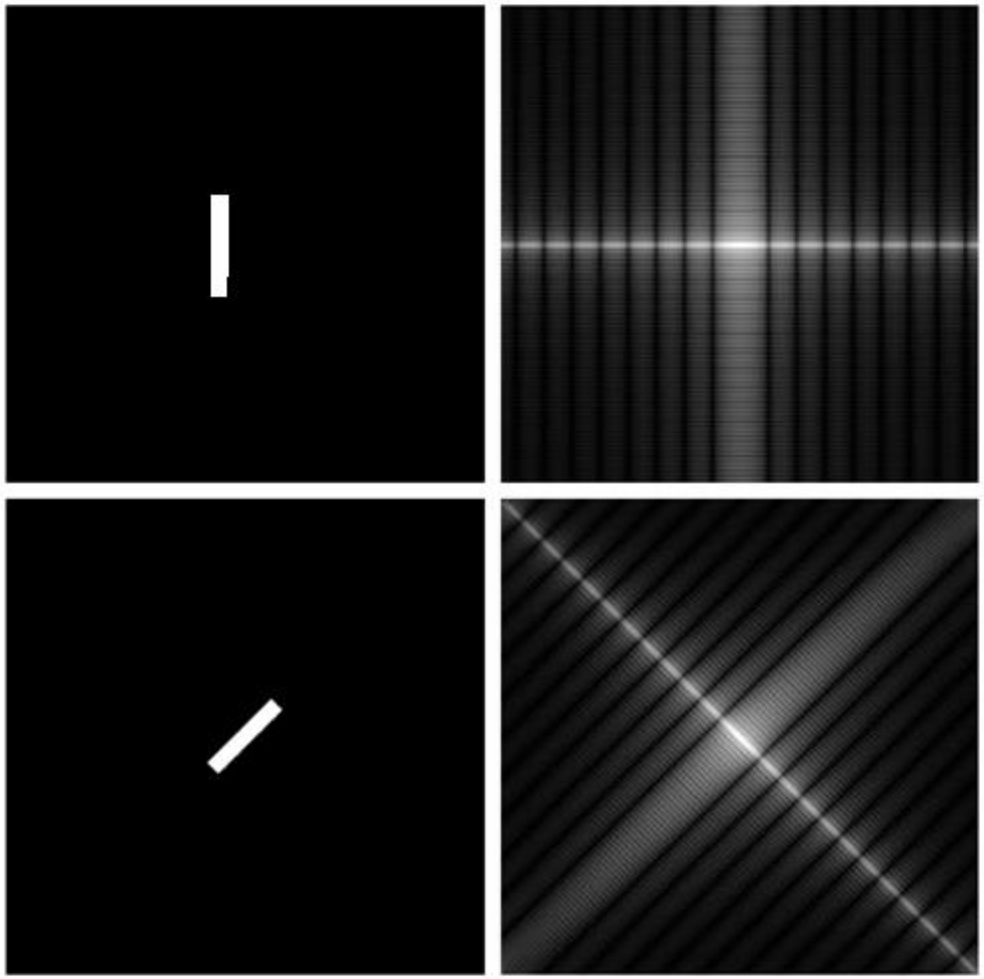
FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained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

Centering the spectrum and log enhancement



DFT Properties: Rotation

- Rotating f(x,y) by θ rotates F(u,v) by θ



DFT Properties: Multiplication and Convolution

Spatial Domain *(x)*		Frequency Domain *(u)*
$g = f * h$	$\longleftrightarrow$	$G = FH$
$g = fh$	$\longleftrightarrow$	$G = F * H$

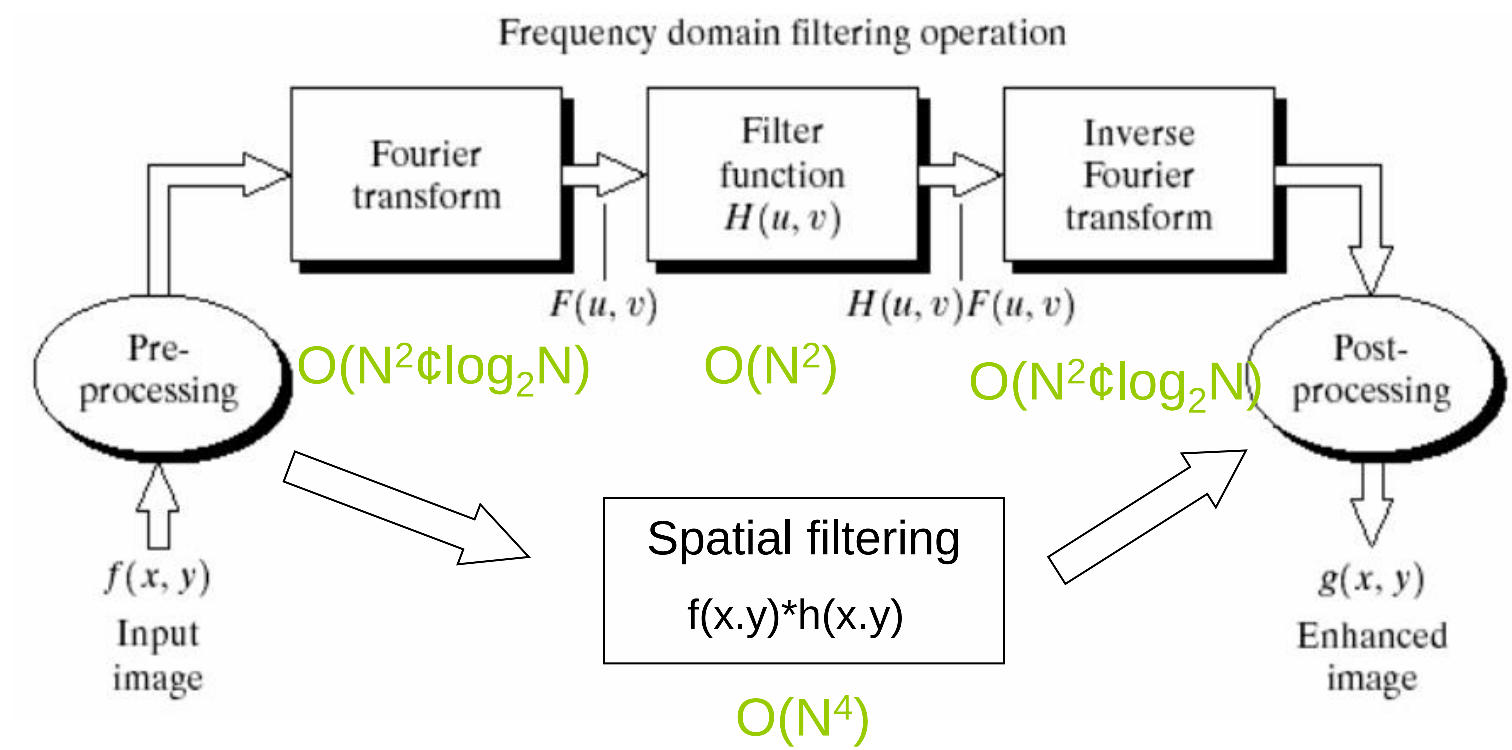
So, we can find *g(x)* by Fourier transform

$$\begin{tikzcd}
g & = & f \arrow[d, "\mathbf{FT}"] & * & h \arrow[d, "\mathbf{FT}"] \\
G \arrow[u, "\mathbf{IFT}"] & = & F & \times & H
\end{tikzcd}$$

Filtering Image in Frequency Domain

To filter an image in the frequency domain:

1. Compute *F(u,v)* the DFT of the image
2. Multiply *F(u,v)* by a filter function *H(u,v)*
3. Compute the inverse DFT of the result



DFT Properties (contd.)

$$F[f(x,y)+g(x,y)] = F[f(x,y)] + F[g(x,y)]$$ (Addition)

$$af(x,y) <--> aF(u,v)$$ (Scale)

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

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

$$|F(u,v)| = |F(-u,-v)|$$ (symmetric)

$$\frac{\partial^n f(x,y)}{\partial x^n} \Leftrightarrow (ju)^n F(u,v)$$

$$(-jx)^n f(x,y) \Leftrightarrow \frac{\partial^n F(u,v)}{\partial u^n}$$

(Differentiation)

IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

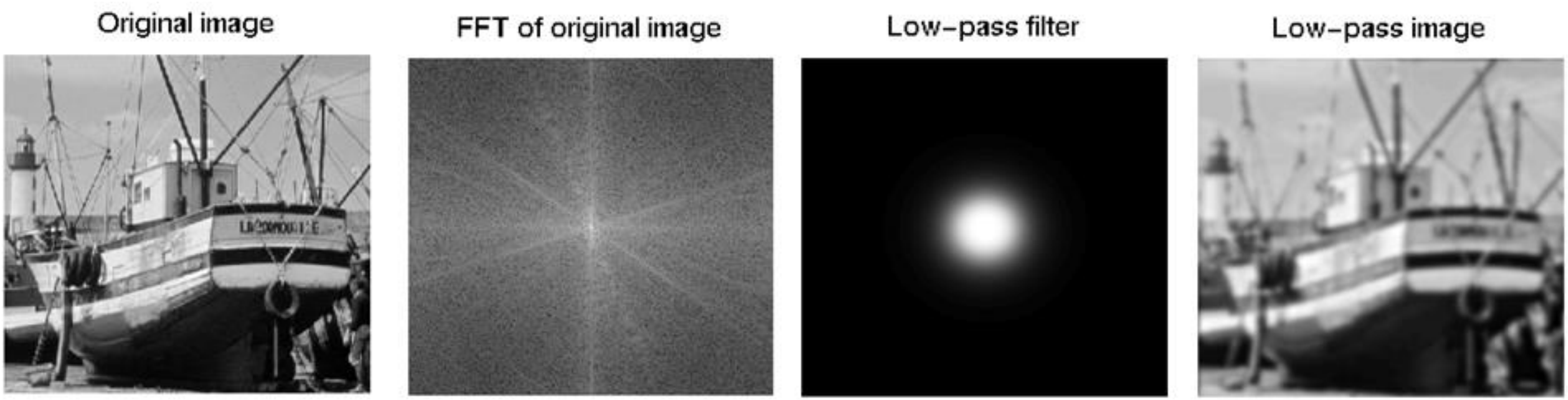
Image Smoothing

Image Sharpening

Smoothing: Averaging / Lowpass Filtering

Smoothing/Blurring results from:

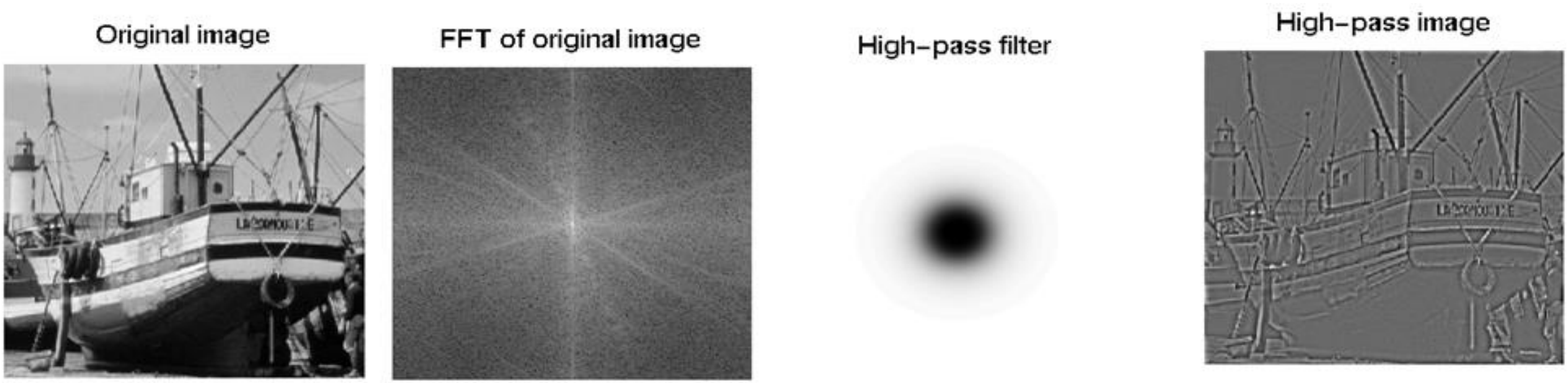
- Pixel averaging in the spatial domain:
  - Each pixel in the output is a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 1.
- Lowpass filtering in the frequency domain:
  - High frequencies are diminished or eliminated



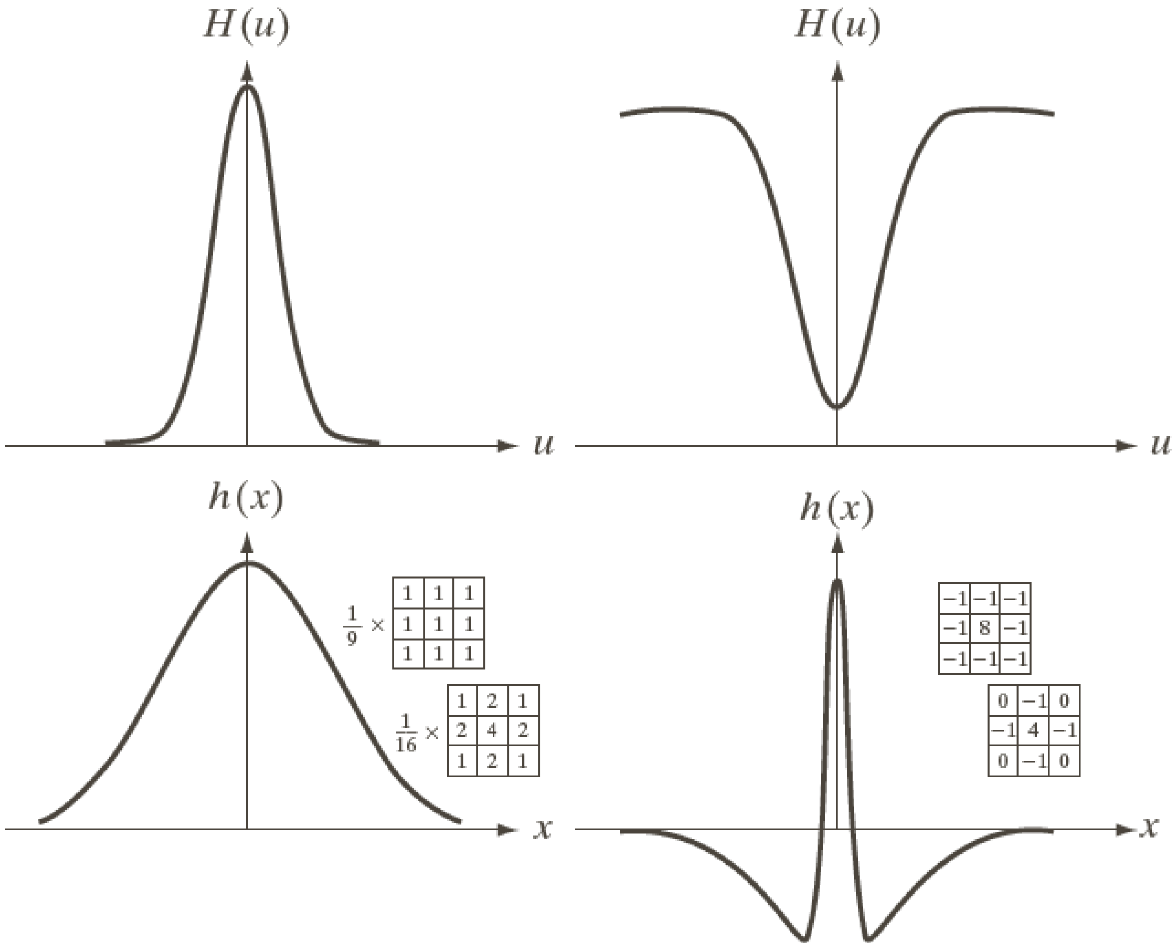
Sharpening: Differencing / Highpass Filtering

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.



$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

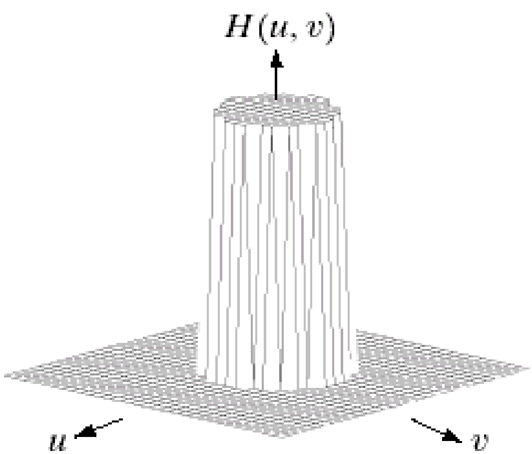


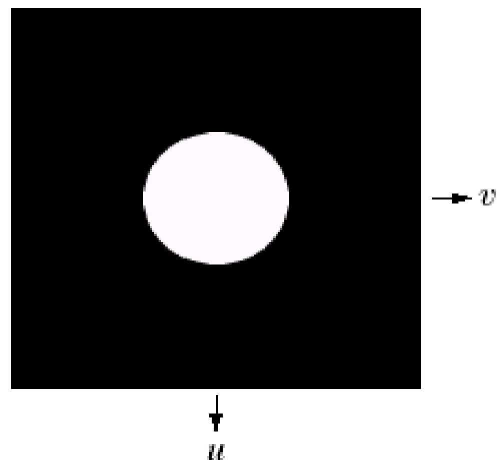
a c
b d

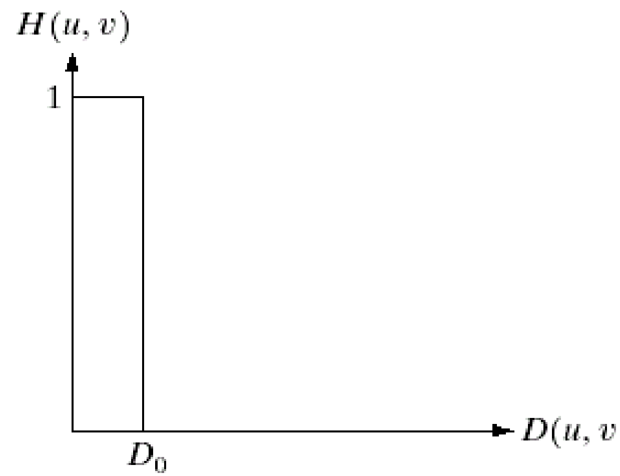
FIGURE 4.37 (a) A 1-D Gaussian lowpass filter in the frequency domain. (b) Spatial lowpass filter corresponding to (a). (c) Gaussian highpass filter in the frequency domain. (d) Spatial highpass filter corresponding to (c). The small 2-D masks shown are spatial filters we used in Chapter 3.

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin of the transform







changing the distance changes the behaviour of the filter

Ideal Low Pass Filter (cont…)

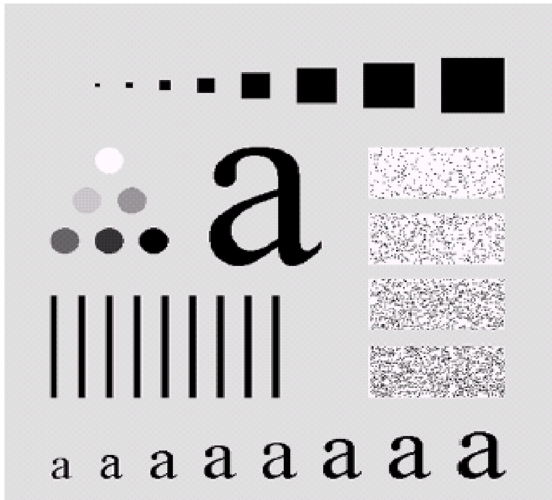
The transfer function for the ideal low pass filter can be given as:

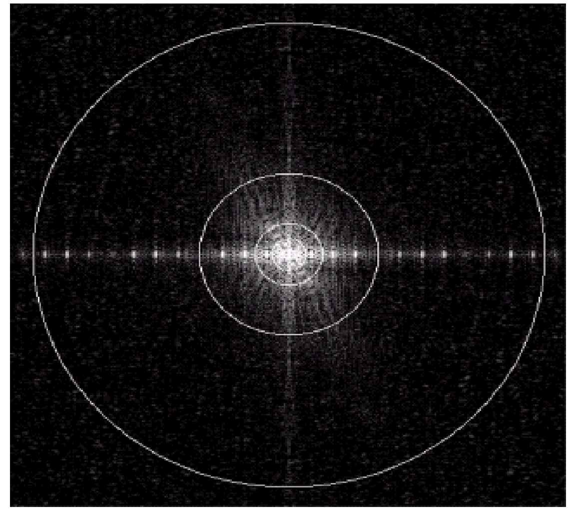
$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \leq D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$

where *D(u,v)* is given as:

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

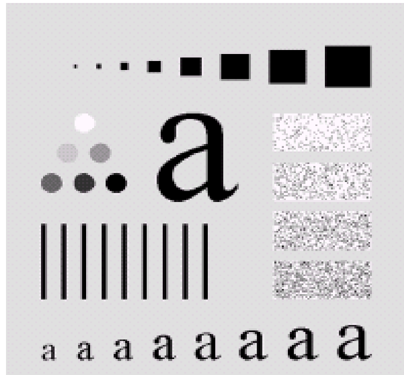
Ideal Low Pass Filter (cont…)



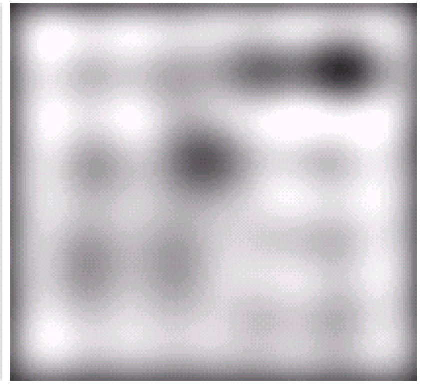


Above we show an image, it's Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it

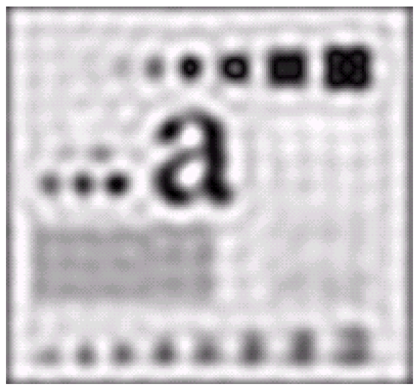
Ideal Low Pass Filter (cont…)



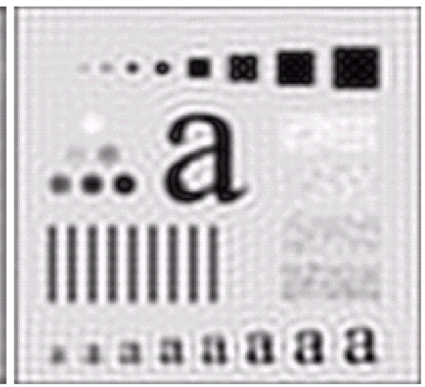
Original image



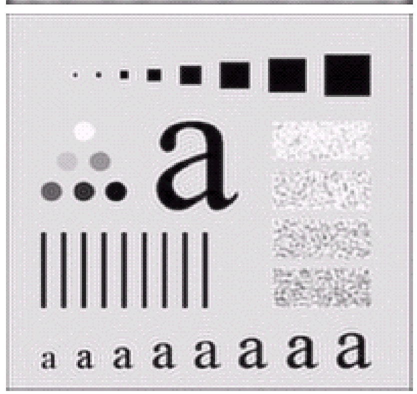
Result of filtering with ideal low pass filter of radius 5



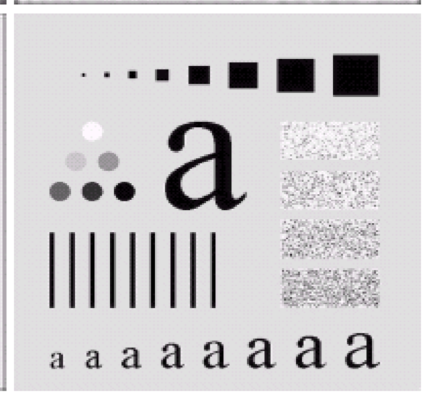
Result of filtering with ideal low pass filter of radius 15



Result of filtering with ideal low pass filter of radius 30



Result of filtering with ideal low pass filter of radius 80



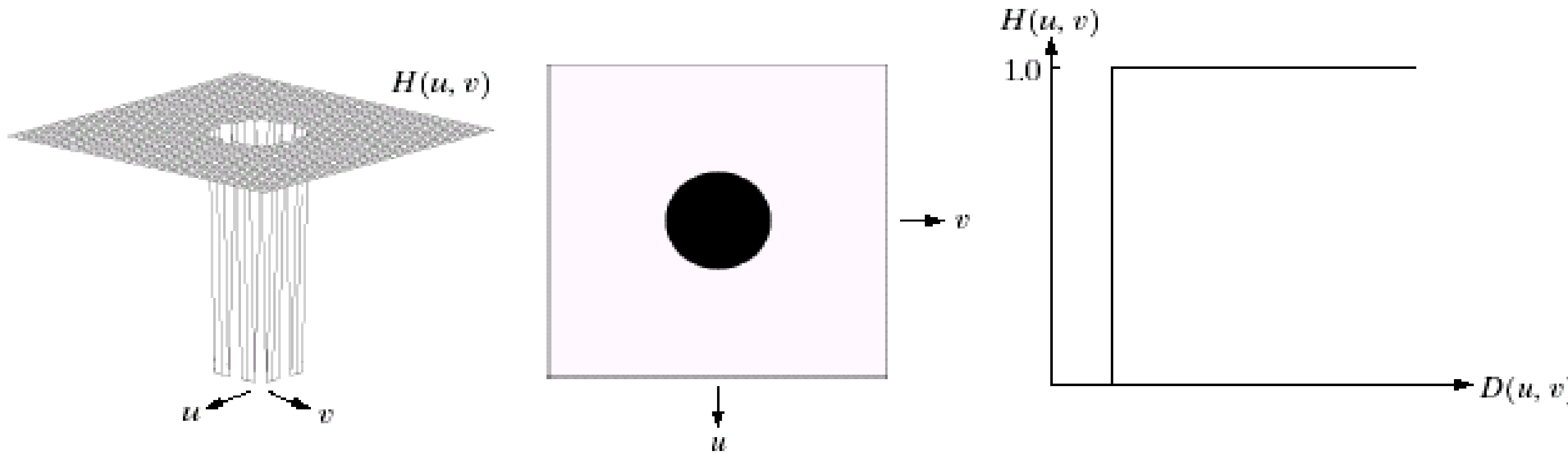
Result of filtering with ideal low pass filter of radius 230

Ideal High Pass Filters

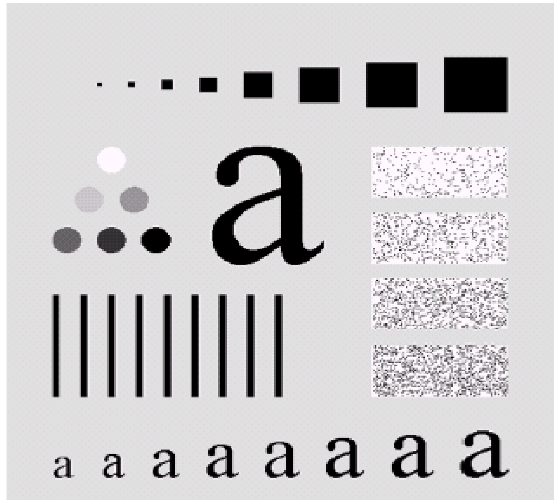
The ideal high pass filter is given as:

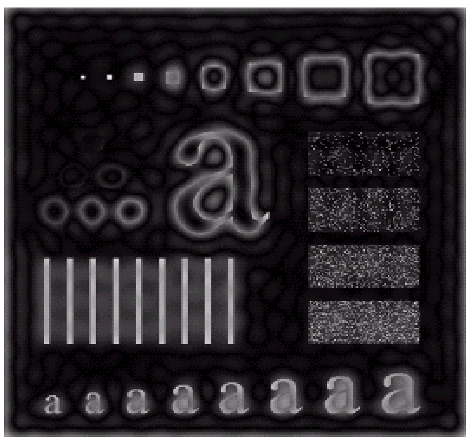
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 is the cut off distance as before

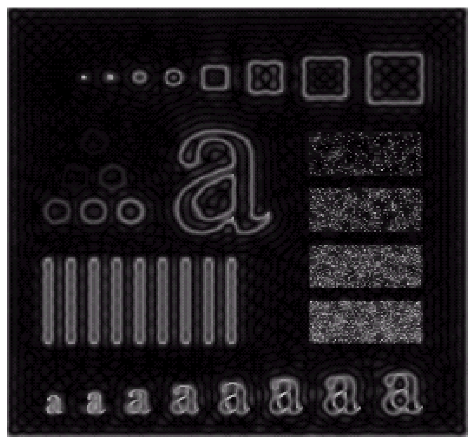


Ideal High Pass Filters (cont…)

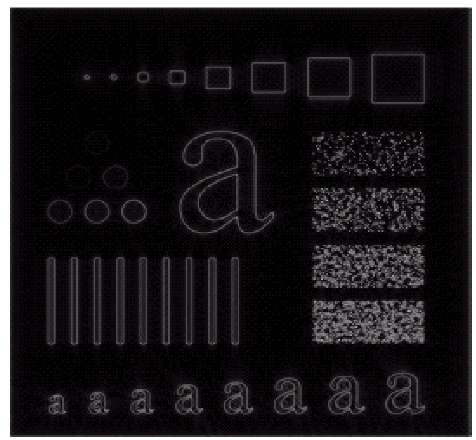




Results of ideal high pass filtering with $D_0 = 15$



Results of ideal high pass filtering with $D_0 = 30$



Results of ideal high pass filtering with $D_0 = 80$

Ideal Band Pass Filters

The ideal band pass filter is given as:

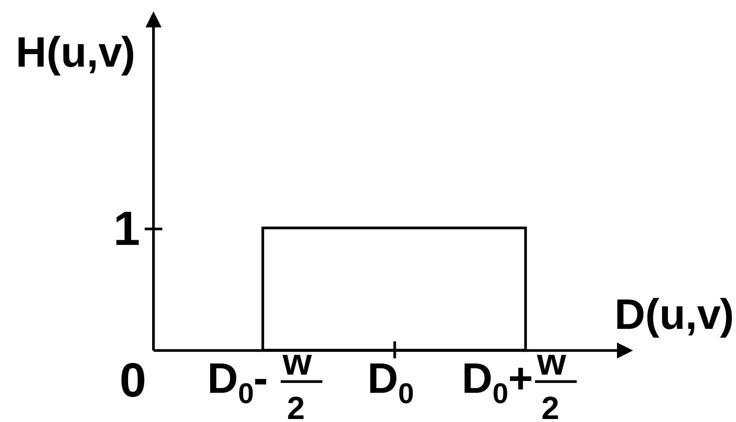
$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 - \frac{w}{2} \\ 1 & D_0 - \frac{w}{2} \le D(u,v) \le D_0 + \frac{w}{2} \\ 0 & D(u,v) > D_0 + \frac{w}{2} \end{cases}$$

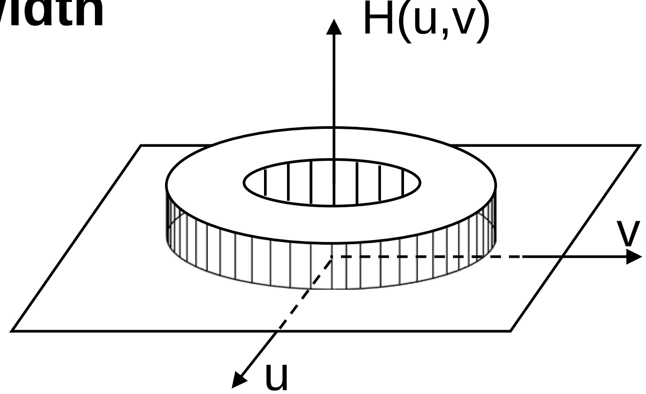
where

$$D(u,v) = \sqrt{u^2 + v^2}$$

D_0 = cut off frequency

w = band width

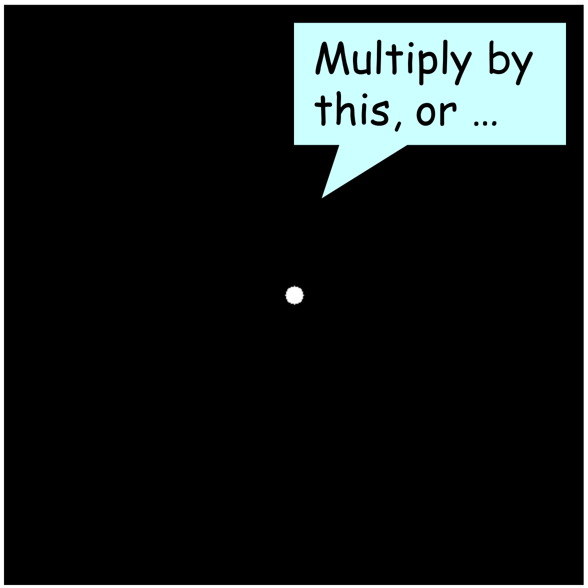


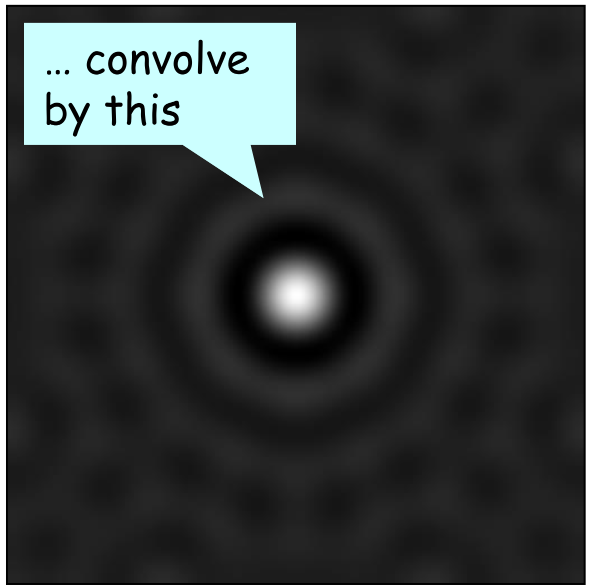


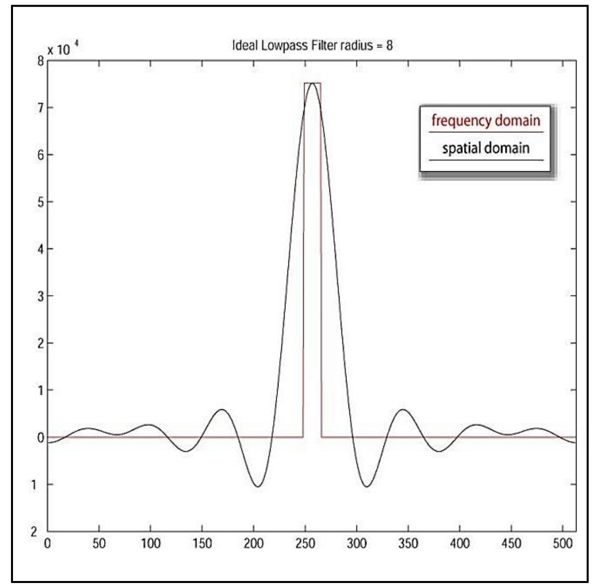
The Ringing problem

Ideal Filters Do Not Produce Ideal Results

Image size: 512x512
FD filter radius: 8







Fourier Domain Rep.

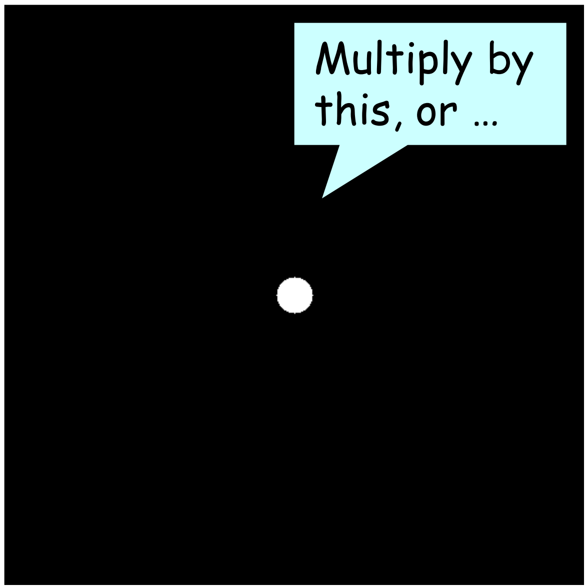
Spatial Representation

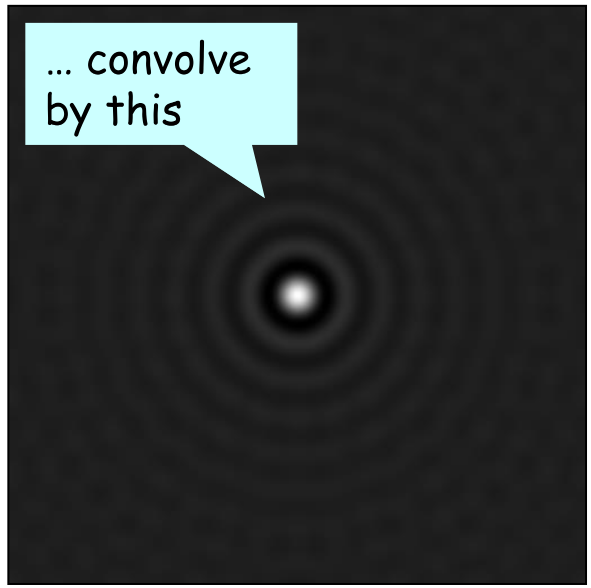
Central Profile

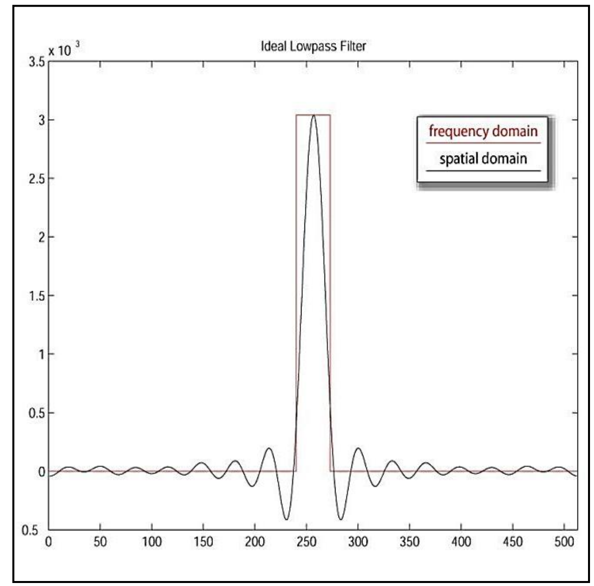
The Ringing problem

Ideal Filters Do Not Produce Ideal Results

Image size: 512x512
FD filter radius: 16







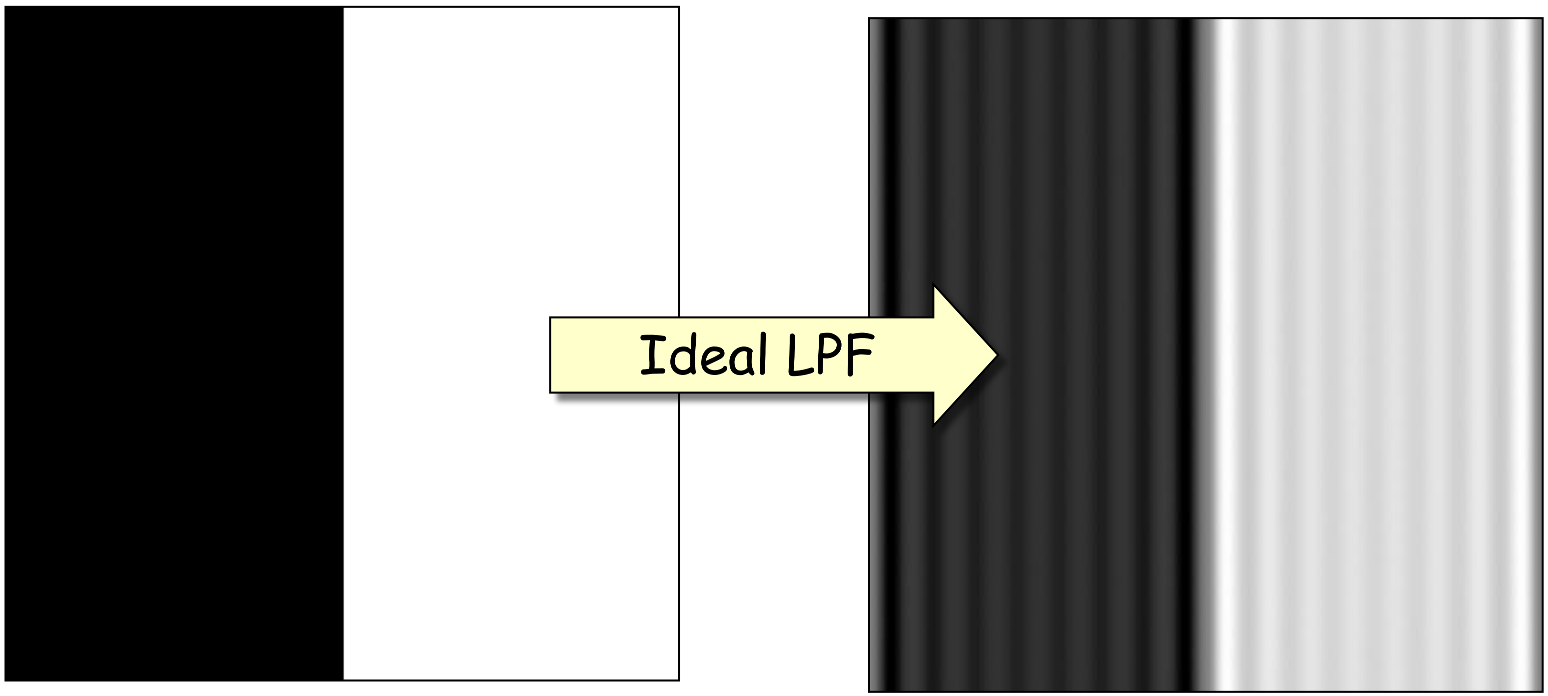
Fourier Domain Rep.

Spatial Representation

Central Profile

$\uparrow D_0 \longrightarrow \downarrow$ Ringing radius + $\downarrow$ blur

The Ringing problem

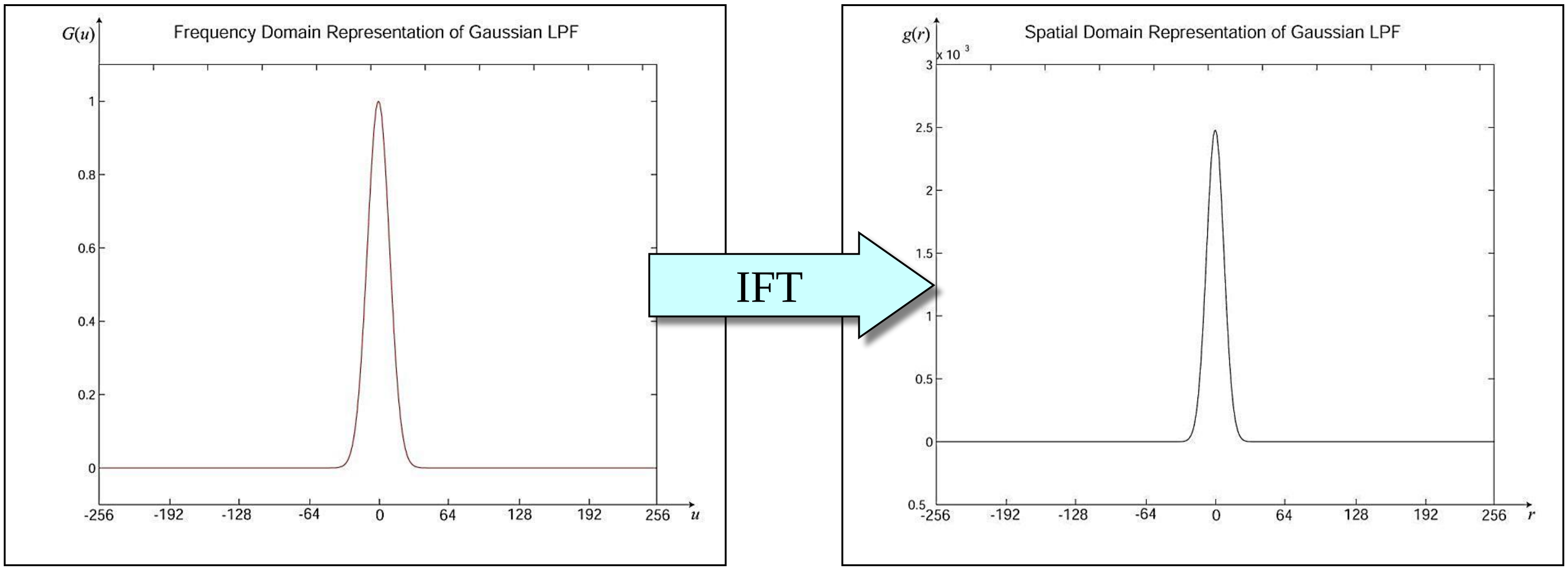


Blurring the image above w/ an ideal lowpass filter…

…distorts the results with ringing or ghosting.

Optimal Filter

The Gaussian

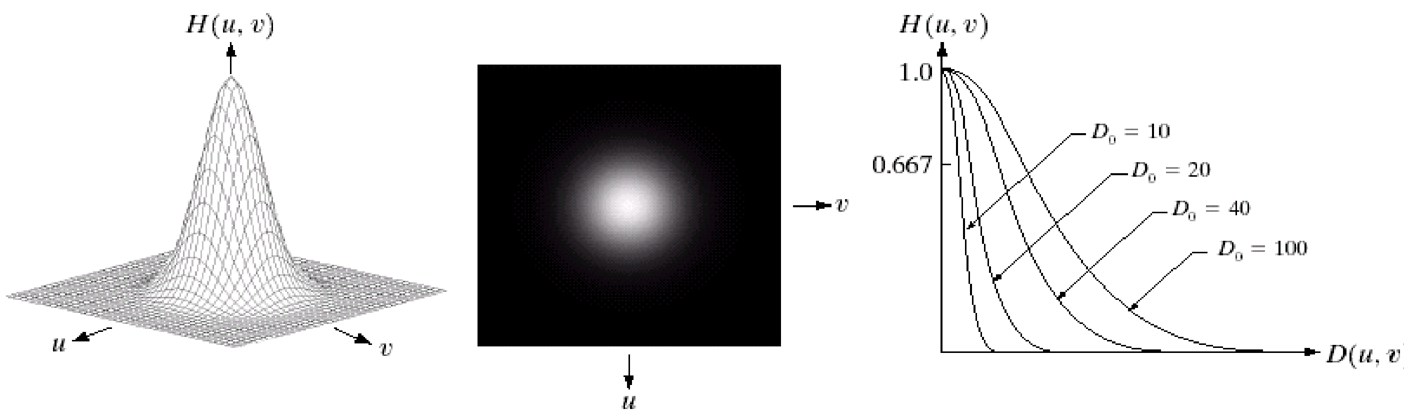


The Gaussian filter optimizes it by providing the sharpest cutoff possible without ringing.

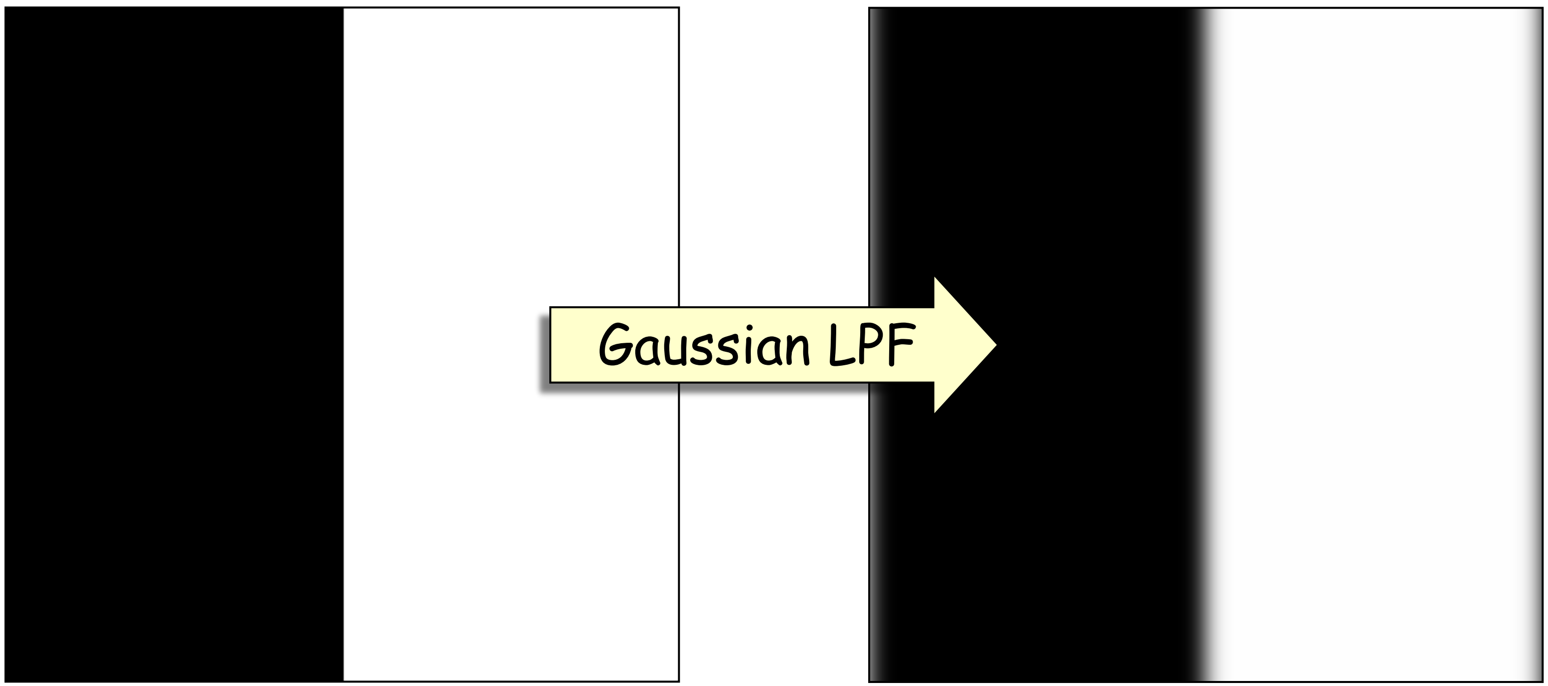
Gaussian Lowpass Filters

The transfer function of a Gaussian lowpass filter is defined as:

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



Optimal Filter: The Gaussian

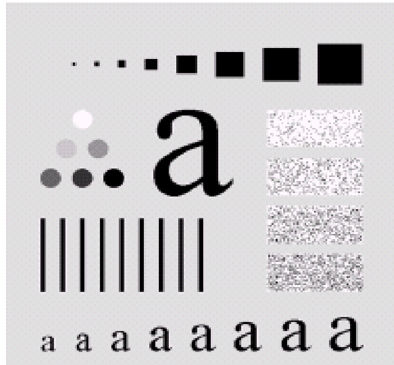


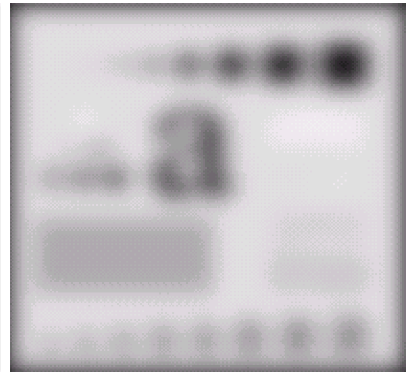
With a gaussian lowpass filter, the image above …

… is blurred without ringing or ghosting.

Gaussian Lowpass Filters (cont…)

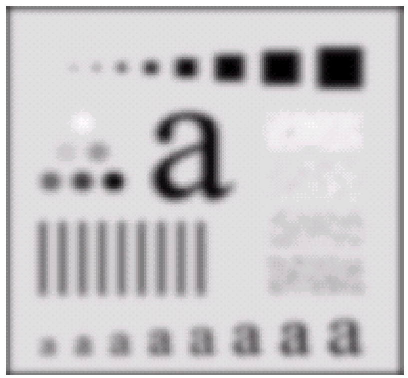
Original image

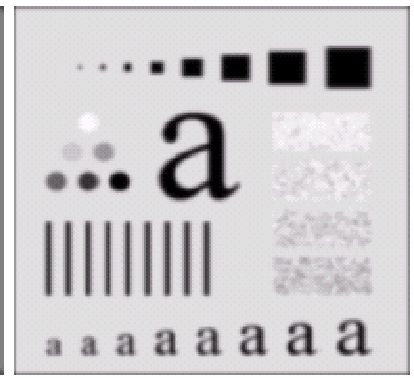




Result of filtering with Gaussian filter with cutoff radius 5

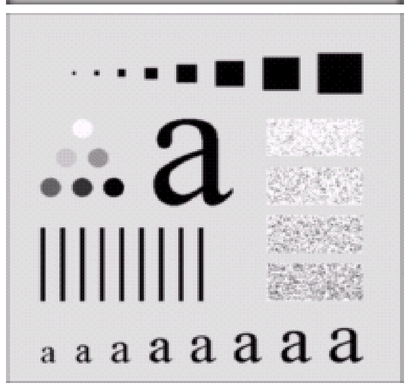
Result of filtering with Gaussian filter with cutoff radius 15

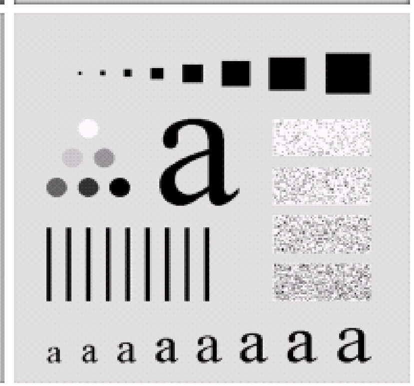




Result of filtering with Gaussian filter with cutoff radius 30

Result of filtering with Gaussian filter with cutoff radius 85





Result of filtering with Gaussian filter with cutoff radius 230

Resolution Sequence

Original Image

$\sigma_0 = 0$

Resolution Sequence

Gaussian
LPF

$\sigma_1 = 1$

Resolution Sequence

Gaussian
LPF
$\sigma_2 = 2$

Resolution Sequence

Gaussian LPF

$\sigma_3 = 4$

Resolution Sequence

Gaussian
LPF

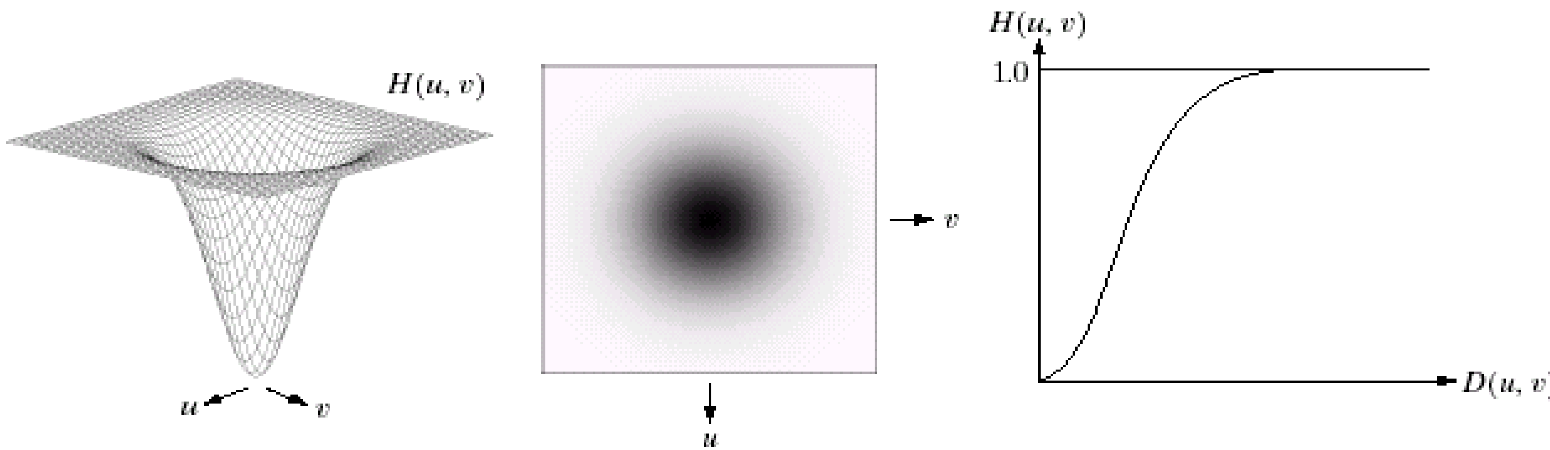
$\sigma_4 = 8$

Gaussian High Pass Filters

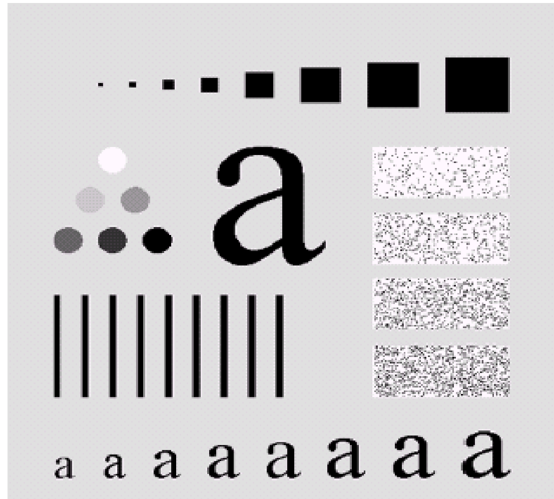
The Gaussian high pass filter is given as:

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

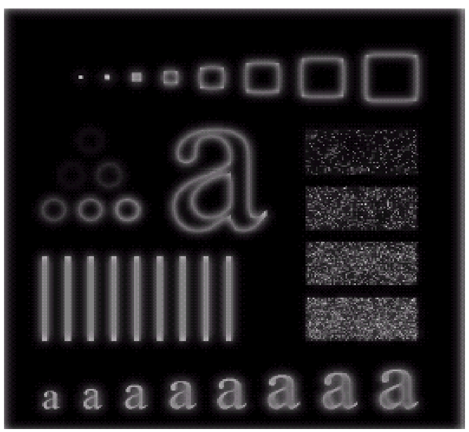
where D_0 is the cut off distance as before

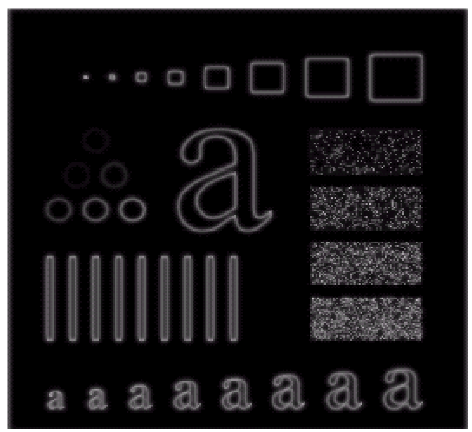


Gaussian High Pass Filters (cont…)



Results of Gaussian high pass filtering with $D_0 = 15$





Results of Gaussian high pass filtering with $D_0 = 30$

Results of Gaussian high pass filtering with $D_0 = 80$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_9 = 256$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_9\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_8 = 128$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_8\right) \right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_7 = 64$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_7\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_6 = 32$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_6\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_5 = 16$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_5\right)\right].$$

Sharpening above a Specific Scale.

An image is sharpened by taking a linear combination of the image and a highpass filtered version of itself. The scale of the sharpening can be controlled via the cutoff of the HPF and a multiplicative constant. In the following examples the image has been sharpened via

$$\mathbf{I}_{\mathrm{hfe},\sigma} = \mathbf{I} + \alpha \mathbf{I}_{\mathrm{hpf},\sigma} = \mathbf{I} + \alpha\Big(\mathbf{I} - \big[\mathbf{I} * g(\sigma)\big]\Big) = (1+\alpha)\mathbf{I} - \alpha\big[\mathbf{I} * g(\sigma)\big],$$

where g is a 2D Gaussian with $\sigma \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ and α is a scale factor, usually in (0,2). After the computation, each image was histogram matched to **I**.

$\sigma_0 = 0$ Original Image

Sharpening above a Specific Scale

$\sigma_0 = 1$, $\alpha=1$

Sharpening above a Specific Scale

$\sigma_0 = 2, \alpha=1$

Sharpening above a Specific Scale

$\sigma_0 = 4, \alpha=1$

Sharpening above a Specific Scale

$\sigma_0 = 8, \alpha = 1$

Sharpening above a Specific Scale

$\sigma_0 = 16, \alpha=1$

Sharpening above a Specific Scale

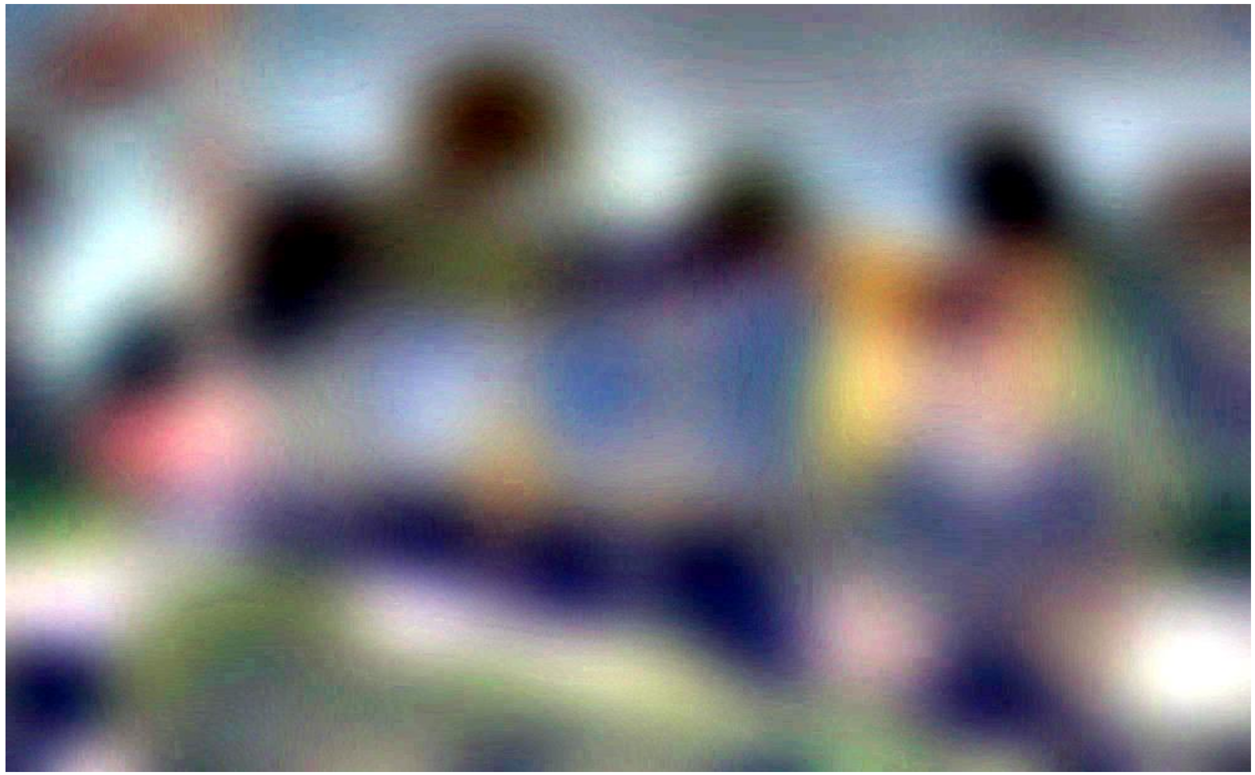
$\sigma_0 = 32, \alpha=1$

Sharpening above a Specific Scale

$\sigma_0 = 64$, $\alpha=1$

Sharpening above a Specific Scale

Bandpass Sequence



Difference between Gaussian LPF images: $\sigma_6 = 32$ and $\sigma_7 = 64$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_6\right) - g\left(\sigma_7\right) \right] = \left[\mathbf{I} * g\left(\sigma_6\right) \right] - \left[\mathbf{I} * g\left(\sigma_7\right) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_5 = 16$ and $\sigma_6 = 32$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_5\right) - g\left(\sigma_6\right) \right] = \left[\mathbf{I} * g\left(\sigma_5\right) \right] - \left[\mathbf{I} * g\left(\sigma_6\right) \right].$$

Bandpass Sequence

Difference between Gaussian LPF images: $\sigma_4 = 8$ and $\sigma_5 = 16$.

$$\mathbf{J} = \mathbf{I} * \left[g\left(\sigma_4\right) - g\left(\sigma_5\right) \right] = \left[\mathbf{I} * g\left(\sigma_4\right) \right] - \left[\mathbf{I} * g\left(\sigma_5\right) \right].$$

Emphasizing a Specific Pass Band.

An image can be bandpass filtered by subtracting two differently Gaussian filtered copies of it. That specific band can be emphasized in the image by adding it back to the image. In the following examples the image has been emphasized via

$$
\begin{aligned}
\mathbf{I}_{\mathrm{bpe},\sigma_0,\sigma_1} &= \mathbf{I} + \alpha \mathbf{I}_{\mathrm{bpf},\sigma_0,\sigma_1} \\
&= \mathbf{I} + \alpha \Big[\mathbf{I} - \Big(\big[\mathbf{I} * g(\sigma_0) \big] - \big[\mathbf{I} * g(\sigma_1) \big] \Big) \Big] \\
&= (1+\alpha)\,\mathbf{I} - \alpha \Big(\mathbf{I} * \big[g(\sigma_0) - g(\sigma_1) \big] \Big),
\end{aligned}
$$

where $\sigma_0, \sigma_1 \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ and α is a scale factor, usually in (0,2). After the computation, each image was histogram matched to **I**.

$(\sigma_1, \sigma_0) = (1,0)$, $\alpha=1$

Emphasizing a Specific Pass Band

$(\sigma_1, \sigma_0) = (2,1),\ \alpha = 1$

Emphasizing a Specific Pass Band

$(\sigma_1, \sigma_0) = (4,2)$, $\alpha=1$

Emphasizing a Specific Pass Band

$(\sigma_1, \sigma_0) = (8,4), \alpha=1$

Emphasizing a Specific Pass Band

$(\sigma_1,\sigma_0) = (16,8),\ \alpha=1$

Emphasizing a Specific Pass Band

$(\sigma_1, \sigma_0) = (32, 16),\ \alpha = 1$

Emphasizing a Specific Pass Band

$(\sigma_1,\sigma_0) = (64,32)$, $\alpha=1$

Emphasizing a Specific Pass Band

Noise Enhancement: the Problem with Sharpening

Effects of Noise on Enhancement of HF

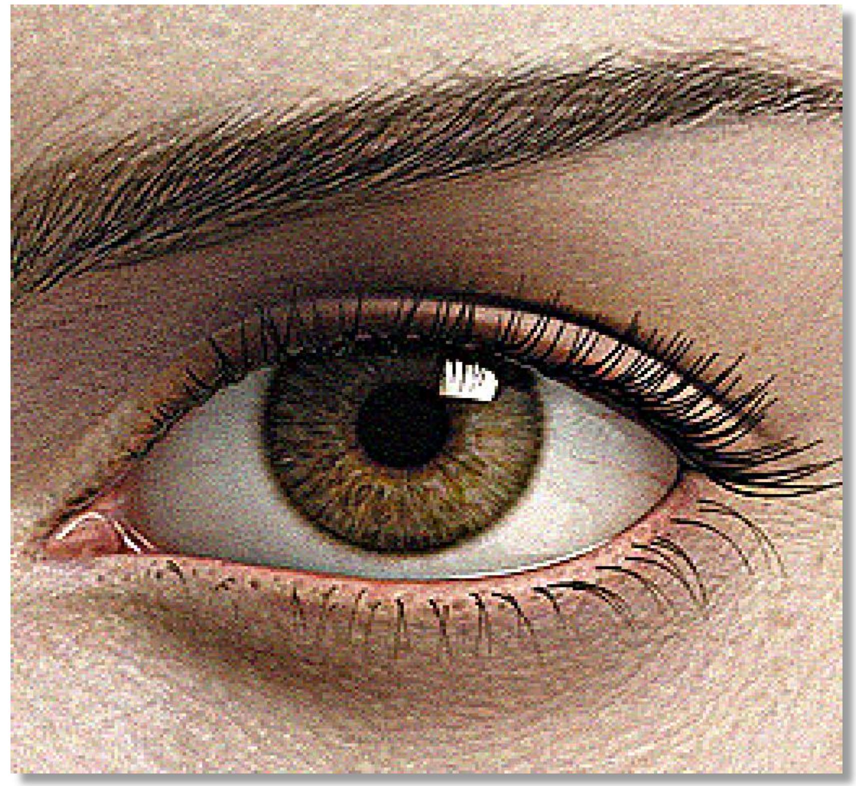
original image

HF enhanced original

Noise Enhancement: the Problem with Sharpening

Effects of Noise on Enhancement of HF

noisy image



HF enhanced noisy image