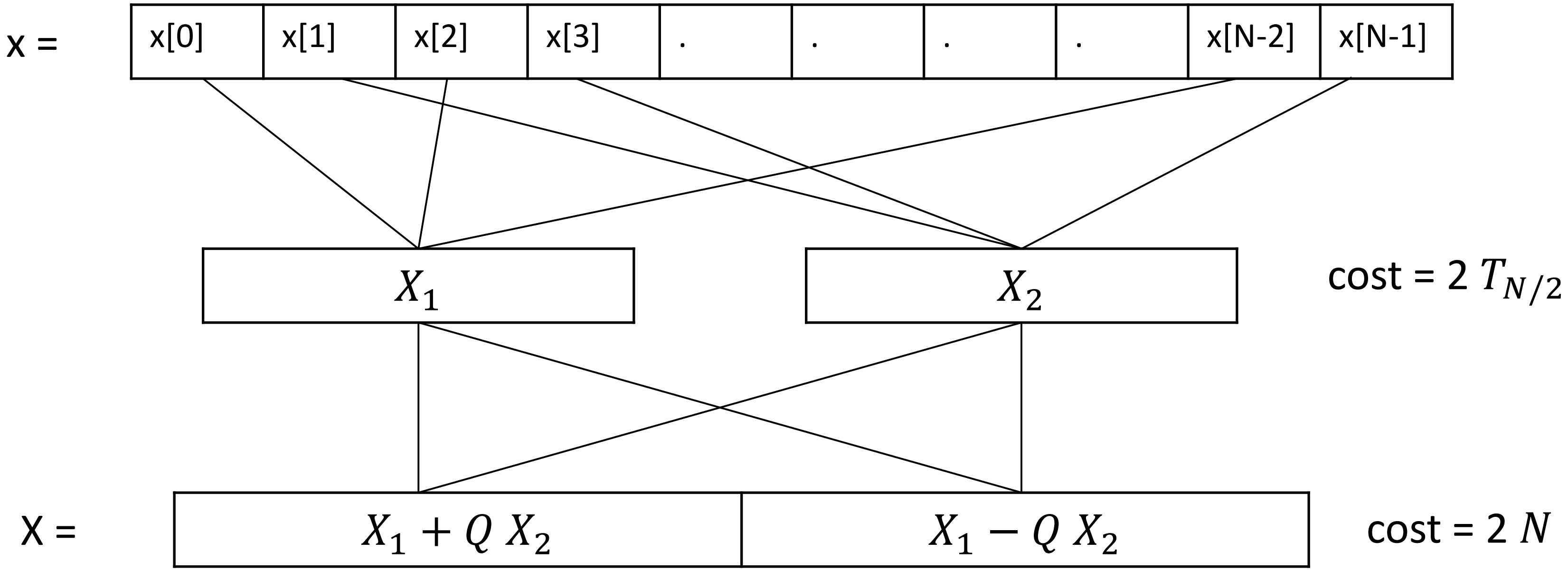
Digital Image Processing (CSE/ECE 478)
Lecture9: Fast Fourier Transforms and Filtering in Fourier Domain

Vineet Gandhi

Center for Visual Information Technology (CVIT), IIIT Hyderabad

Fast Fourier Transform



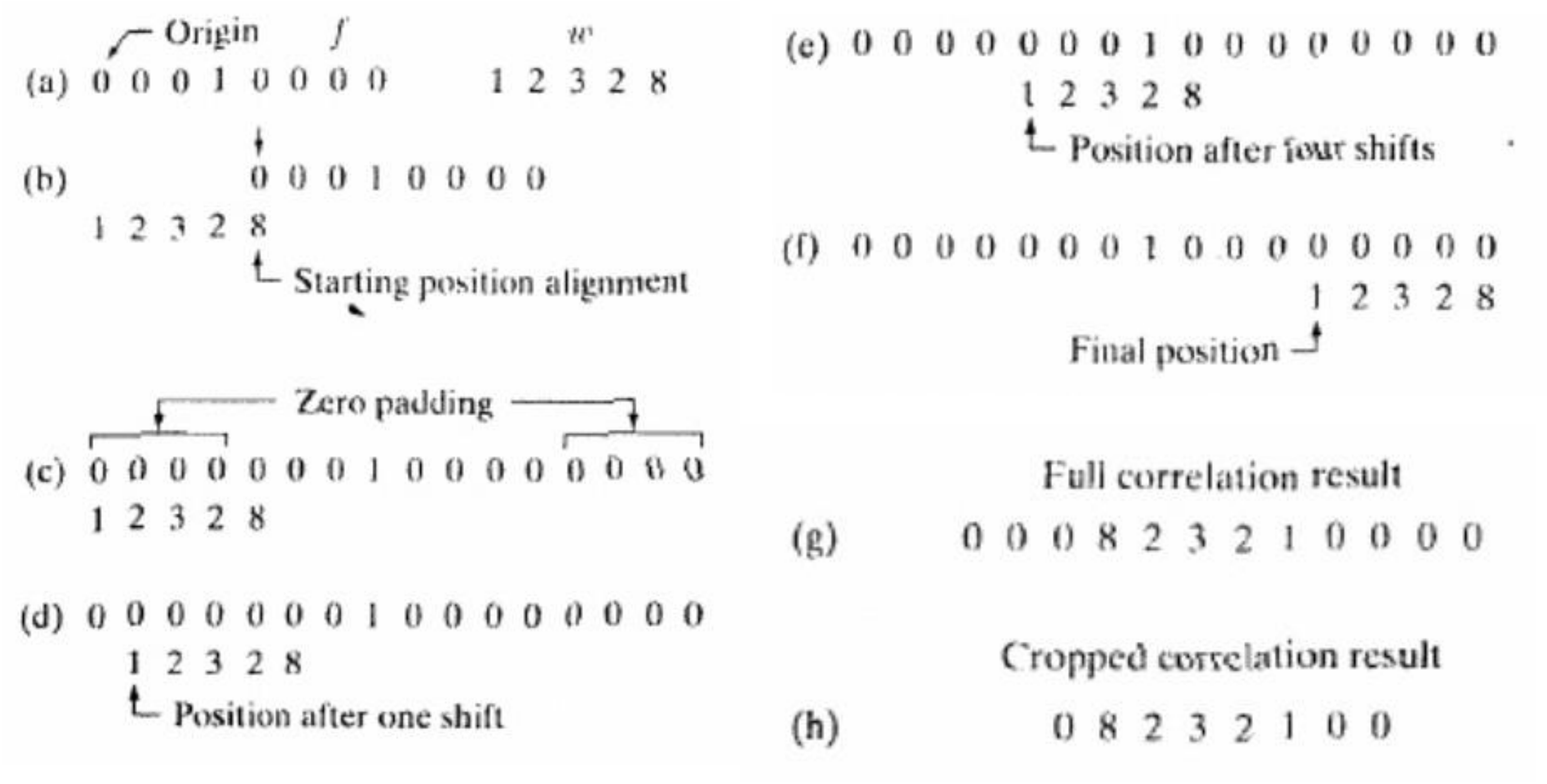
DFT vs FFT computation times

n	$N = 2^n$	N^2	$N \log N$
10	1 024	1 048 576	10 240
12	4 096	16 777 216	49 152
14	16 384	268 435 456	229 376
16	65 536	4 294 967 296	1 048 576

Today's class

- Convolution Theorem
- Frequency domain filtering
  - Low pass
  - High Pass
  - Laplacian

Correlation



Convolution

Origin f w rotated 180°

0 0 0 1 0 0 0 0   8 2 3 2 1 (i)

0 0 0 1 0 0 0 0 (j)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (k)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (l)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (m)
8 2 3 2 1

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 (n)
8 2 3 2 1

Full convolution result

0 0 0 1 2 3 2 8 0 0 0 0 (o)

Cropped convolution result

0 1 2 3 2 8 0 0 (p)

Convolution vs Correlation (2D)

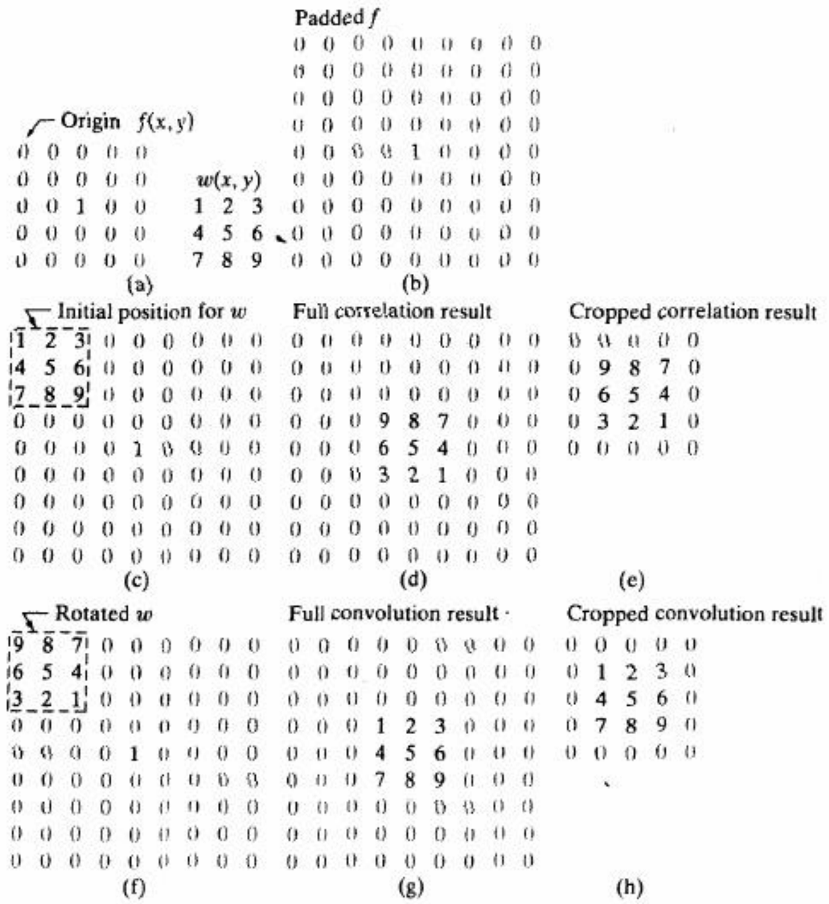


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Convolution (2D)

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

- Evaluated for all values of displacement variables x and y
- Filter size m × n (notational convenience → m, n are assumed odd)
- a = (m-1)/2 and b = (n-1)/2

Convolution Theorem

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

In other words:

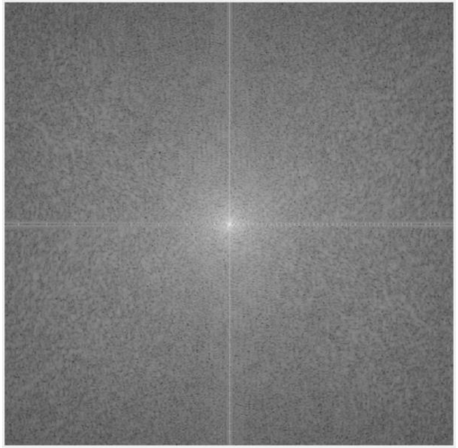
$$\Im(f(x,y) \bigstar h(x,y)) = F(u,v)H(u,v)$$

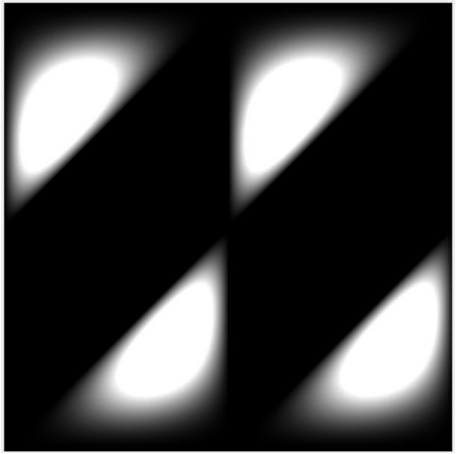
$$f(x,y) \bigstar h(x,y) = \Im^{-1}(F(u,v)H(u,v))$$

Correspondence to spatial filtering

-1	0	1
-2	0	2
-1	0	1

Correspondence to spatial filtering





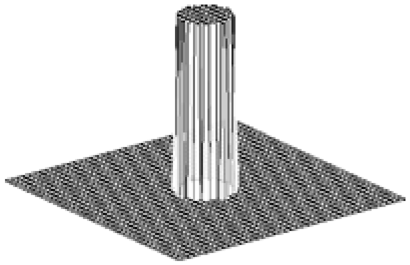
-1	0	1
-2	0	2
-1	0	1

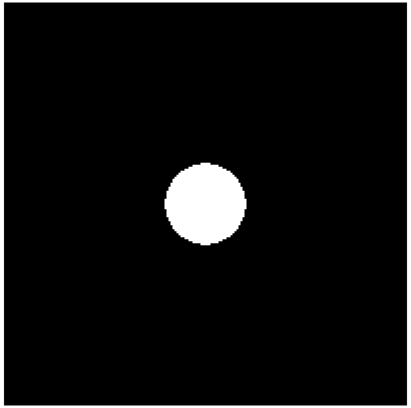
Correspondence to spatial filtering

```
%Sobel filter in frequency domain
f = rgb2gray(imread('boy.jpg'));
h = [-1 0 1; -2 0 2; -1 0 1];
F = fft2(double(f), 402, 402);
H = fft2(double(h), 402, 402);
F_fH = fftshift(H).*fftshift(F);
ffi = ifft2(ifftshift(F_fH));
```

Ideal Low Pass Filters

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





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

D_0 → cut off frequency

Image courtesy: Gonzalez and Woods

Ideal Low Pass Filters

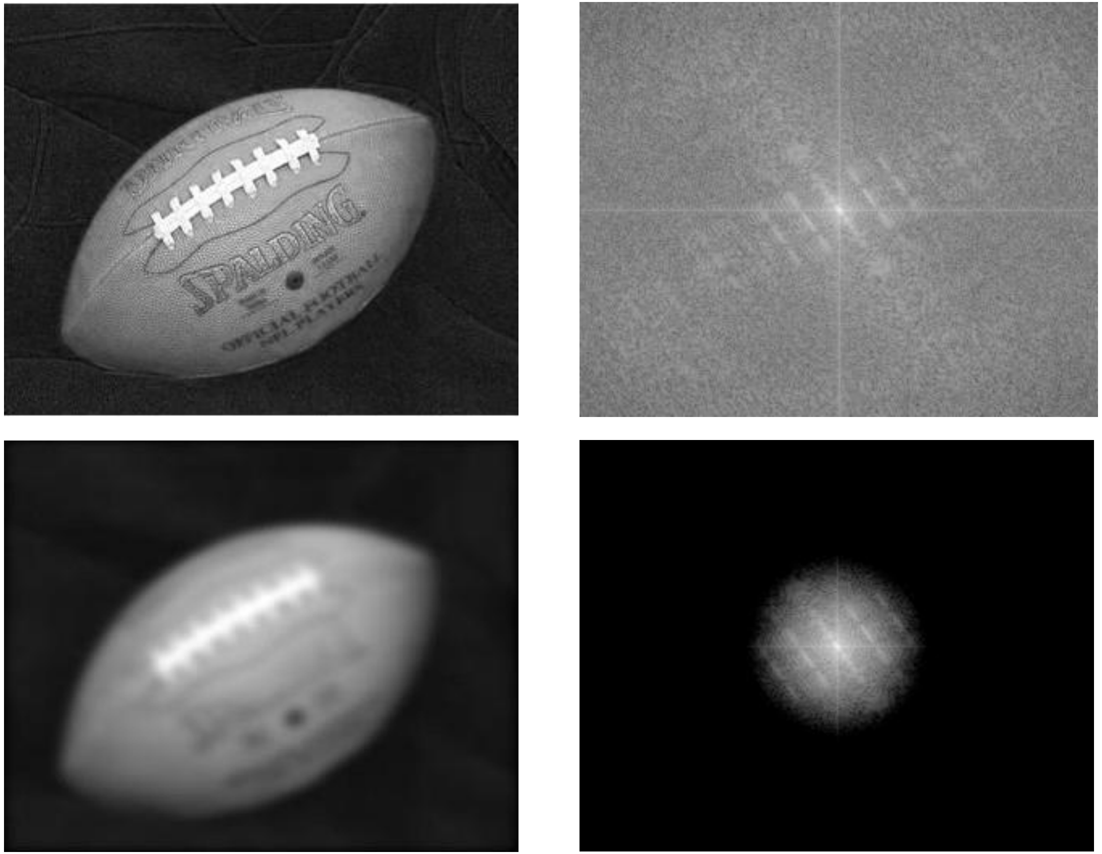
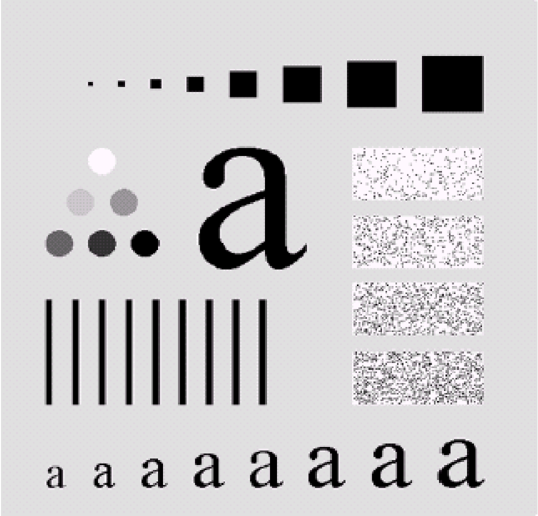
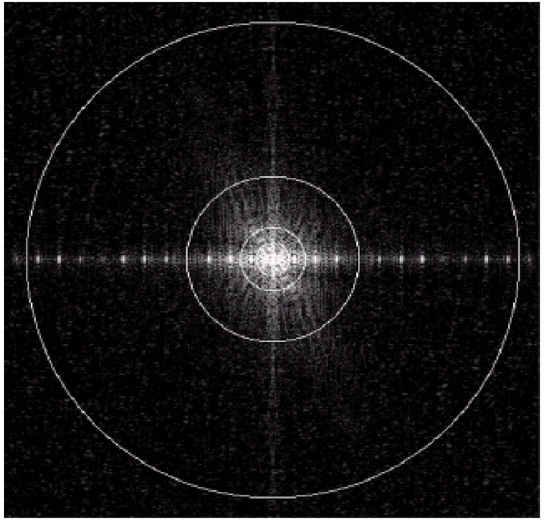


Image courtesy: cs.uregina.ca

Ideal Low Pass Filters

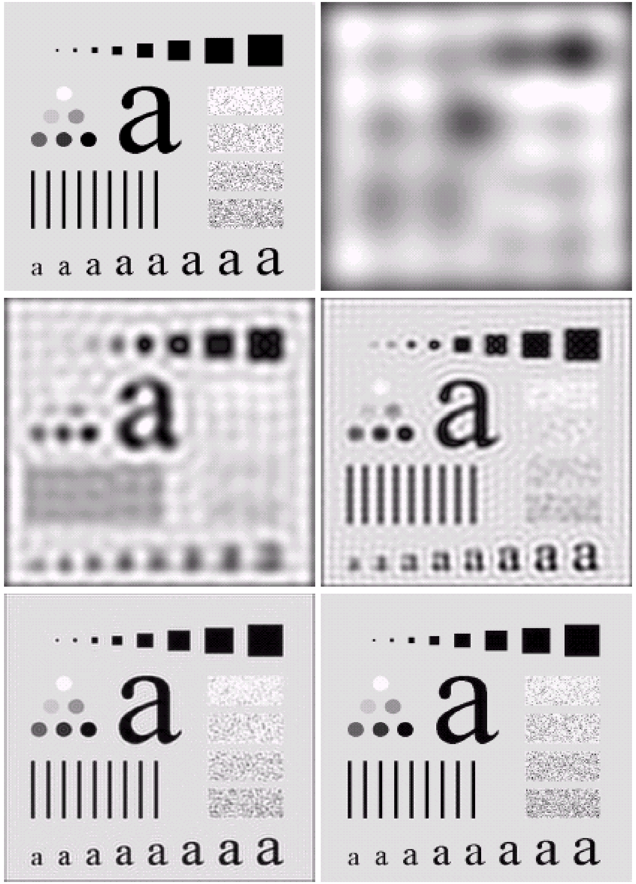




Radii 10,30,60,160 and 460 → power 87, 93.1, 95.7, 97.8 and 99..2

Image courtesy: Gonzalez and Woods

Ideal Low Pass Filters



ILPF radius 10

ILPF radius 30

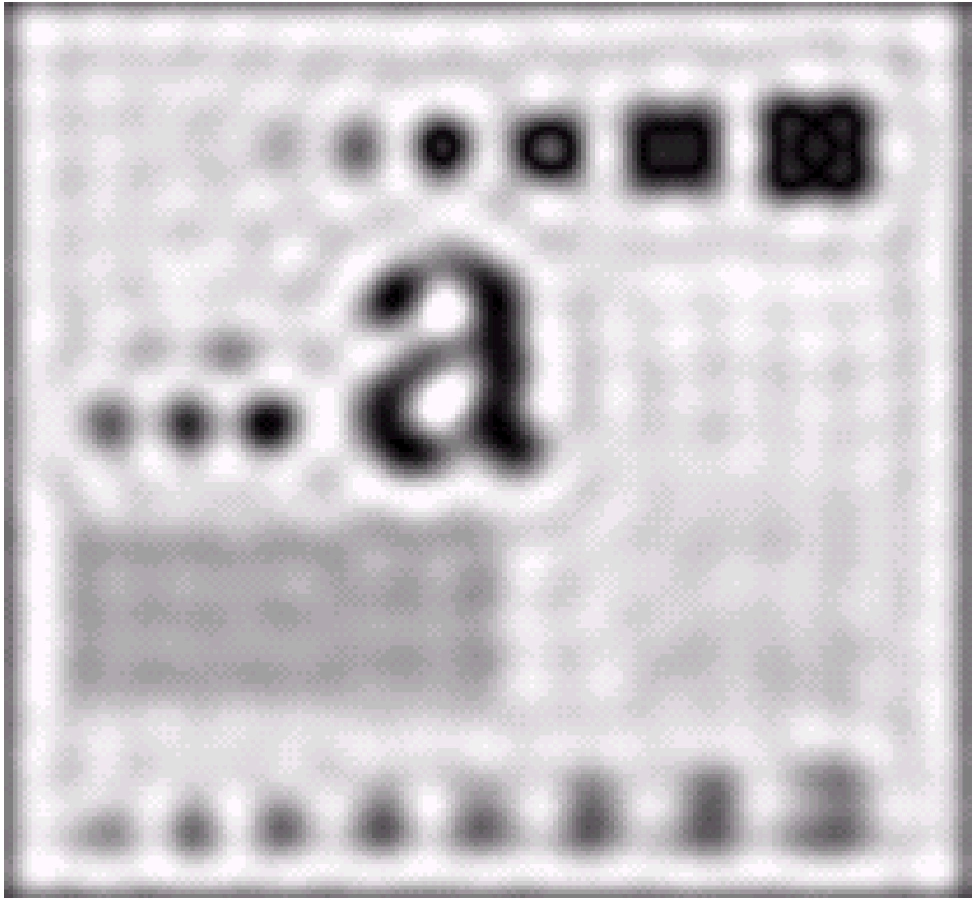
ILPF radius 60

ILPF radius 160

ILPF radius 460

Image courtesy: Gonzalez and Woods

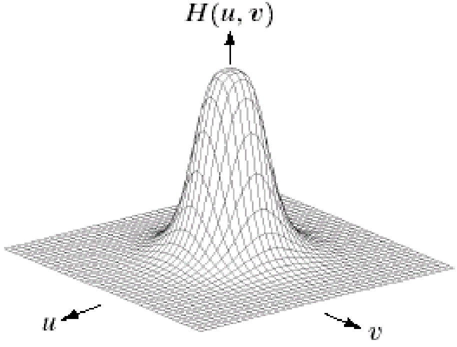
Ideal Low Pass Filters

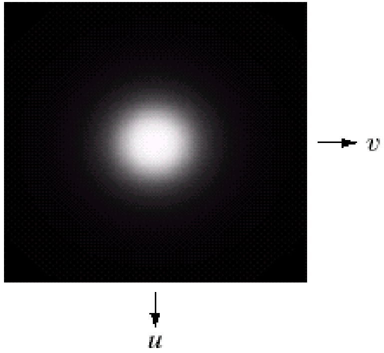


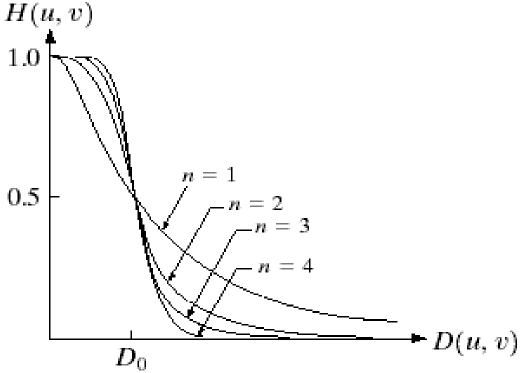
ILPF radius 30

Image courtesy: Gonzalez and Woods

Butterworth Low Pass Filters





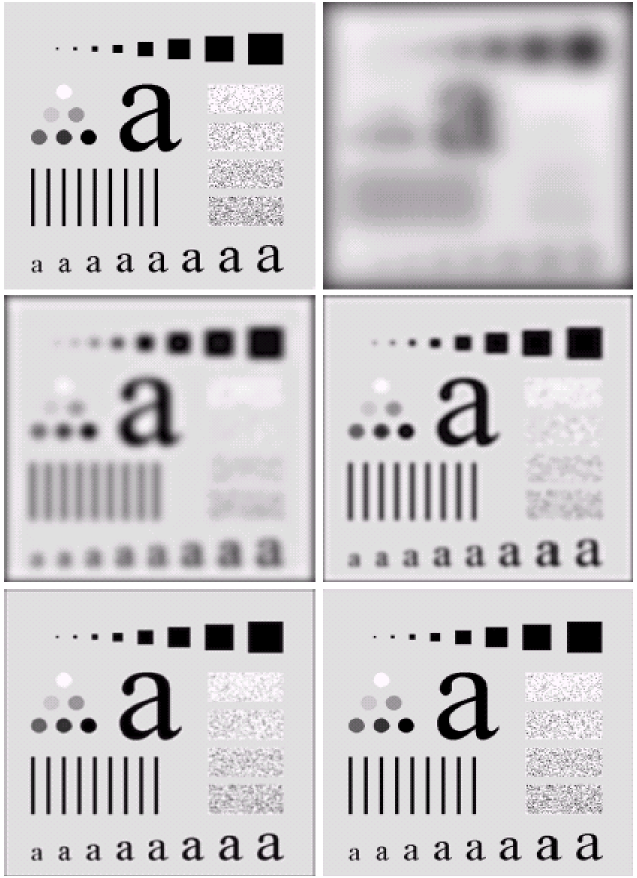


$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$ where $$D(u,v) = [(u-M/2)^2 + (v-N/2)^2]^{1/2}$$

Image courtesy: Gonzalez and Woods

Butterworth Low Pass Filters (BLPF)

Order two, i.e.
n=2



BLPF cut off frequency 10

BLPF cut off frequency 30

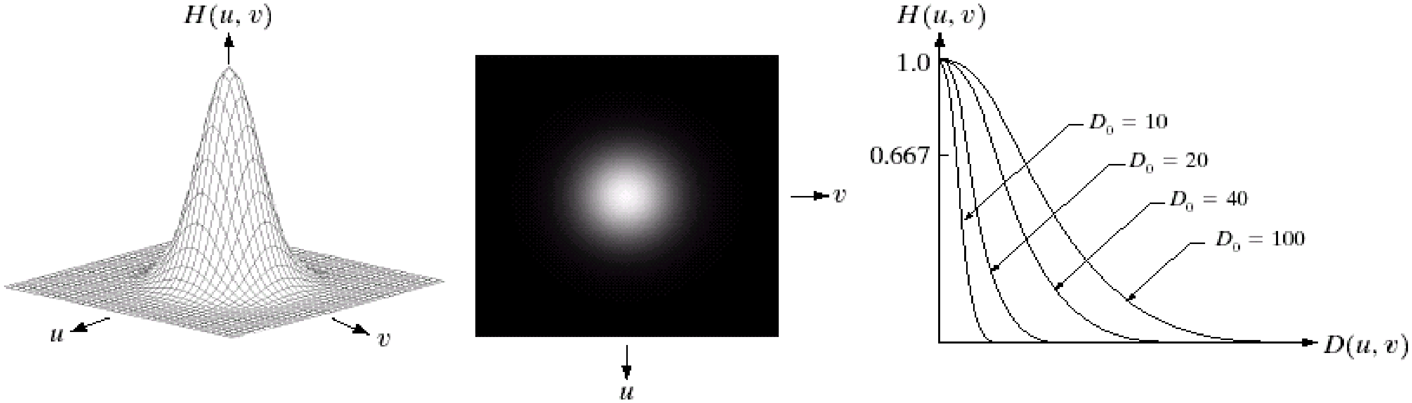
BLPF cut off frequency 60

BLPF cut off frequency 160

BLPF cut off frequency 460

Image courtesy: Gonzalez and Woods

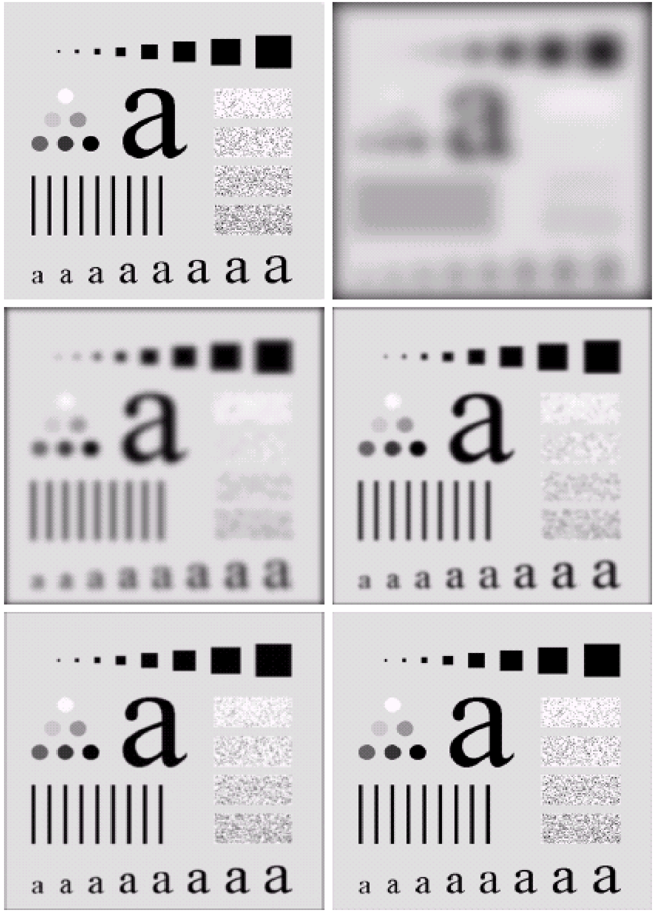
Gaussian Low Pass Filters



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

Image courtesy: Gonzalez and Woods

Gaussian Low Pass Filters (GLPF)



GLPF cut off frequency 10

GLPF cut off frequency 30

GLPF cut off frequency 60

GLPF cut off frequency 160

GLPF cut off frequency 460

Image courtesy: Gonzalez and Woods

Comparison (ILPF, BLPF, GLPF)

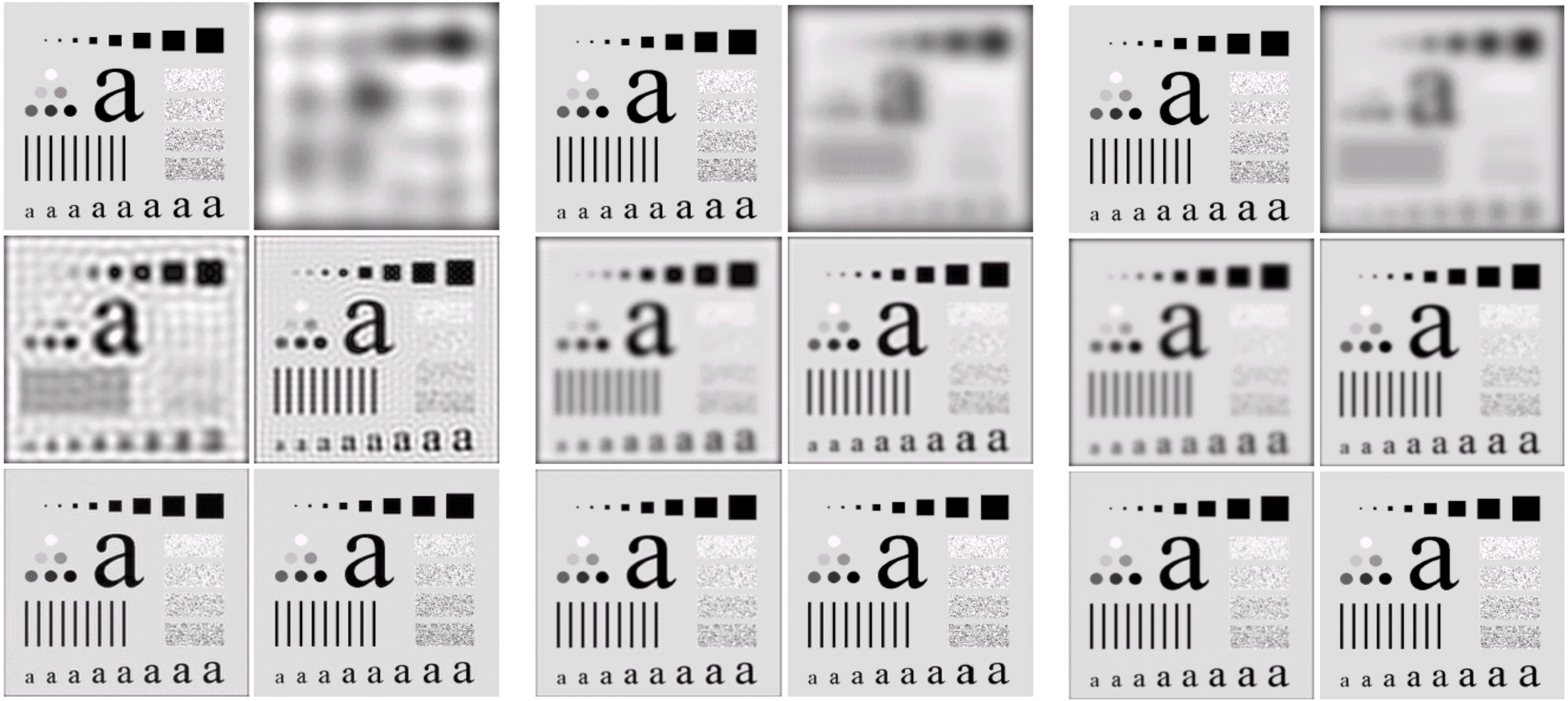
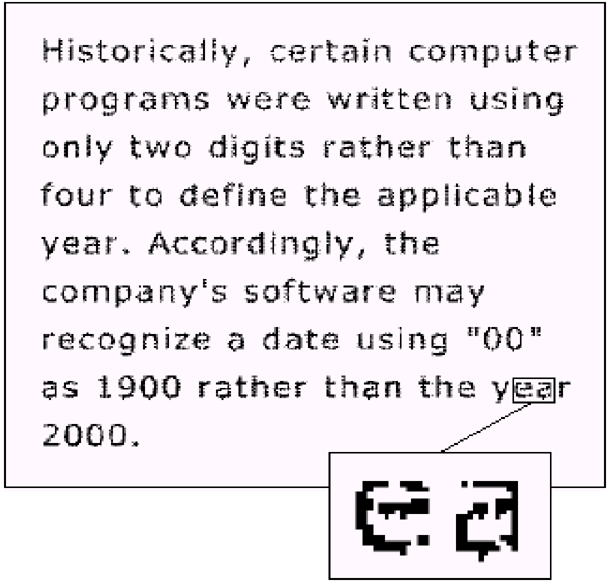


Image courtesy: Gonzalez and Woods

Low pass filtering application



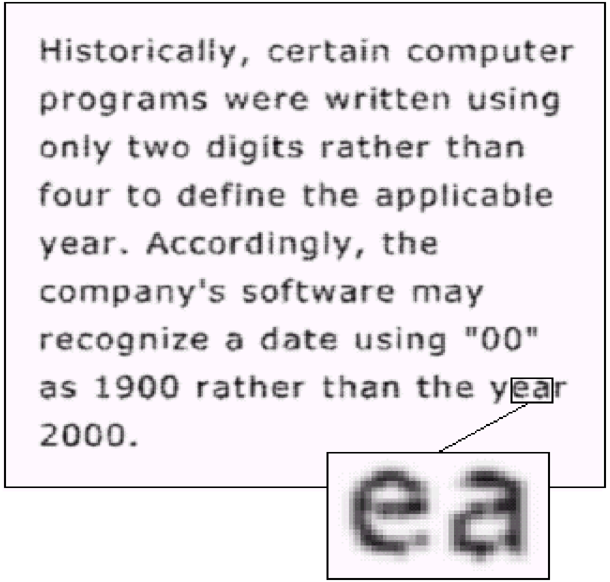


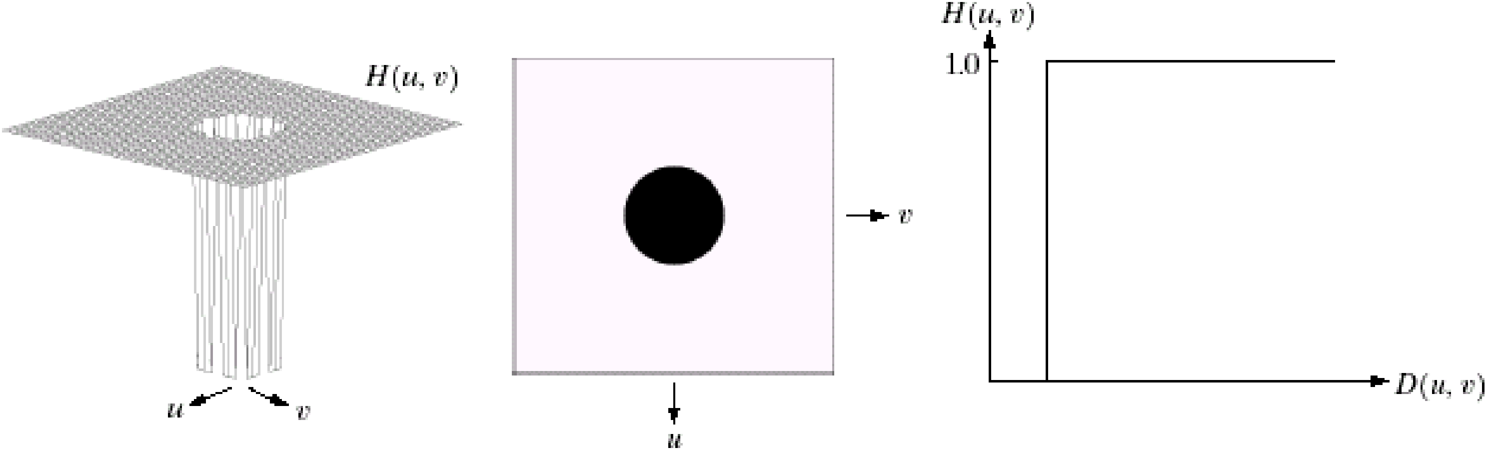
Image courtesy: Gonzalez and Woods

Image Sharpening in Frequency Domain

High Pass filter can be obtained from a given low pass filter:

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

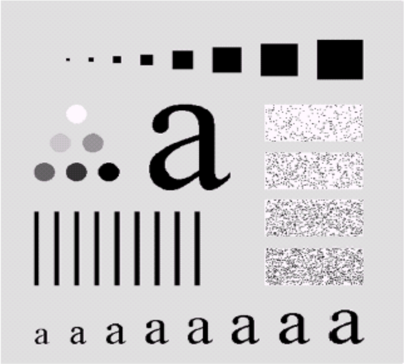
Ideal High Pass Filters

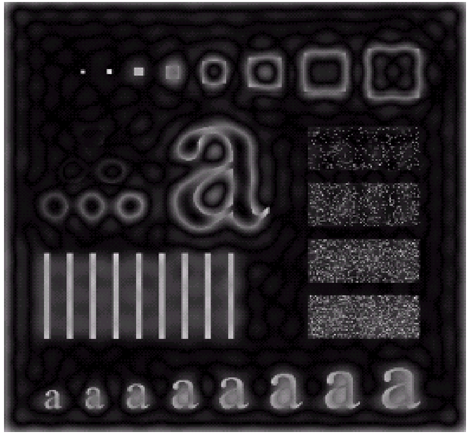


$$H(u,v)=\begin{cases}0 \text{ if } D(u,v)\le D_0\\1 \text{ if } D(u,v)>D_0\end{cases}$$

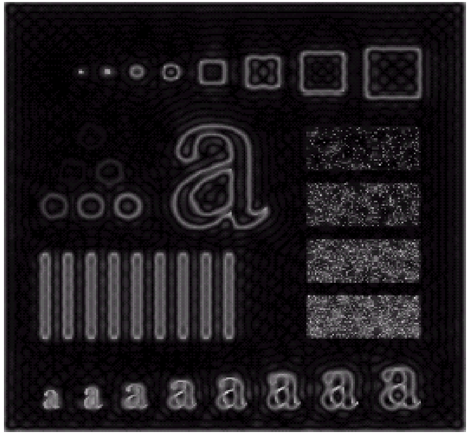
Image courtesy: Gonzalez and Woods

Ideal High Pass Filters

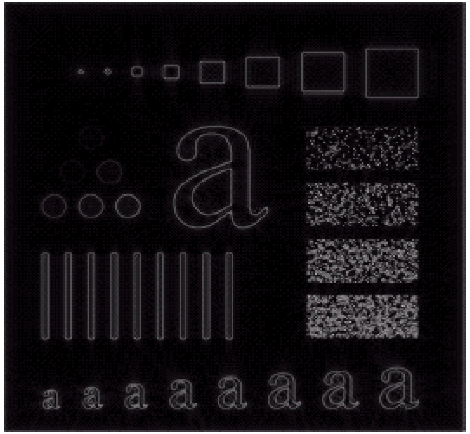




IHPL with $D_0 = 30$



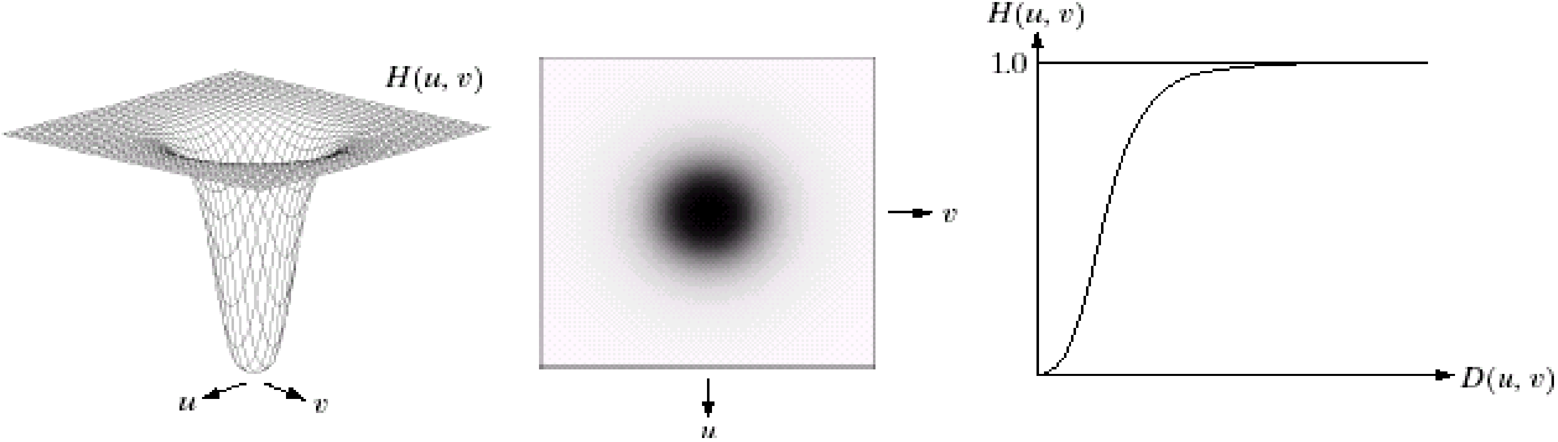
IHPF with $D_0 = 60$



IHPF with $D_0 = 160$

Image courtesy: Gonzalez and Woods

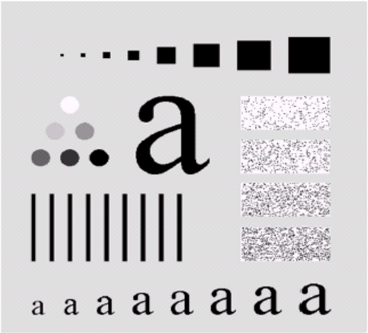
Butterworth High Pass Filters

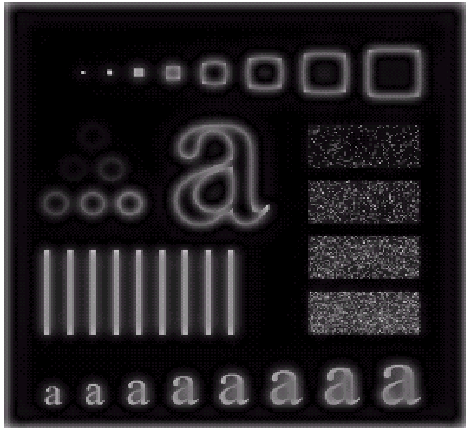


$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

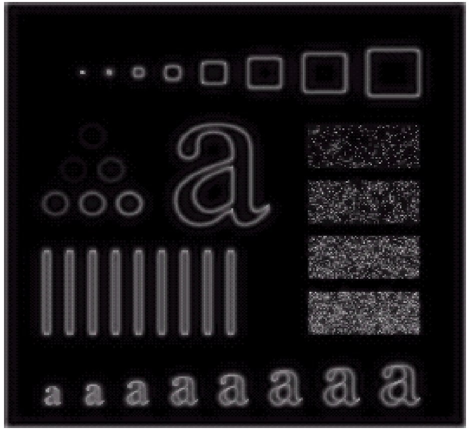
Image courtesy: Gonzalez and Woods

Butterworth High Pass Filters

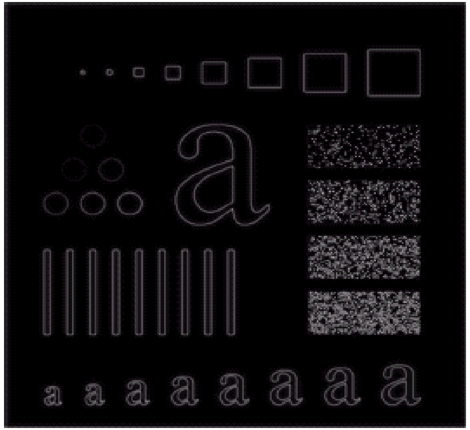




BHPL with D_0 = 30



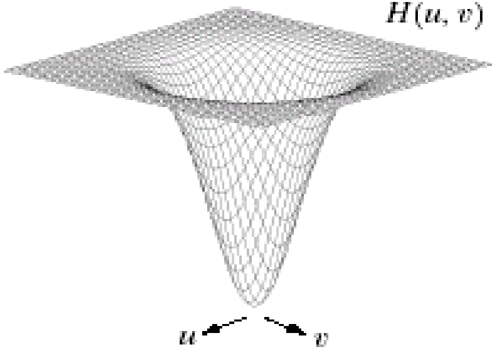
BHPF with D_0 = 60

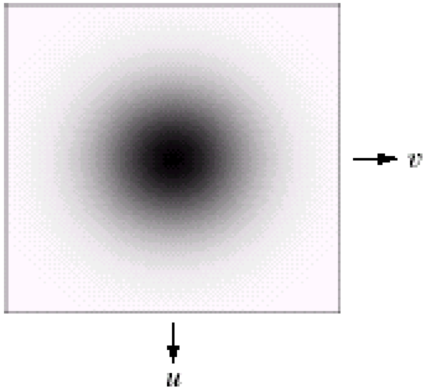


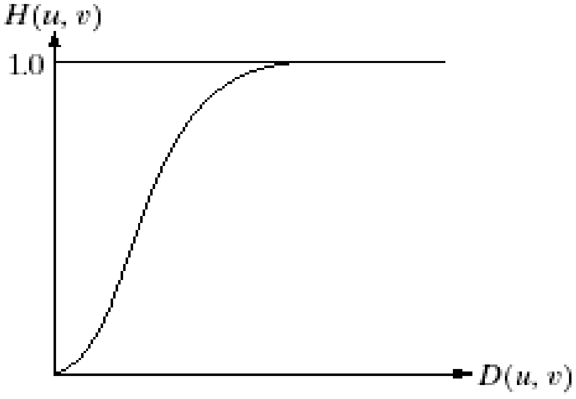
BHPF with D_0 = 160

Image courtesy: Gonzalez and Woods

Gaussian High Pass Filters



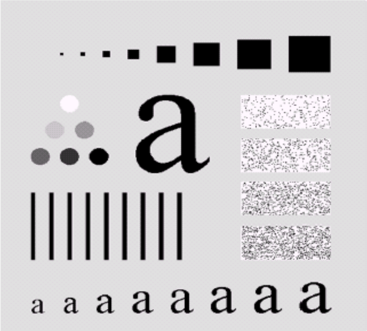


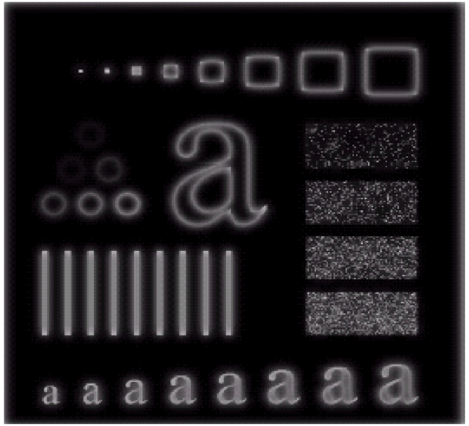


$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

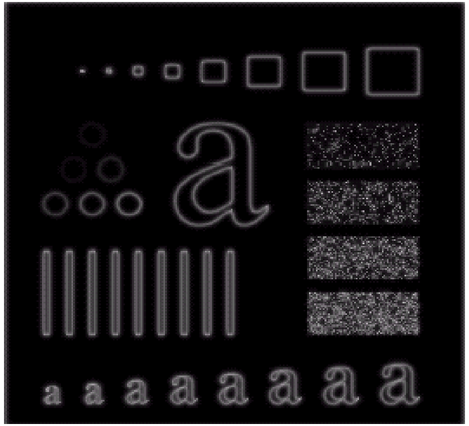
Image courtesy: Gonzalez and Woods

Gaussian High Pass Filters

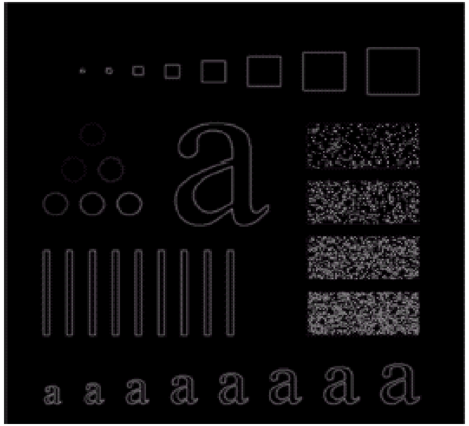




GHPL with D_0 = 30



GHPF with D_0 = 60



GHPF with D_0 = 160

Image courtesy: Gonzalez and Woods

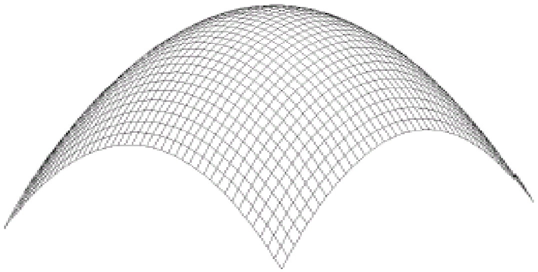
Laplacian in frequency domain

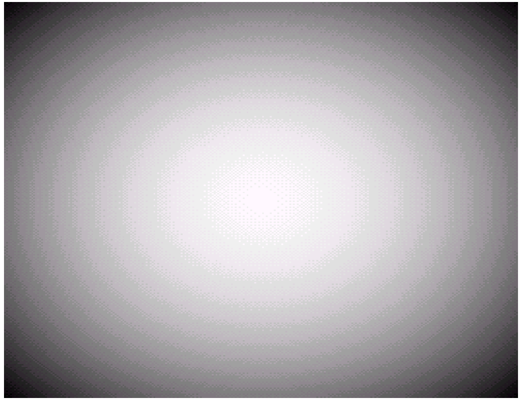
$$\Im\left[\frac{d^n f(x)}{dx^n}\right] = (ju)^n F(u)$$

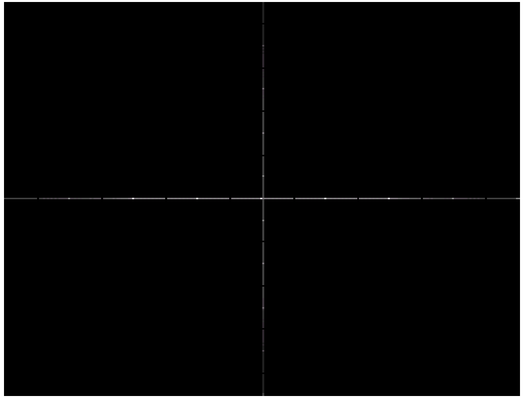
$$\Im\left[\frac{\partial^2(f(x,y))}{\partial x^2} + \frac{\partial^2(f(x,y))}{\partial y^2}\right] = (ju)^2 F(u,v) + (jv)^2 F(u,v)$$

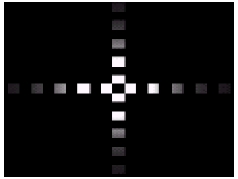
$$= -(u^2 + v^2) F(u,v)$$

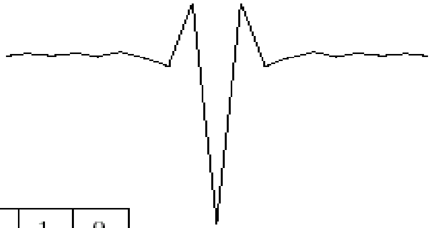
Laplacian in frequency domain











0	1	0
1	−4	1
0	1	0

Image courtesy: Gonzalez and Woods

Notch Reject filter (Notch pass filter)

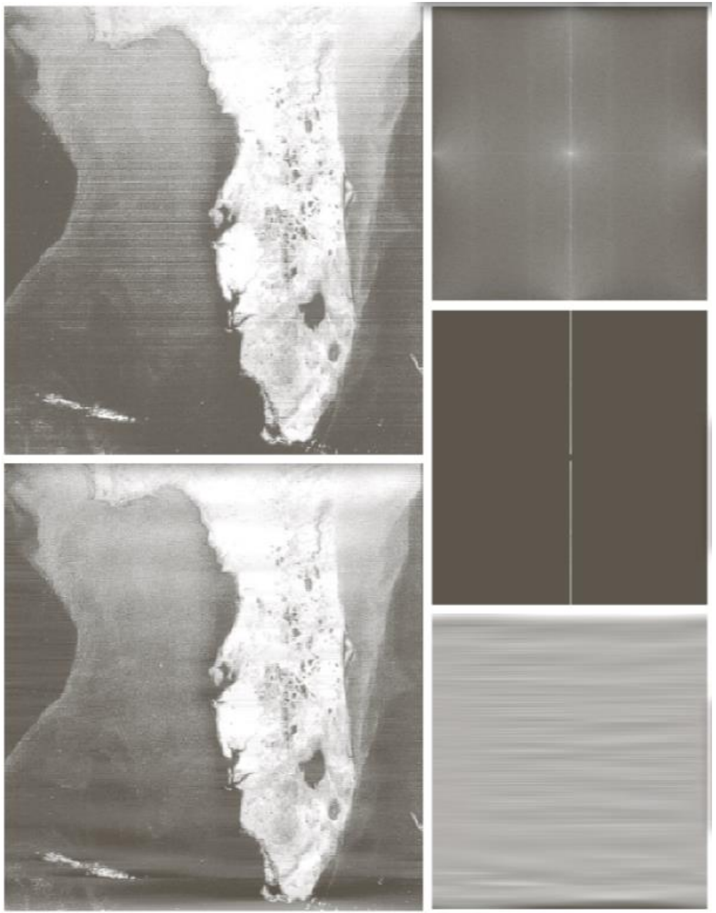


Image courtesy: Gonzalez and Woods

Filtering in frequency domain

- Band reject (Band pass filters)
- Unsharp Masking and High boost filtering
- Homomorphic filtering