Digital Image Processing

Dr. Mubashir Ahmad (Ph.D.)

Low Pass Filters in Frequency Domain

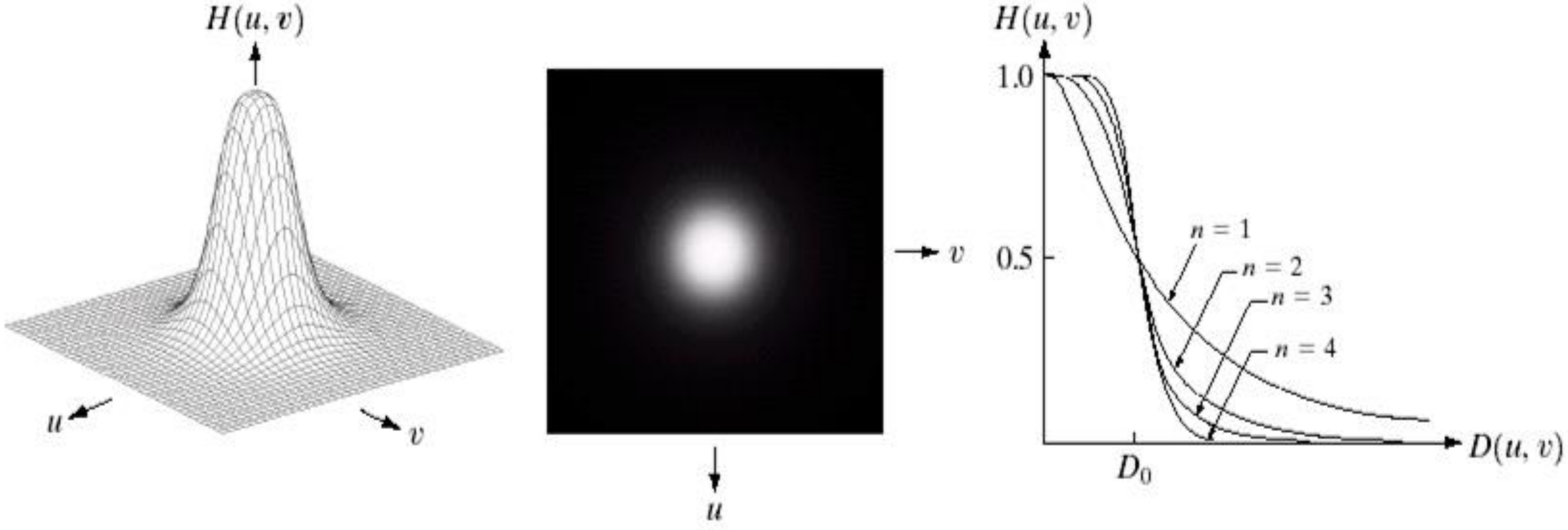
- Ideal low pass filters
- Butterworth low pass filters
- Gaussian low pass filters

Butterworth Lowpass Filters

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

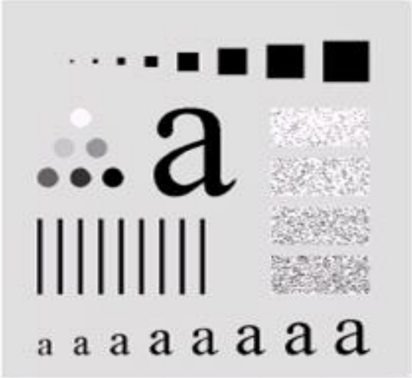
The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

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



Butterworth Lowpass Filter (cont…)

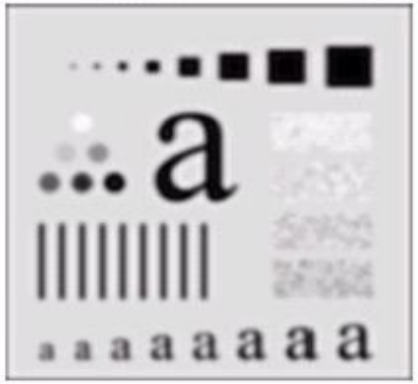
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



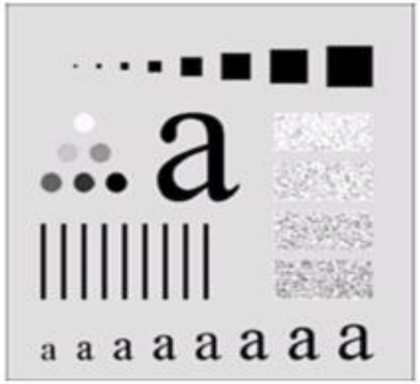
Original image

Result of filtering with Butterworth filter of order 2 and cutoff radius 5

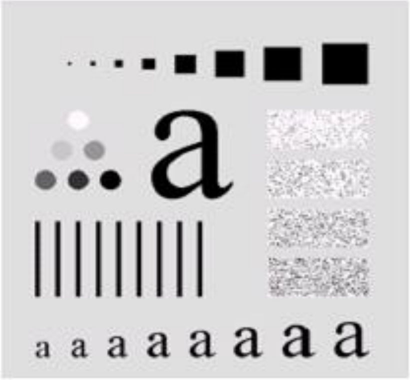
Result of filtering with Butterworth filter of order 2 and cutoff radius 15



Result of filtering with Butterworth filter of order 2 and cutoff radius 30



Result of filtering with Butterworth filter of order 2 and cutoff radius 80



Result of filtering with Butterworth filter of order 2 and cutoff radius 230

Butterworth Lowpass Filter (cont…)

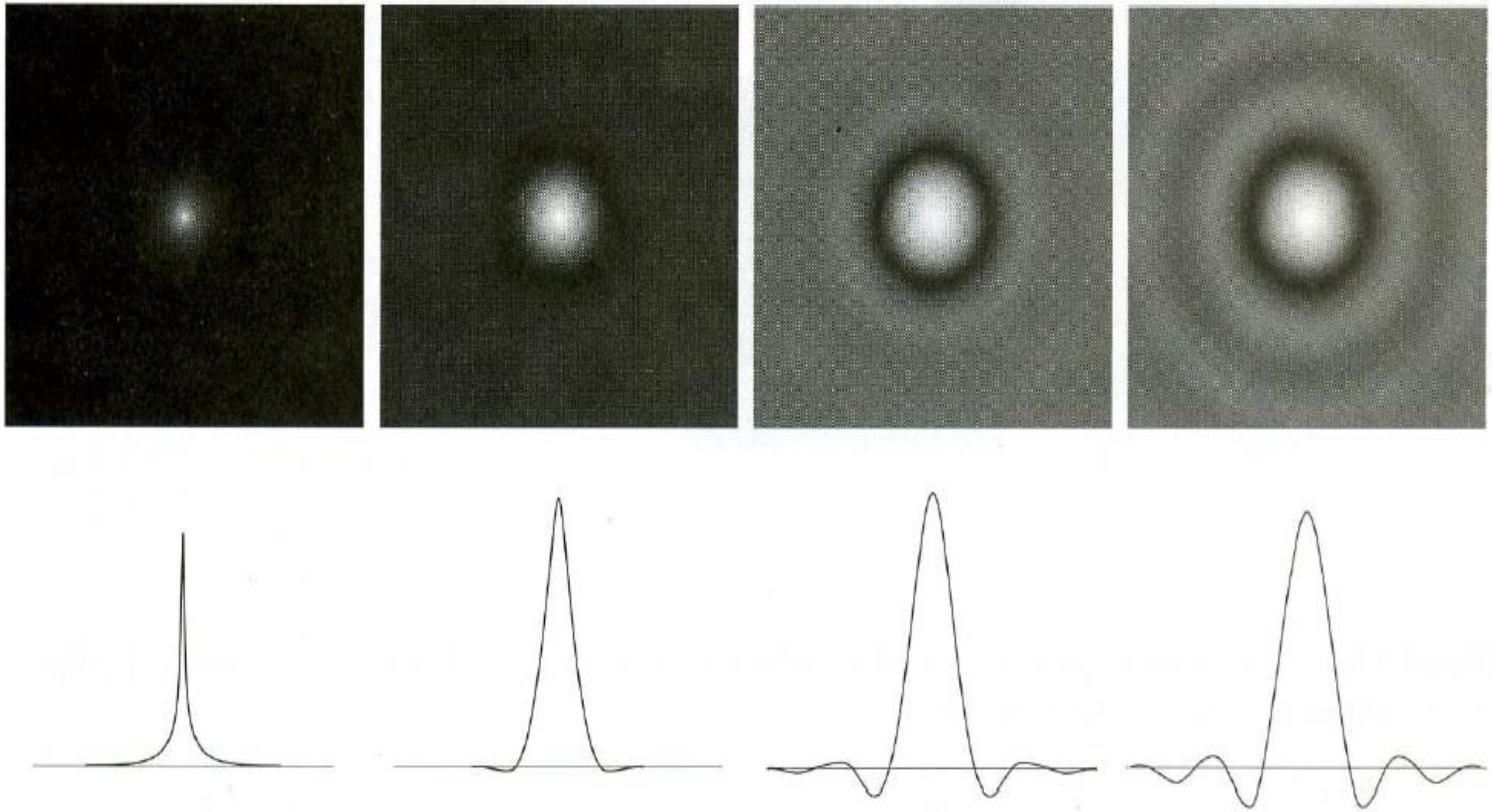
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Result of filtering with Butterworth filter of order 2 and cutoff radius 5

Butterworth Lowpass Filter (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Result of filtering with Butterworth filter of order 2 and cutoff radius 15



a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Calculate the spatial domain to the frequency domain to perform Butterworth Low pass filter, Where cutoff frequency = 20 and n = 2

$$F(x, y) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

- Input image f(x, y) in a spatial domain.
- Multiply the input image with (-1)^x+y to move the transform in the center.

$$(-1)^{x+y} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Butterworth Low pass filter

- After the transform center arrangement we got the results as follows.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix}$$

Butterworth Low pass filter: Compute the DFT of the image

- Compute the DFT of the image.
- F(u, v) = kernel * f(x, y) * kernel ^Transpose

$$f(u,v) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}$$

Butterworth Low pass filter: Compute the DFT of the image

$$\Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Distance calculation

- Distance formula = (x^2 + y^2)^1/2

$$V = \begin{bmatrix} -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$D(U, V) = \begin{bmatrix} (-2, -2) & (-1, -2) & (0, -2) & (1, -2) \\ (-2, -1) & (-1, -1) & (0, -1) & (1, -1) \\ (-2, 0) & (-1, 0) & (0, 0) & (1, 0) \\ (-2, 1) & (-1, 1) & (0, 1) & (1, 1) \end{bmatrix}$$

Distance calculation

- D(-2, -2) = 2.82
- D(-1, -2) = 2.23
- D(0, -2) = 2
- D(1, -1) = 2.23
- D(-2, -1) = 2.23
- D(-1, -1) = 1.41
- D(0, -1) = 1
- D(1, -1) = 1.41
- D(-2, 0) = 2
- D(-1, 0) = 1.41
- D(0, 0) = 0
- D(1, 0) = 1
- D(-2, 1) = 2.23
- D(-1, 1) = 1.41
- D(0, 1) = 1
- D(1, 1) = 1.41

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

$$D(U, V) = \begin{bmatrix} 2.8284 & 2.2361 & 2.0000 & 2.2361 \\ 2.2361 & 1.4142 & 1.0000 & 1.4142 \\ 2.0000 & 1.0000 & 0 & 1.0000 \\ 2.2361 & 1.4142 & 1.0000 & 1.4142 \end{bmatrix}$$

Apply Butterworth formula

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

$$\text{Butterworth} = \begin{pmatrix} 0.99 & 0.99 & 0.99 & 0.99 \\ 0.99 & 1 & 1 & 1 \\ 0.99 & 1 & 1 & 1 \\ 0.99 & 1 & 1 & 1 \end{pmatrix}$$

Applying the Butterworth filter to the image in Frequency domain

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{pmatrix} 0.99 & 0.99 & 0.99 & 0.99 \\ 0.99 & 1 & 1 & 1 \\ 0.99 & 1 & 1 & 1 \\ 0.99 & 1 & 1 & 1 \end{pmatrix}$$

Inverse Fourier transform

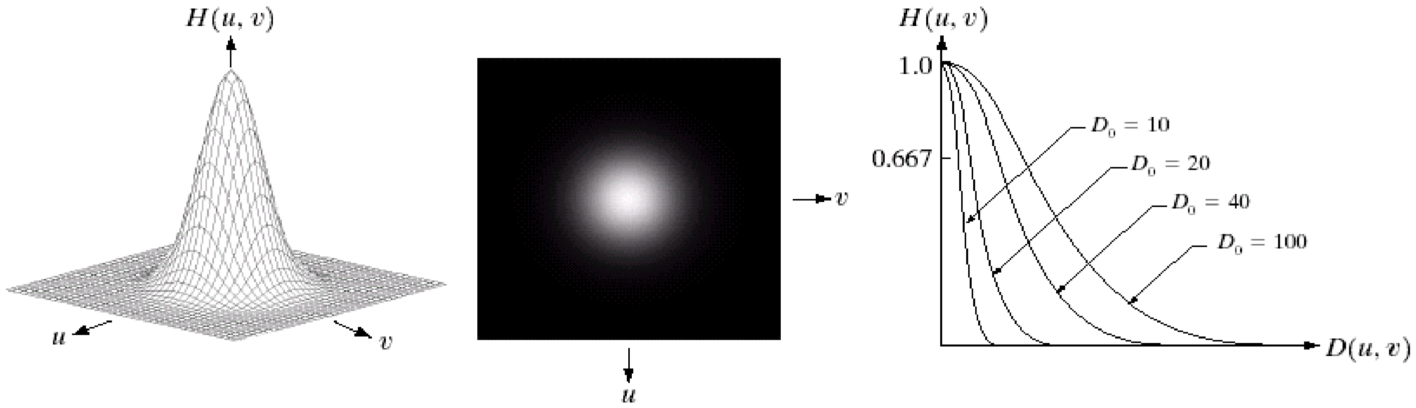
$$f(x,y) = \frac{1}{16}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 7.9101 & 0 & 8.0000 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}$$

$$f(x,y) = \begin{bmatrix} 0.9944 & -0.0056 & 0.9944 & -0.0056 \\ -0.9944 & 0.0056 & -0.9944 & 0.0056 \\ 0.9944 & -0.0056 & 0.9944 & -0.0056 \\ -0.9944 & 0.0056 & -0.9944 & 0.0056 \end{bmatrix}$$

Gaussian Lowpass Filters

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

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



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Calculate the spatial domain to the frequency domain to perform the Gaussian Low pass filter, Where cutoff frequency = 1

$$F(x, y) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

- Input image f(x, y) in a spatial domain.
- Multiply the input image with (-1)^x+y to move the transform in the center.

$$(-1)^{x+y} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Gaussian Low pass filter

- After the transform center arrangement we got the results as follows.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix}$$

Gaussian Low pass filter: Compute the DFT of the image

- Compute the DFT of the image.
- F(u, v) = kernel * f(x, y) * kernel ^ Transpose

$$f(u,v) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}$$

Gaussian Low pass filter: Compute the DFT of the image

$$\Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Gaussian Low pass filter formula

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

$$x = \begin{bmatrix} -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

```
distances = sqrt(x.^2 + y.^2);
```

$$\begin{bmatrix} 2.8284 & 2.2361 & 2.0000 & 2.2361 \\ 2.2361 & 1.4142 & 1.0000 & 1.4142 \\ 2.0000 & 1.0000 & 0 & 1.0000 \\ 2.2361 & 1.4142 & 1.0000 & 1.4142 \end{bmatrix}$$

```
gaussian_filter = exp(-(distances)^2 / (2*sigma^2));
```

gaussian_filter =

$$
\begin{bmatrix}
0.0183 & 0.0821 & 0.1353 & 0.0821 \\
0.0821 & 0.3679 & 0.6065 & 0.3679 \\
0.1353 & 0.6065 & 1.0000 & 0.6065 \\
0.0821 & 0.3679 & 0.6065 & 0.3679
\end{bmatrix}
$$

```
filtered_image_F = F (u v) .* gaussian_filter;
```

filtered_image_F =

$$
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
1.0827 & 0 & 8.0000 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
$$

Applying inverse Fourier transform

$$f(x,y) = \frac{1}{16}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.0827 & 0 & 8.0000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{pmatrix}$$

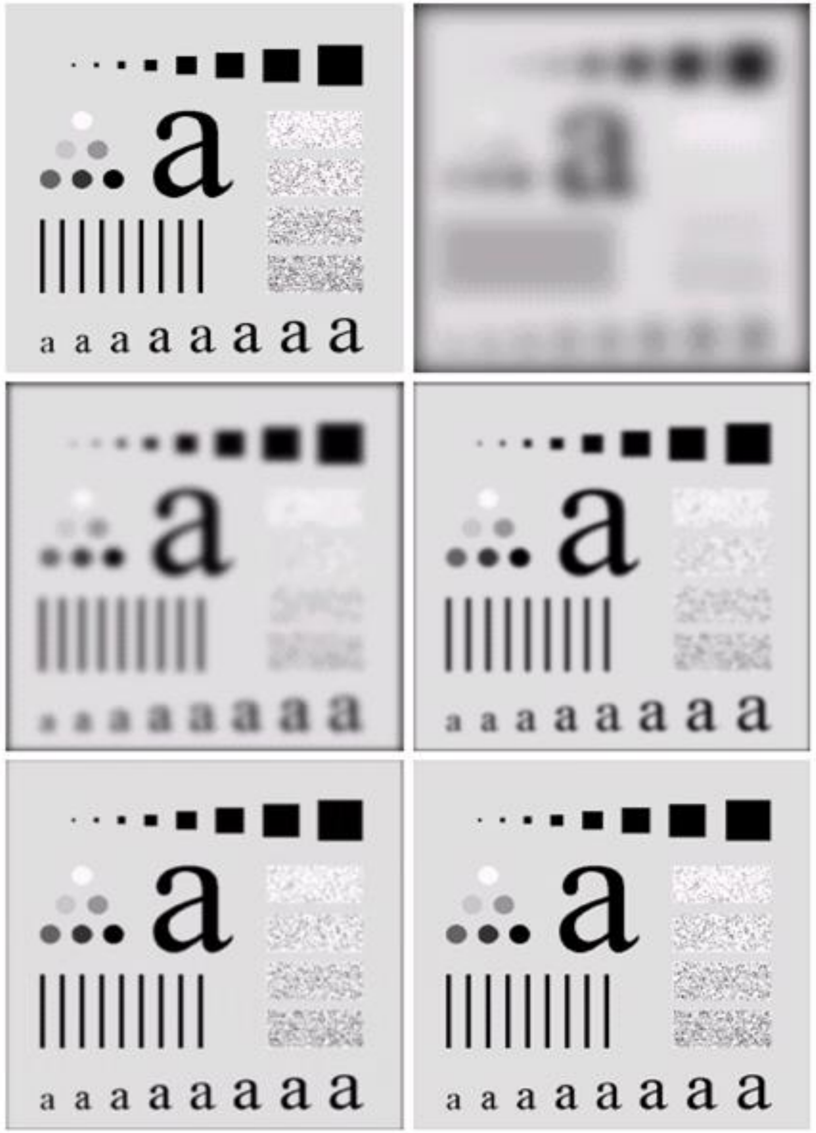
$$f(x,y) = \begin{bmatrix} 0.5677 & -0.4323 & 0.5677 & -0.4323 \\ -0.5677 & 0.4323 & -0.5677 & 0.4323 \\ 0.5677 & -0.4323 & 0.5677 & -0.4323 \\ -0.5677 & 0.4323 & -0.5677 & 0.4323 \end{bmatrix}$$

Gaussian Lowpass Filters (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

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

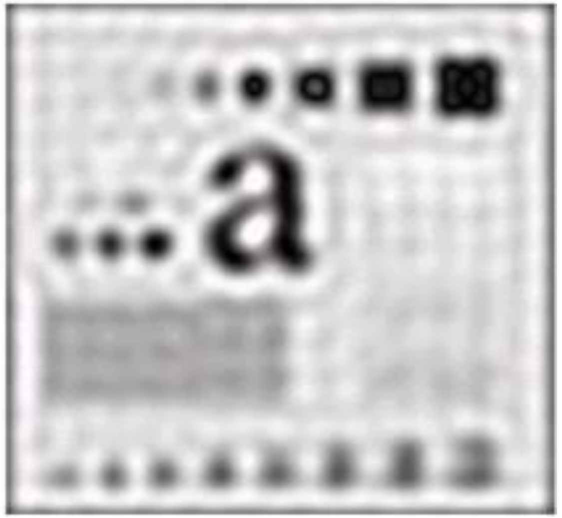
Result of filtering with Gaussian filter with cutoff radius 85

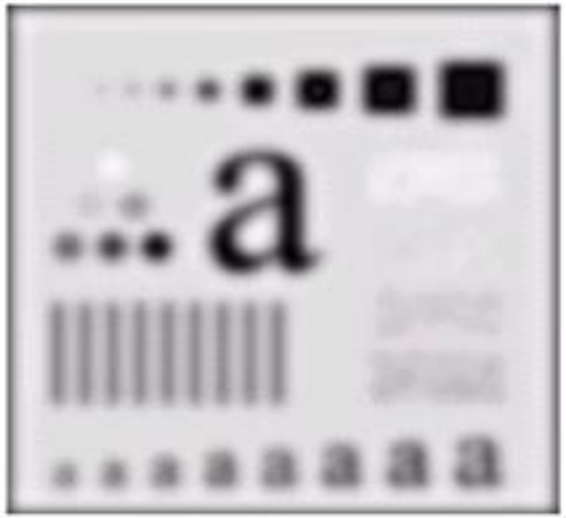
Result of filtering with Gaussian filter with cutoff radius 230

Lowpass Filters Compared

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

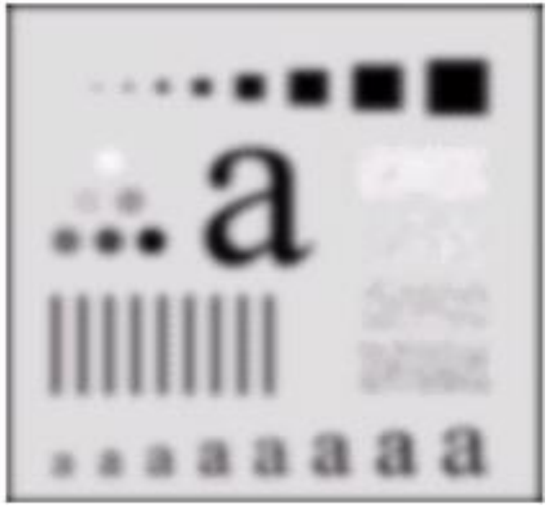
Result of filtering with ideal low pass filter of radius 15





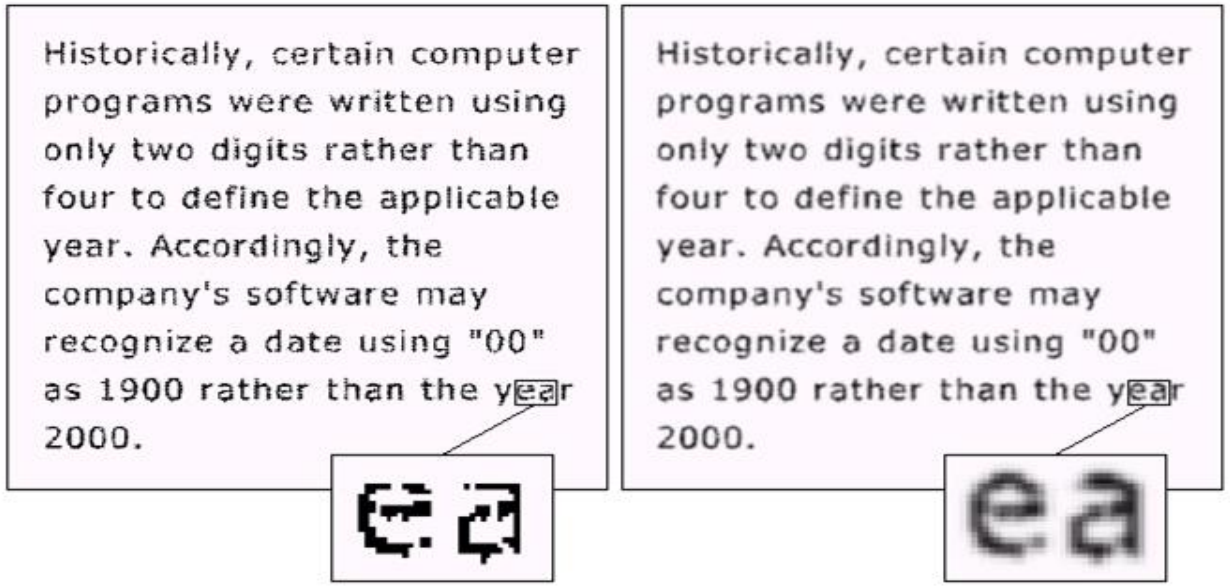
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Result of filtering with Gaussian filter with cutoff radius 15



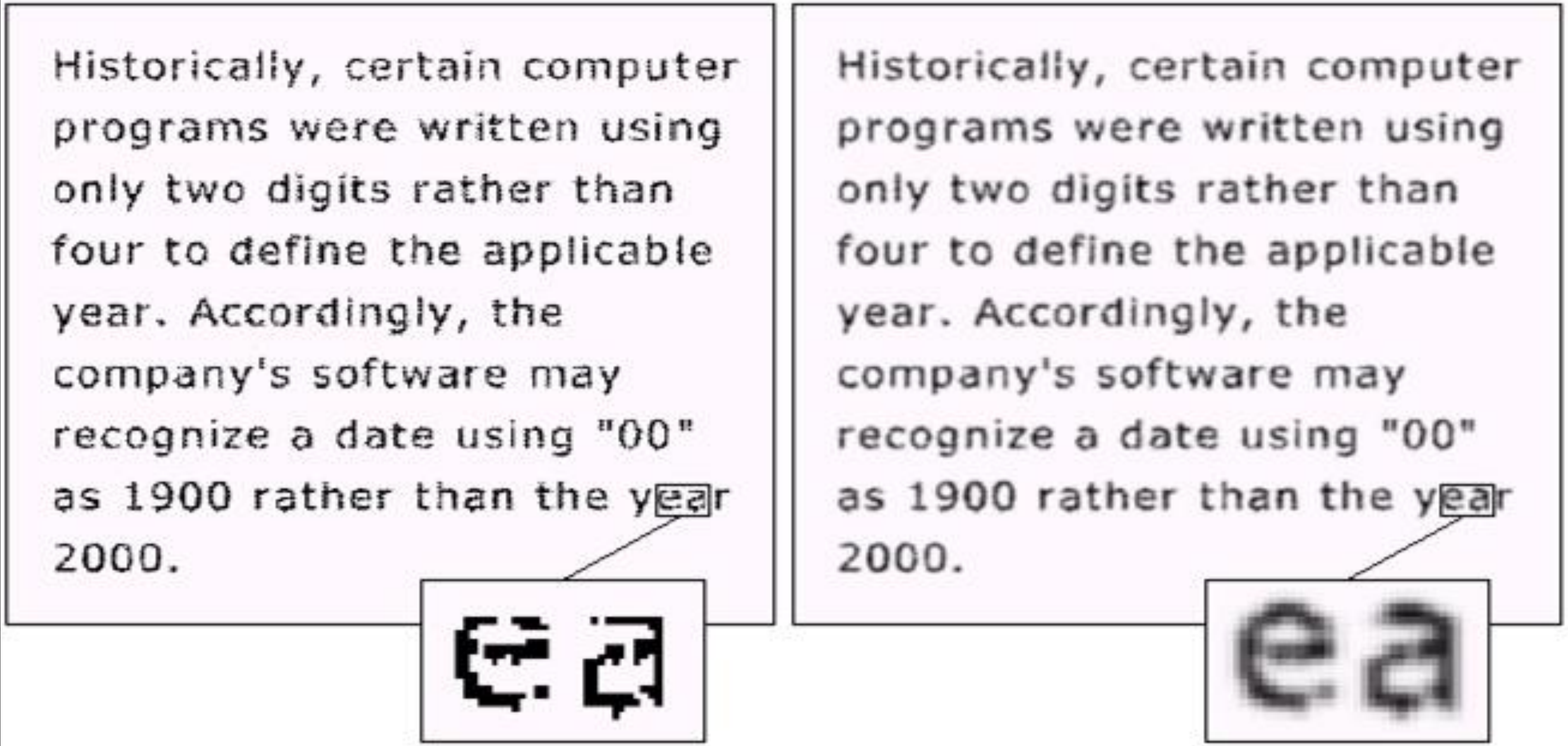
Lowpass Filtering Examples

A low pass Gaussian filter is used to connect broken text



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Lowpass Filtering Examples



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Lowpass Filtering Examples (cont…)

Different low pass Gaussian filters are used to remove blemishes in a photograph

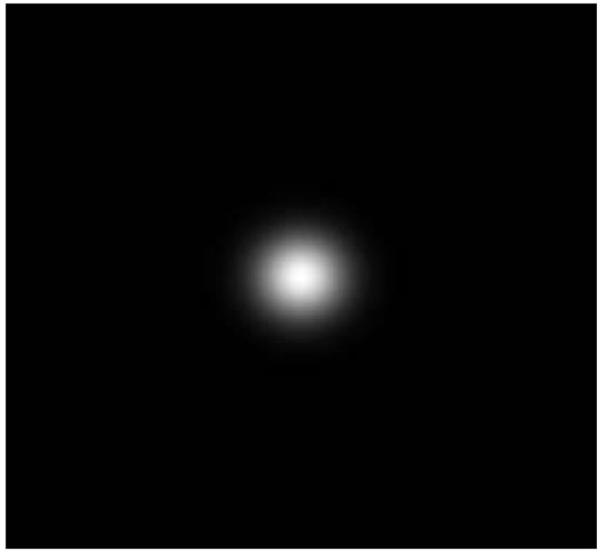
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Lowpass Filtering Examples (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

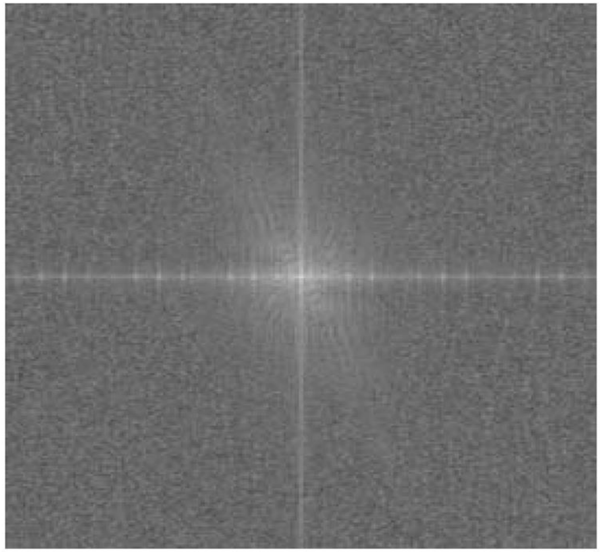
Lowpass Filtering Examples (cont…)

Original image



Gaussian lowpass filter

Spectrum of original image



Processed image

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Sharpening in the Frequency Domain

Edges and fine detail in images are associated with high frequency components

High pass filters – only pass the high frequencies, drop the low ones

High pass filters are precisely the reverse of low pass filters, so:

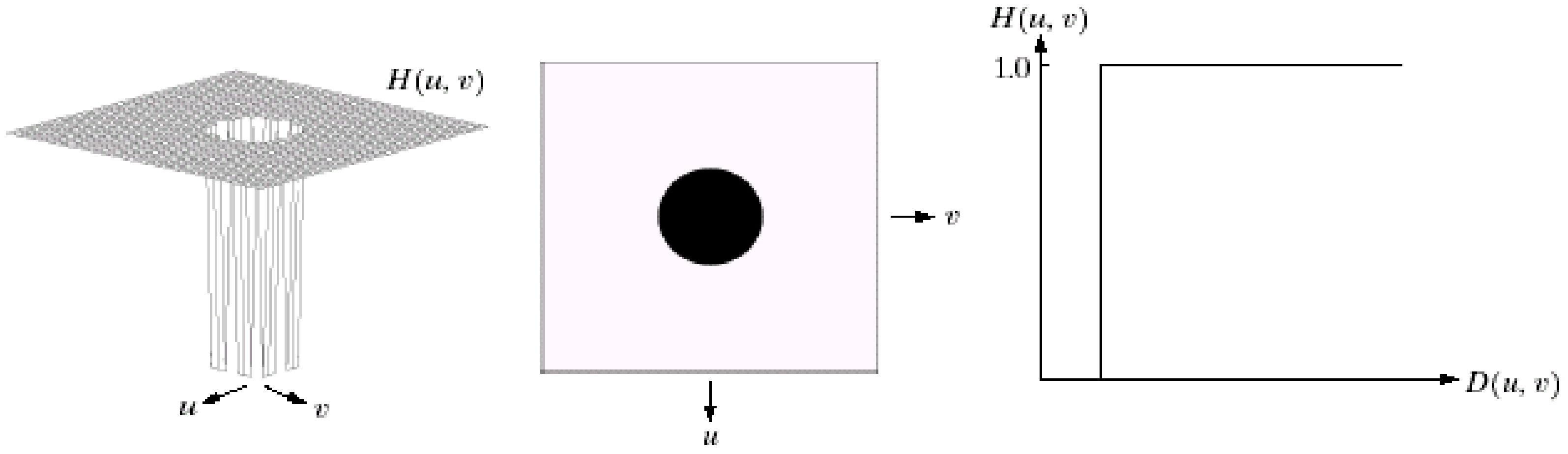
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Ideal High Pass Filters

The ideal high pass filter is given as:

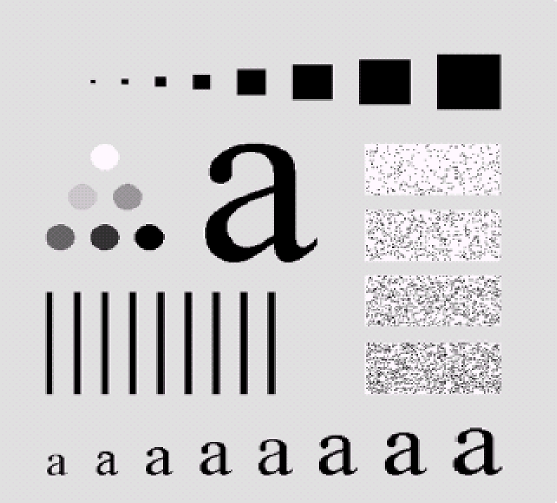
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

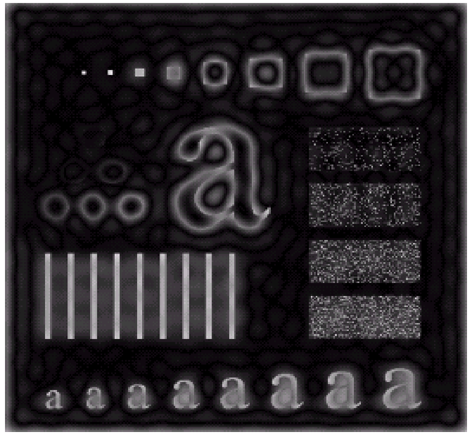
where D_0 is the cut off distance as before



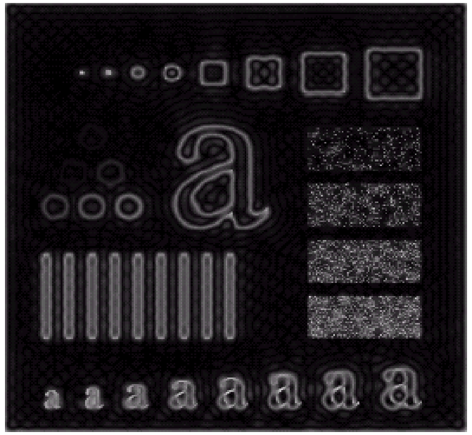
Ideal High Pass Filters (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

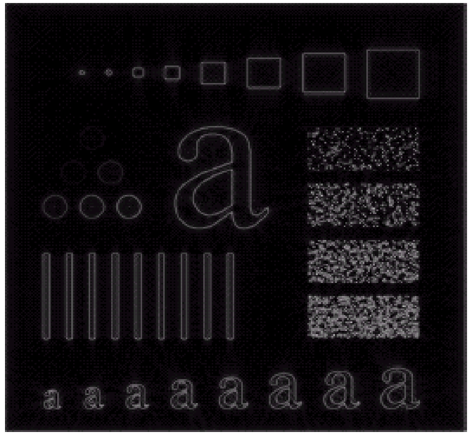




Results of ideal high pass filtering with $D_0 = 15$



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



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

Calculate the spatial domain to the frequency domain to perform Ideal High pass filter, Where cutoff frequency = 0.5

- Input image f(x, y) in a spatial domain.
- Multiply the input image with (-1)^x+y to move the transform in the center.

$$F(x, y) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

$$(-1)^{x+y} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix}$$

Ideal Low pass filter

- After the transform center arrangement we got the results as follows.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix}$$

Ideal Low pass filter: Compute the DFT of the image

- Compute the DFT of the image.
- F(u, v) = kernel * f(x, y) * kernel ^ Transpose

$$f(u,v) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{bmatrix} 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}$$

Ideal Low pass filter: Compute the DFT of the image

$$\Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 4 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Distance measure

$$U = \begin{bmatrix} -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \end{bmatrix} \qquad V = \begin{bmatrix} -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$D(U,V) = \begin{bmatrix} (-2,-2) & (-1,-2) & (0,-2) & (1,-2) \\ (-2,-1) & (-1,-1) & (0,-1) & (1,-1) \\ (-2,0) & (-1,0) & (0,0) & (1,0) \\ (-2,1) & (-1,1) & (0,1) & (1,1) \end{bmatrix}$$

Ideal Low pass filter:

- Now compute the distances
- D(u, v) = sqrt(u.^2 + v.^2)
- D(-2, -2) = (– 2)^2 + (– 2)^2 = (8) ^1/2 = 2.82
- D(-2, -2) = (-1)^2 + (–2)^2 = (5) ^1/2 = 2.23
- D(0, 2) = (0)^2 + (– 2)^2 = (4) ^1/2 = 2
- D(1, -2) = (1)^2 + (– 2)^2 = 2.23
- D(-2, -1) = (– 2)^2 + (– 1)^2 = 1.14
- D(-1, -1) = (-1)^2 + (-1)^2 = 2.23

Ideal Low pass filter: Distance measure

- D(0, -1) = (0)^2 + (-1)^2 = 1
- D(1, -1) = (1)^2 + (-1)^2 = 1.41
- D(-2, 0) = (– 2)^2 + (0)^2 = 2
- D(-1, 0) = (-1)^2 + (0)^2 = 1
- D(0, 0) = (0)^2 + (0)^2 = 0
- D(1, 0) = (1)^2 + (0)^2 = 1
- D(-2, 1) = (– 2)^2 + (1)^2 = 2.23
- D(-1, 1) = (-1)^2 + (1)^2 = 1.41
- D(0, 1) = (0)^2 + (1)^2 = 1
- D(1, 1) = (1)^2 + (1)^2 = 1.41

Ideal Low pass filter: Distance measure

$$D(u,v) = \begin{bmatrix} 2.8284 & 2.2361 & 2 & 2.2361 \\ 2.2361 & 1.4142 & 1 & 1.4142 \\ 2 & 1 & 0 & 1 \\ 2.2361 & 1.4142 & 1 & 1.4142 \end{bmatrix} = \begin{bmatrix} (-2,-2) & (-1,-2) & (0,-2) & (1,-2) \\ (-2,-1) & (-1,-1) & (0,-1) & (1,-1) \\ (-2,0) & (-1,0) & (0,0) & (1,0) \\ (-2,1) & (-1,1) & (0,1) & (1,1) \end{bmatrix}$$

Cut off frequency D0 = 0.5

- Distance matrix will be

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} x \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Cut off frequency D0 = 0.5

$$f(x,y) = \frac{1}{16}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

$$f(x,y) = \frac{1}{16}\begin{bmatrix} 8 & 8 & 8 & 8 \\ -8 & -8 & -8 & -8 \\ 8 & 8 & 8 & 8 \\ -8 & -8 & -8 & -8 \end{bmatrix}$$

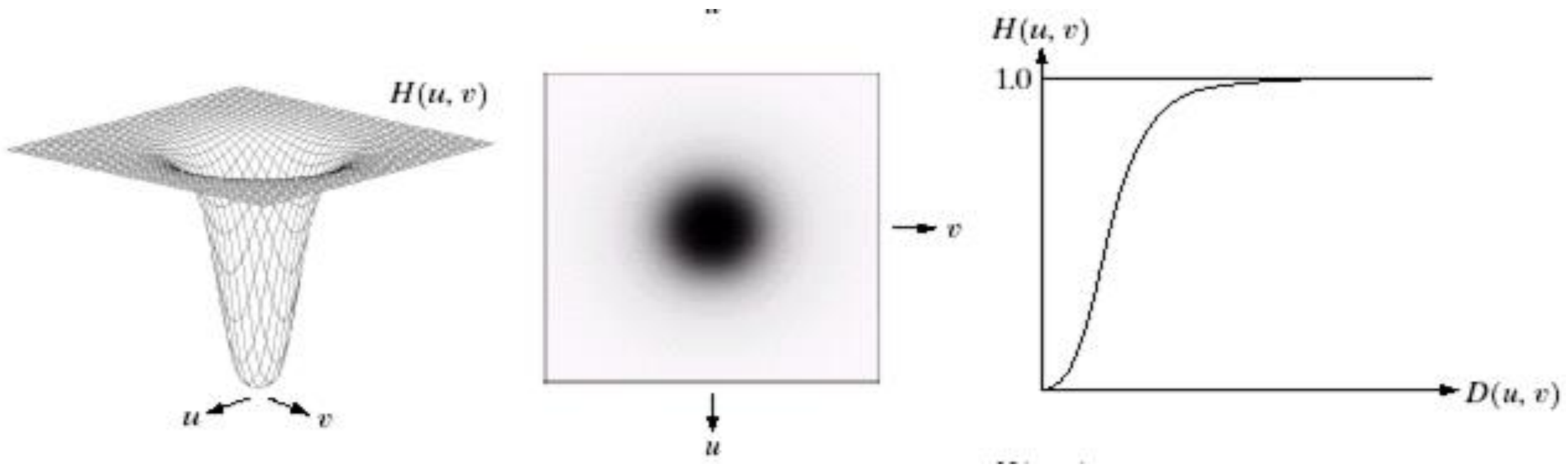
Butterworth High Pass Filters

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

The Butterworth high pass filter is given as:

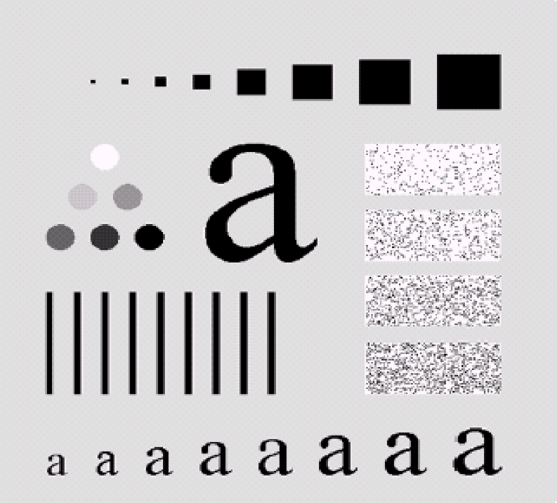
$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

where n is the order and D_0 is the cut off distance as before

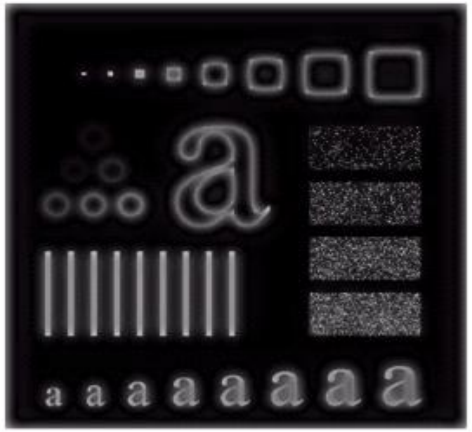


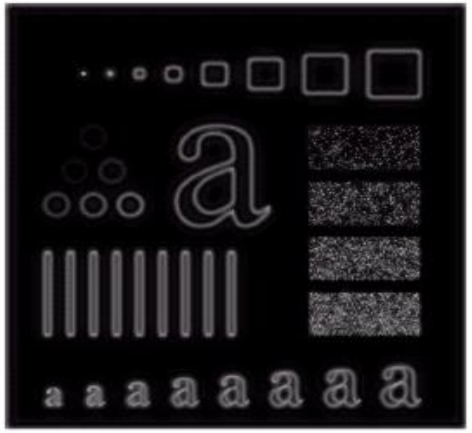
Butterworth High Pass Filters (cont…)

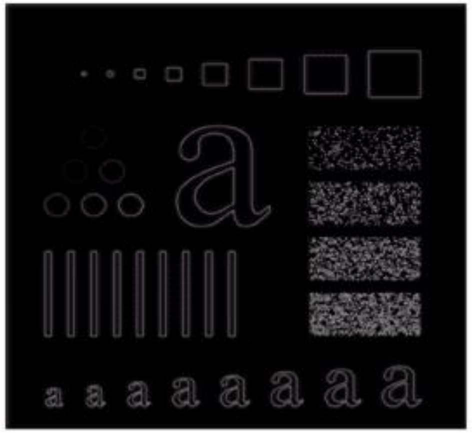
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Results of Butterworth high pass filtering of order 2 with $D_0 = 15$







Results of Butterworth high pass filtering of order 2 with $D_0 = 80$

Results of Butterworth high pass filtering of order 2 with $D_0 = 30$

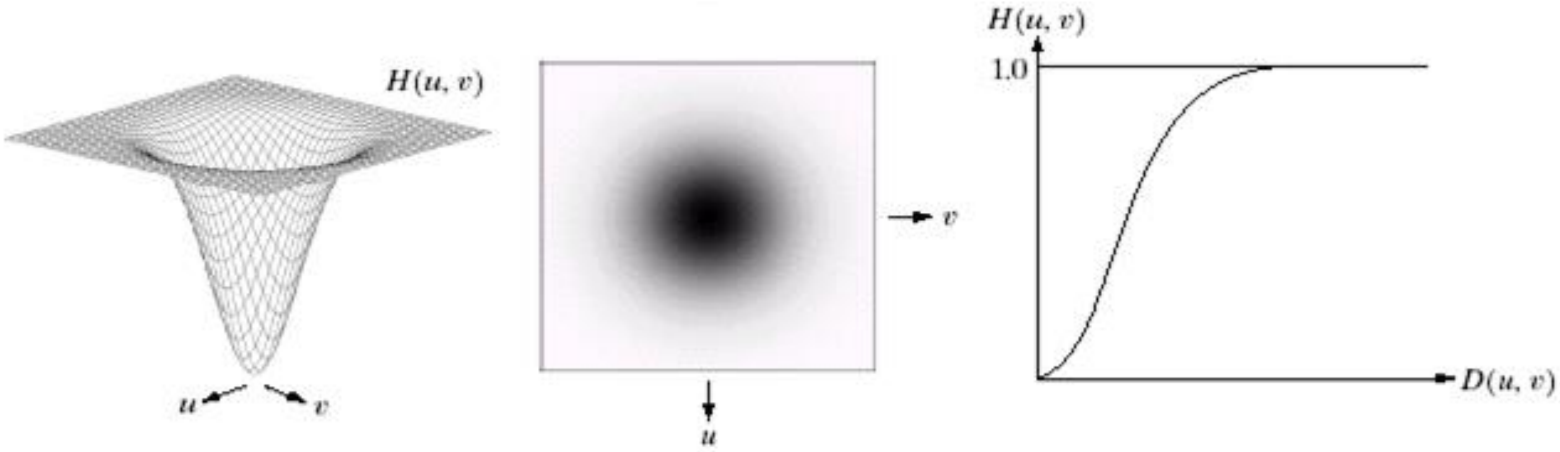
Gaussian High Pass Filters

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

The Gaussian high pass filter is given as:

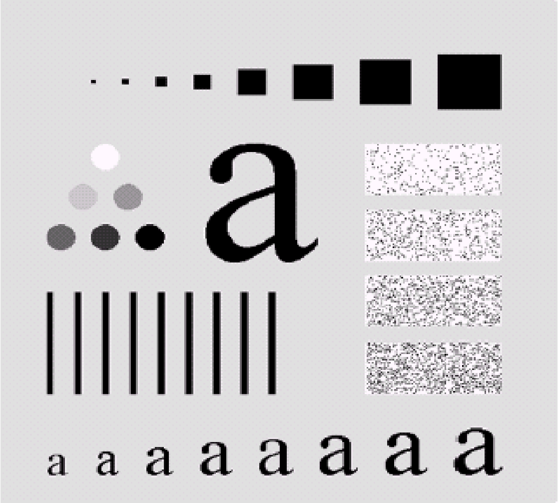
$$H(u,v) = 1 - e^{-D^2(u,v)/2{D_0}^2}$$

where D_0 is the cut off distance as before

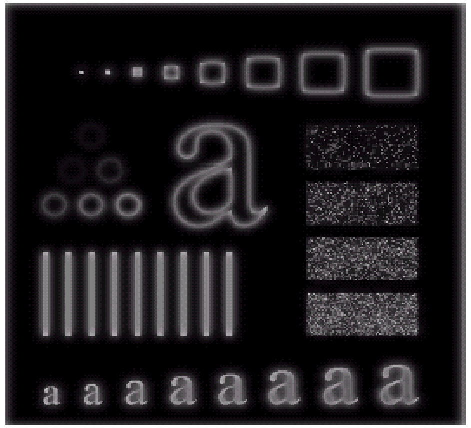


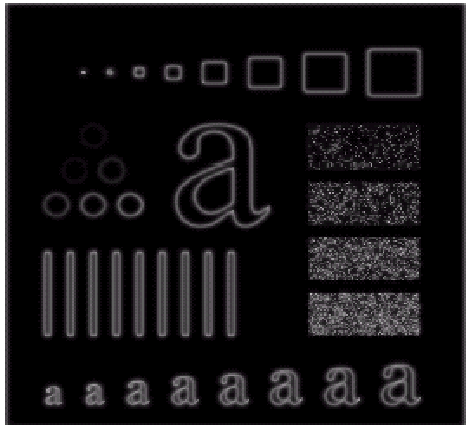
Gaussian High Pass Filters (cont…)

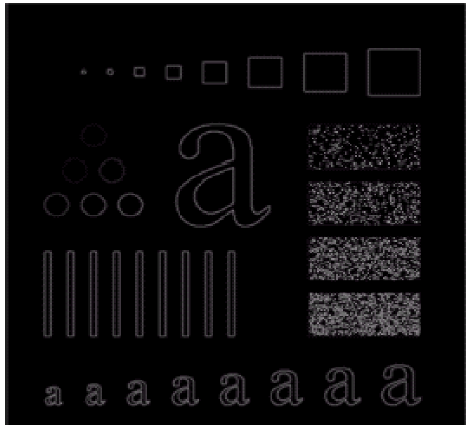
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Results of Gaussian high pass filtering with D_0 = 15



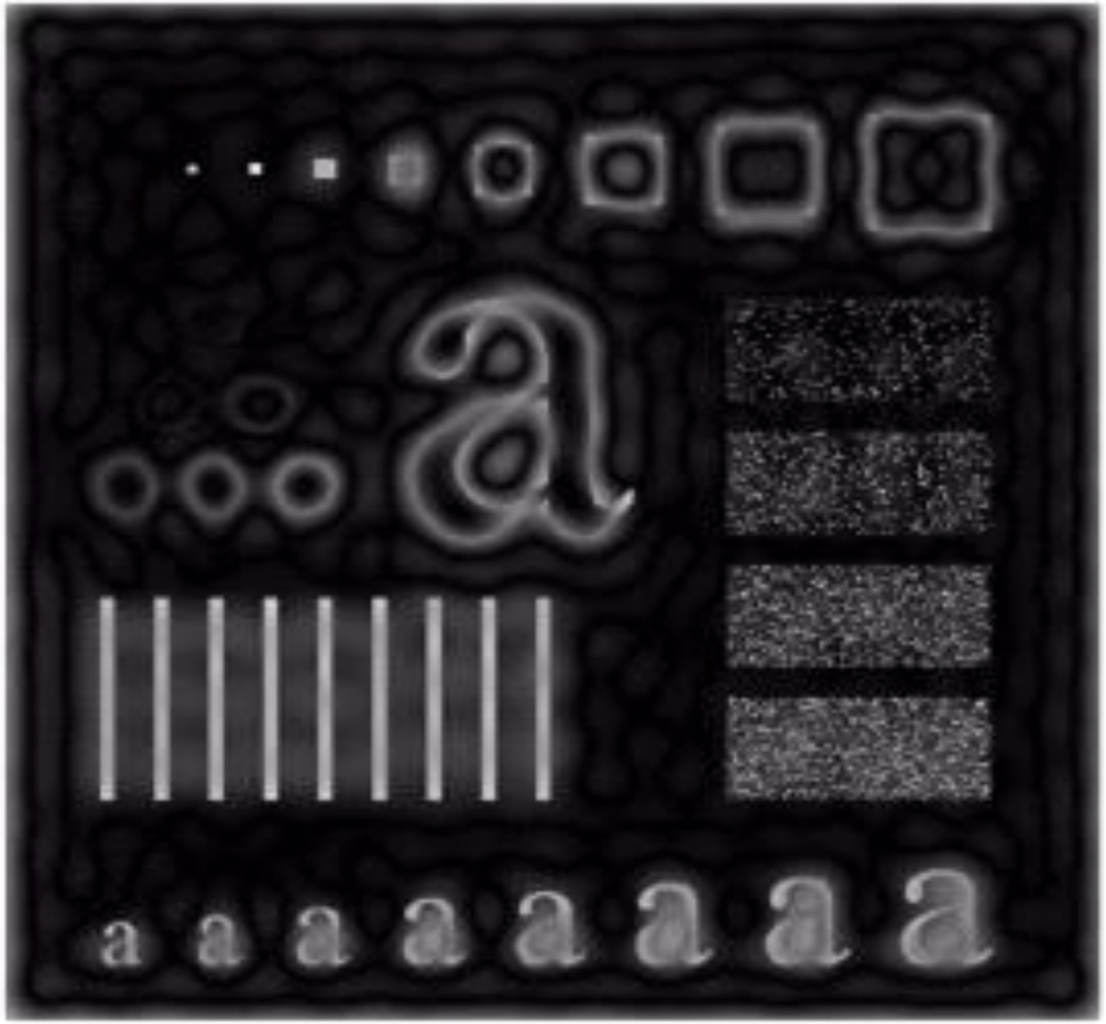




Results of Gaussian high pass filtering with D_0 = 30

Results of Gaussian high pass filtering with D_0 = 80

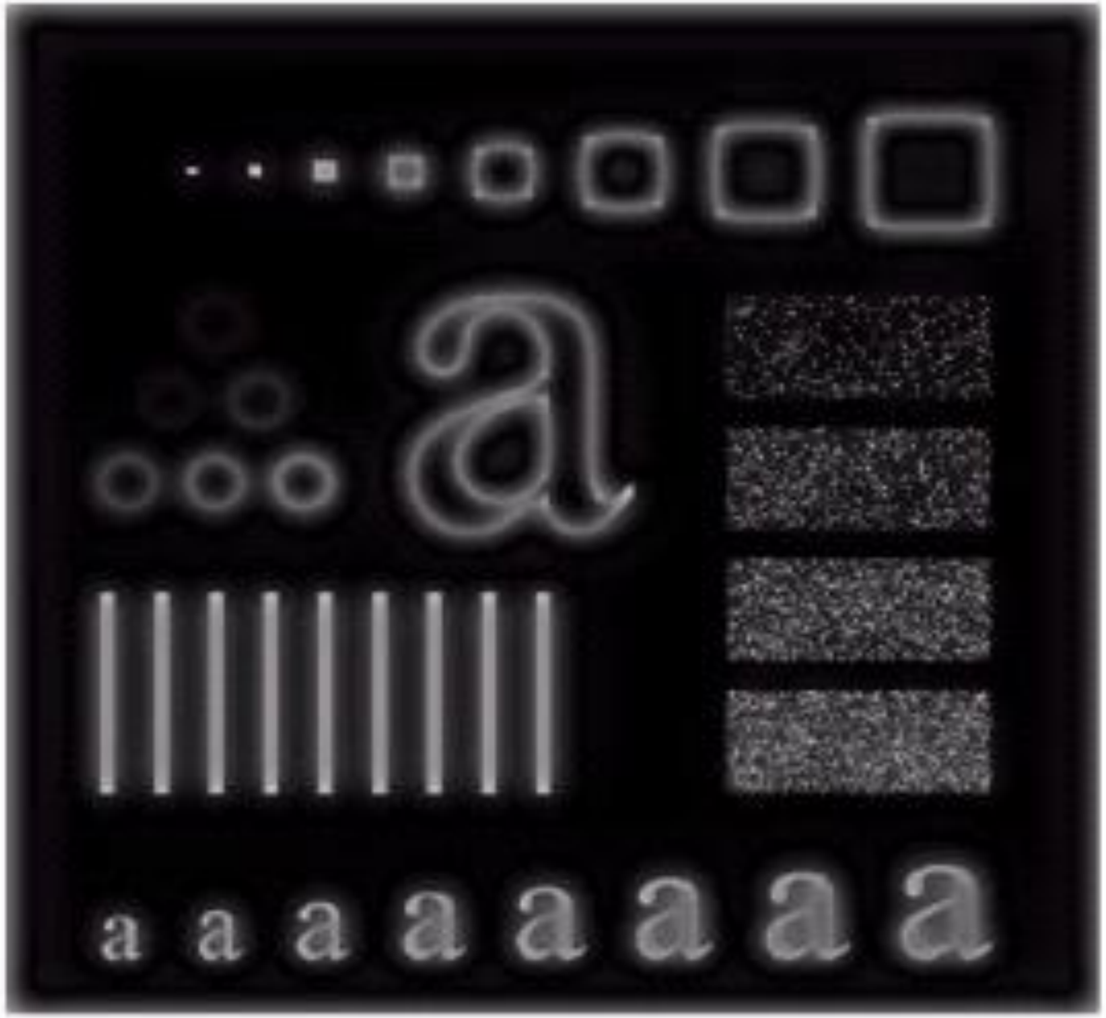
Highpass Filter Comparison



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

Highpass Filter Comparison

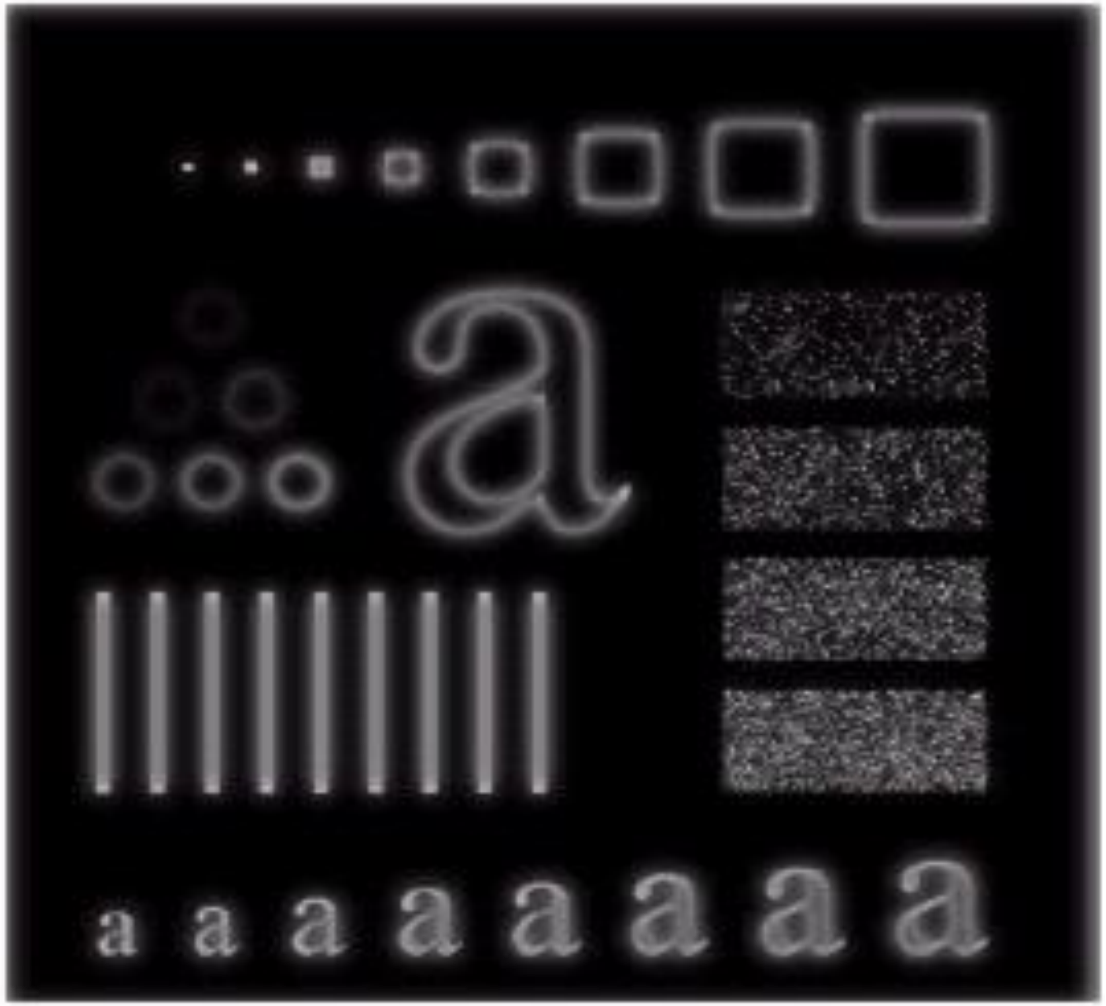
Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Results of Butterworth high pass filtering of order 2 with $D_0 = 15$

Highpass Filter Comparison

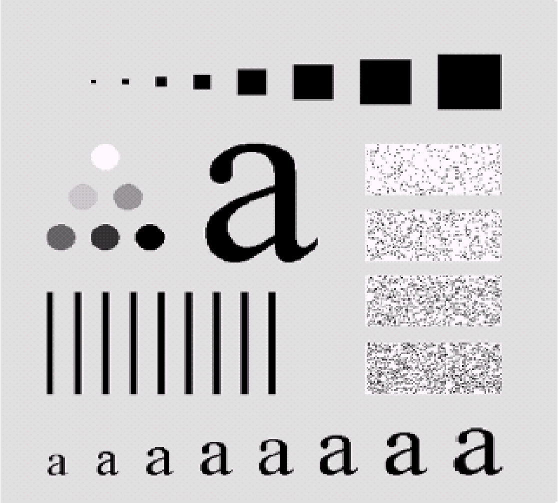
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

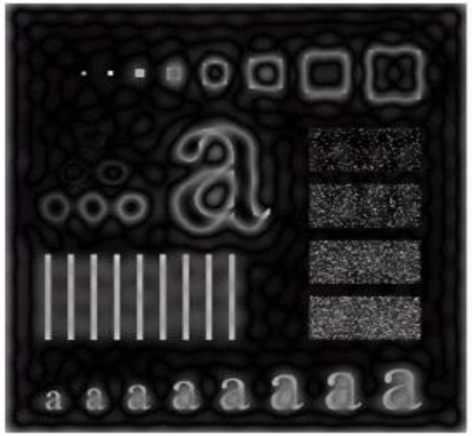


Results of Gaussian high pass filtering with $D_0 = 15$

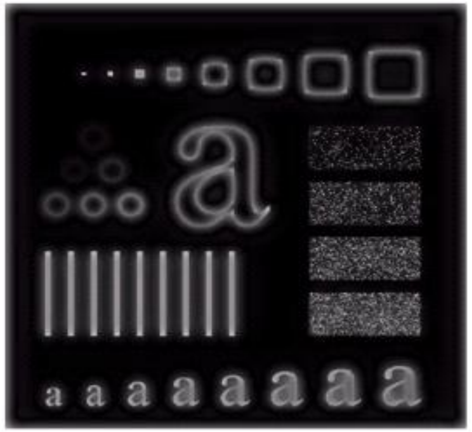
Highpass Filter Comparison

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

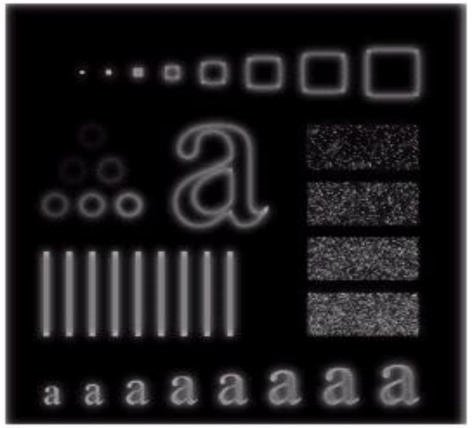




Results of ideal high pass filtering with $D_0 = 15$

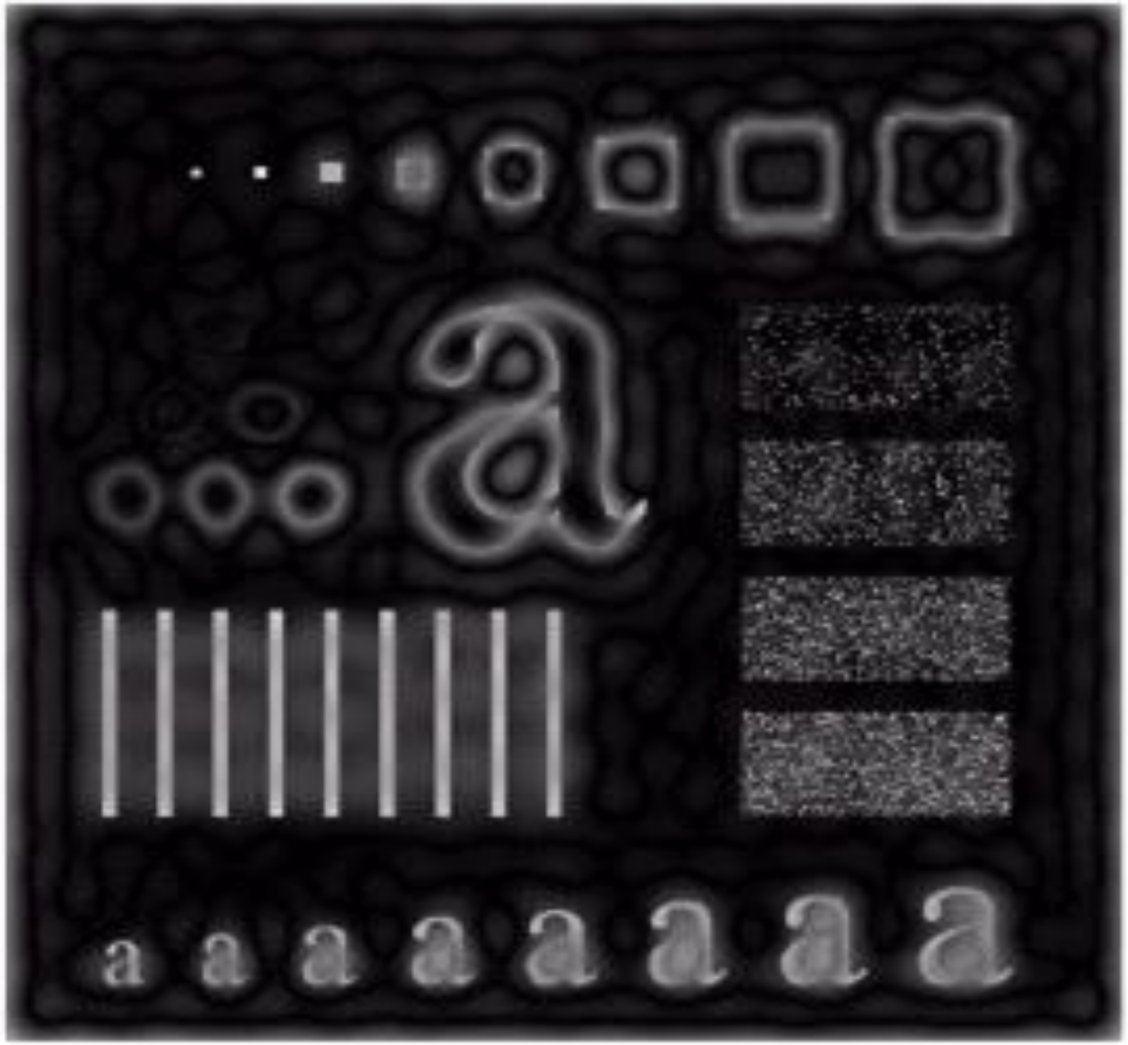


Results of Butterworth high pass filtering of order 2 with $D_0 = 15$



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

Highpass Filter Comparison

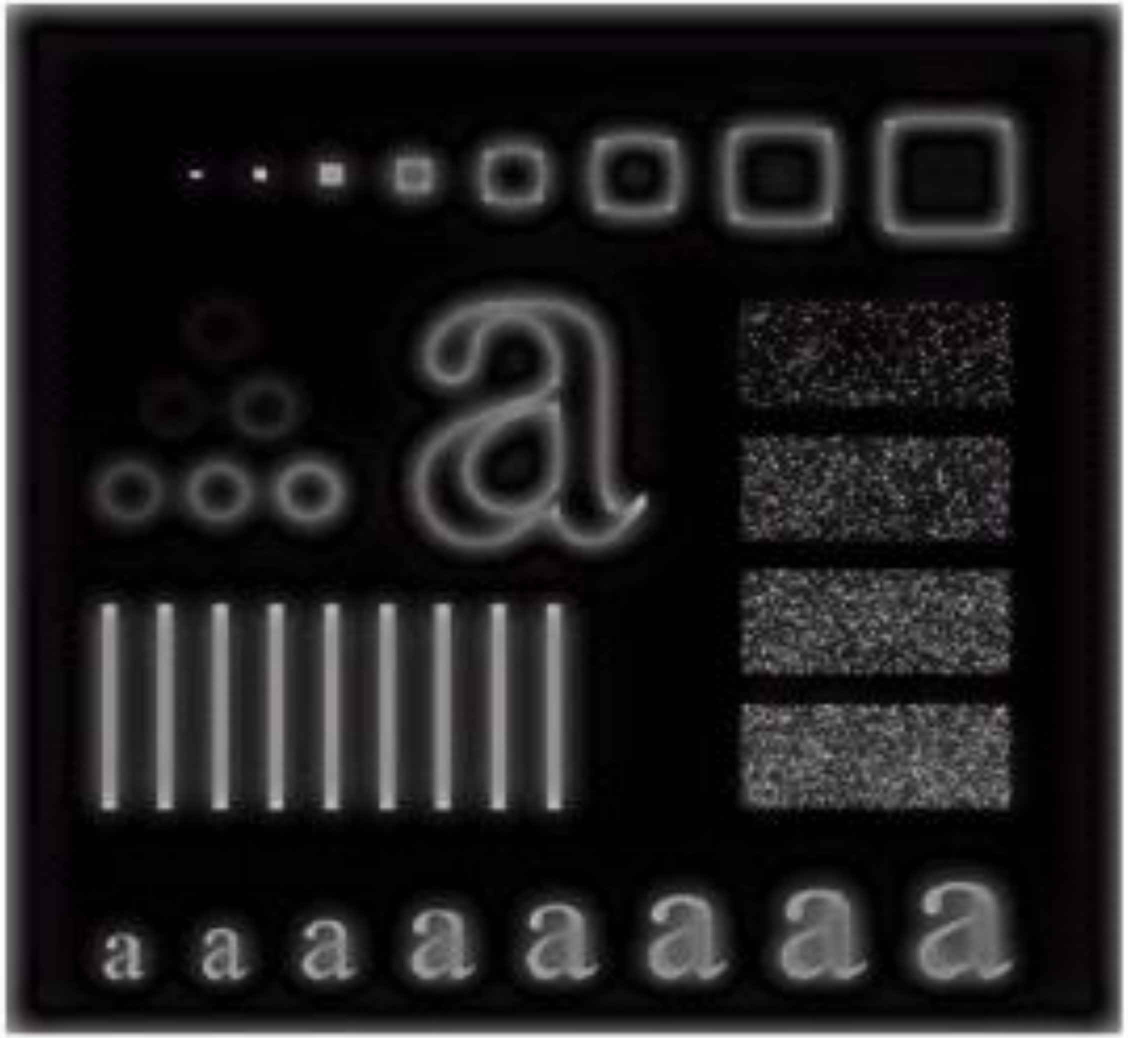


Results of ideal high pass filtering with $D_0 = 15$

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

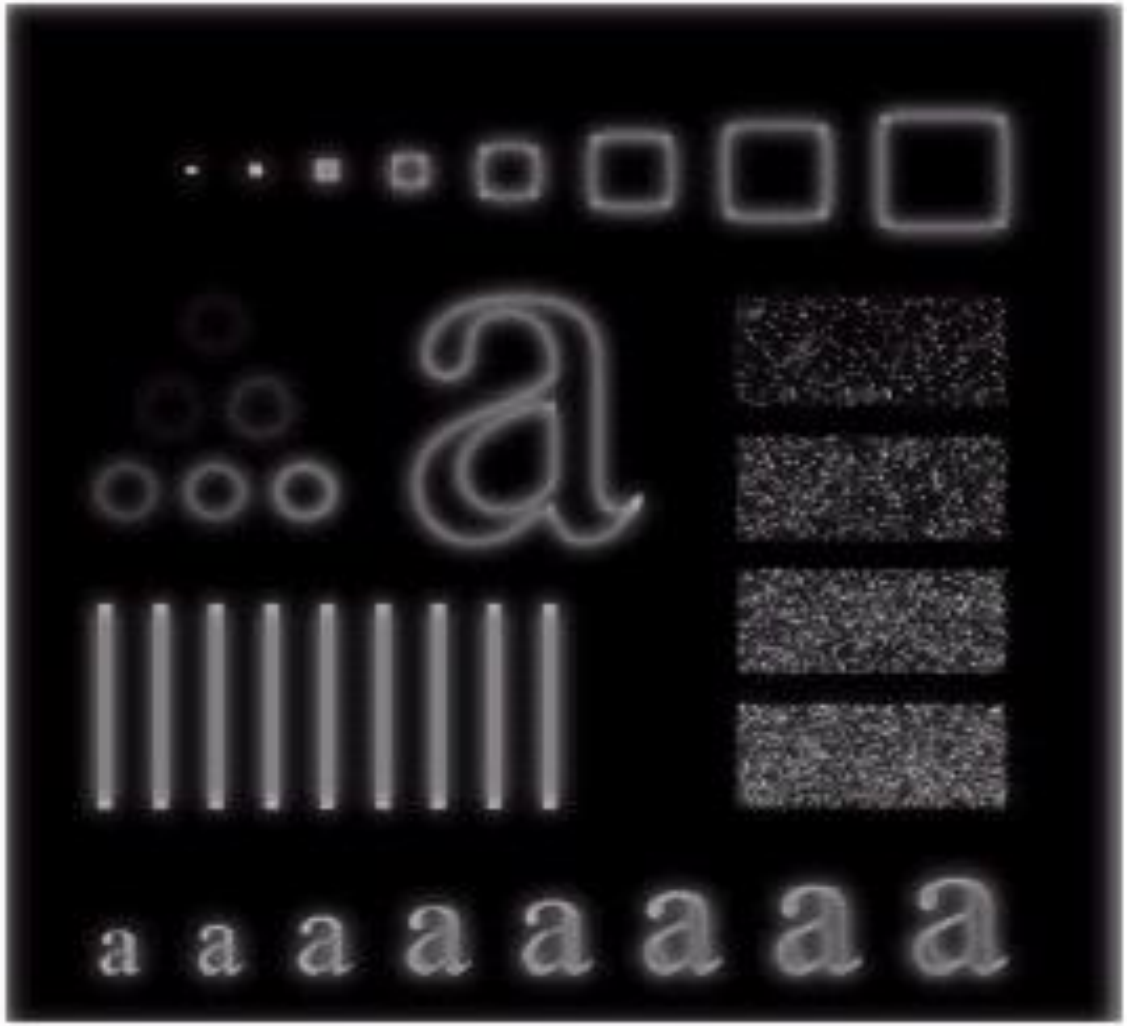
Highpass Filter Comparison

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Results of Butterworth high pass filtering of order 2 with D_0 = 15

Highpass Filter Comparison



Results of Gaussian high pass filtering with D_0 = 15

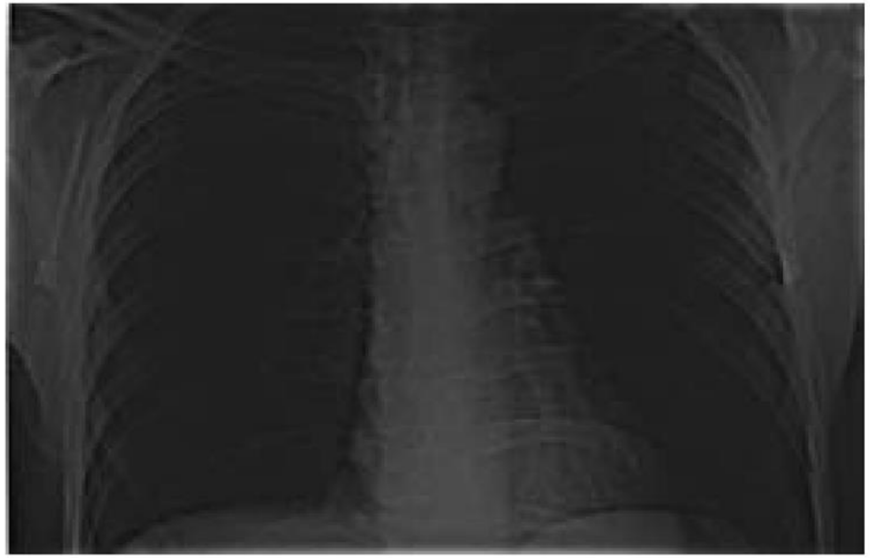
Highpass Filtering Example

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

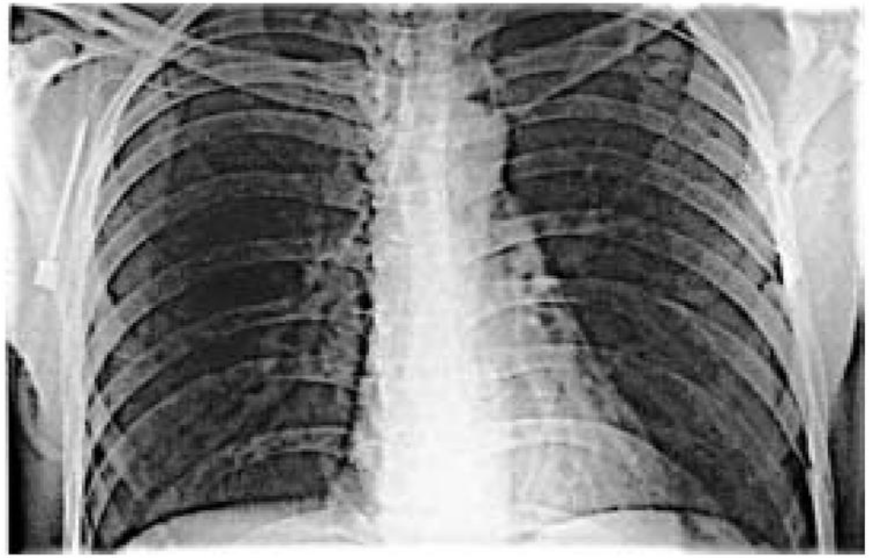
Original image

Highpass filtering result

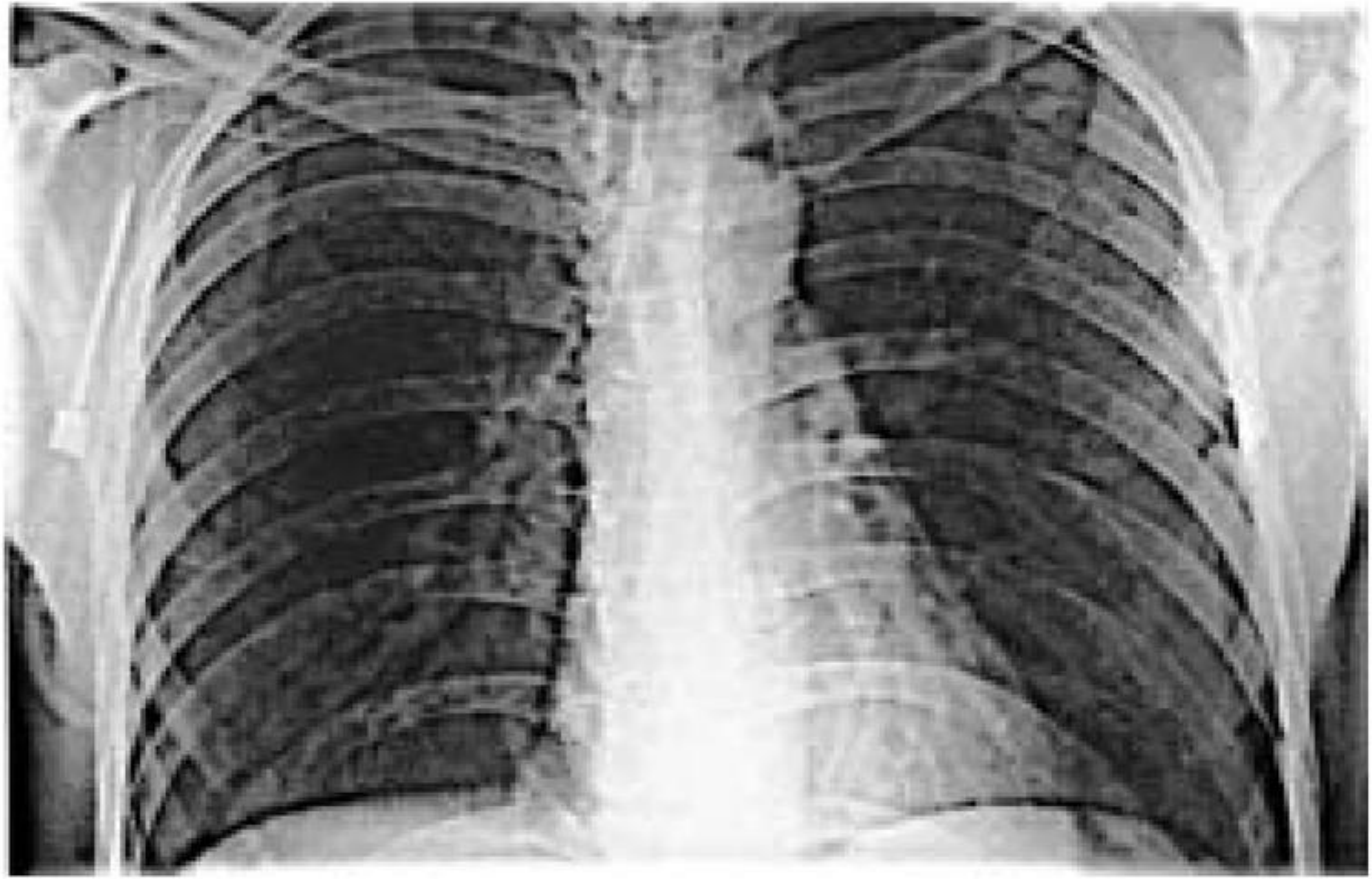
High frequency emphasis result



After histogram equalisation

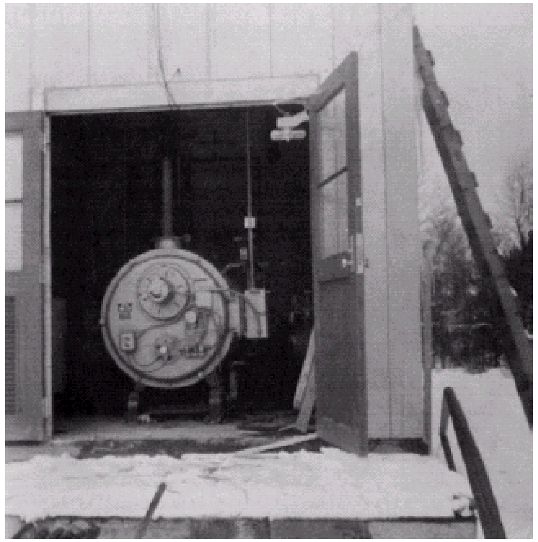






Homomorphic filtering

- Many times, we want to remove shading effects from an image (i.e., due to uneven illumination)
  - Enhance high frequencies
  - Attenuate low frequencies but preserve fine detail.



Homomorphic Filtering (cont'd)

- Consider the following model of image formation:

$$f(x, y) = i(x, y)\, r(x, y)$$

i(x,y): illumination
r(x,y): reflection

- In general, the illumination component i(x,y) varies **slowly** and affects low frequencies mostly.
- In general, the reflection component r(x,y) varies **faster** and affects high frequencies mostly.

<u>IDEA:</u> separate low frequencies due to i(x,y)
from high frequencies due to r(x,y)

How are frequencies mixed together?

- Low and high frequencies from **i(x,y)** and **r(x,y)** are mixed together.

$$f(x, y) = i(x, y)\, r(x, y) \Rightarrow F(u,v) = I(u,v) * R(u,v)$$

- When applying filtering, it is difficult to handle low/high frequencies separately.

$$F(u,v)H(u,v) = [I(u,v) * R(u,v)]H(u,v)$$

Can we separate them?

- Idea:

Take the ln( ) of $f(x, y) = i(x, y)\, r(x, y)$

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

Steps of Homomorphic Filtering

(1) Take $ln(f(x,y)) = ln(i(x,y)) + ln(r(x,y))$

(2) Apply FT: $F(ln(f(x,y)))=F(ln(i(x,y)))+F(ln(r(x,y)))$

or $Z(u,v) = Illum(u,v) + Refl(u,v)$

(3) Apply H(u,v)

$$Z(u,v)H(u,v) = Illum(u,v)H(u,v) + Refl(u,v)H(u,v)$$

Steps of Homomorphic Filtering (cont'd)

(4) Take Inverse FT:

$$F^{-1}(Z(u,v)H(u,v)) = F^{-1}(Illum(u,v)H(u,v)) + F^{-1}(Refl(u,v)H(u,v))$$

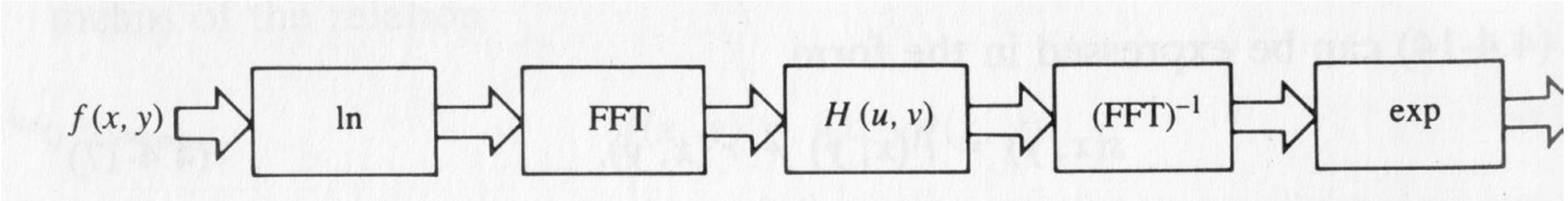
$$s(x,y) = i'(x,y) + r'(x,y)$$

or

(5) Take exp( )

$$e^{s(x,y)} = e^{i'(x,y)} e^{r'(x,y)}$$

$$g(x,y) = i_0(x,y) r_0(x,y)$$



Example using high-frequency emphasis

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[(u^2+v^2)/D_0^2\right]}\right] + \gamma_L$$

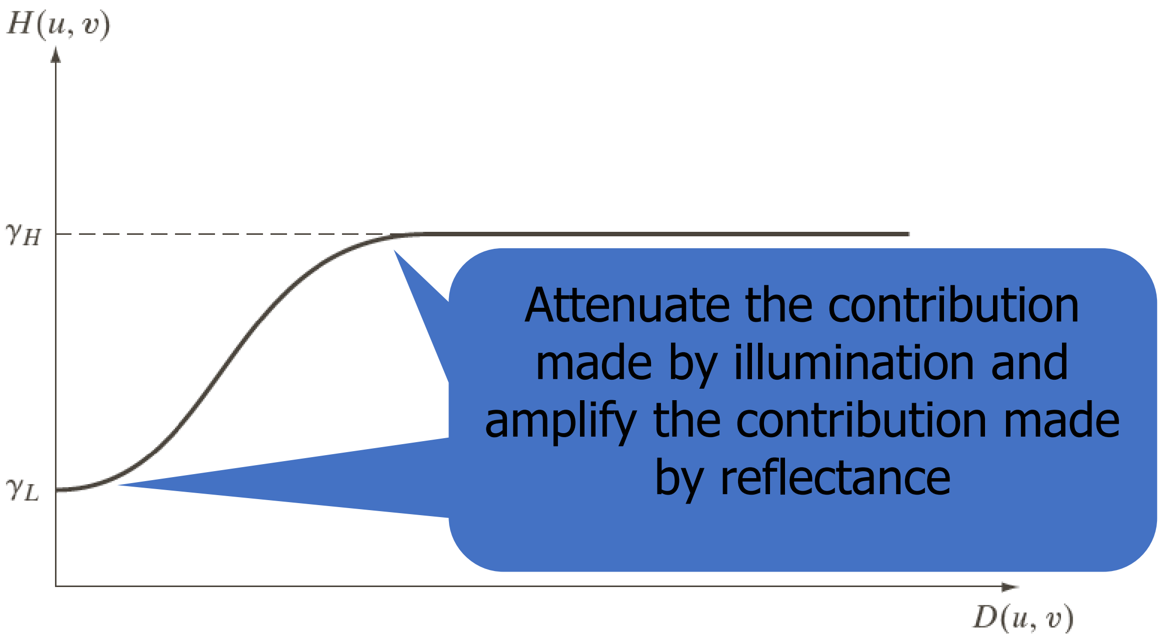


FIGURE 4.61 Radial cross section of a circularly symmetric homomorphic filter function. The vertical axis is at the center of the frequency rectangle and $D(u, v)$ is the distance from the center.

Example 1 homomorphic filters

$$I = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

```
gamma_l = 0.5;
gamma_h = 1.3;
cutoff = 50;
```

→ Parameters

```
f1 = log(1+img);
```

$$\begin{bmatrix} 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \end{bmatrix}$$

Example 1 homomorphic filters

- After the transform center arrangement we got the results as follows.

$$\begin{bmatrix} 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0.6931 & 0 & 0.6931 & 0 \\ -0.6931 & 0 & -0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ -0.6931 & 0 & -0.6931 & 0 \end{bmatrix}$$

Example 1 homomorphic filters

- Compute the DFT of the image.
- F(u, v) = kernel * f(x, y) * kernel ^ Transpose

$$f(u,v) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix} \times \begin{bmatrix} 0.6931 & 0 & 0.6931 & 0 \\ -0.6931 & 0 & -0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ -0.6931 & 0 & -0.6931 & 0 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}$$

Example 1 homomorphic filters

$$f(u,v) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & -0.6931 & 0 \\ 0.6931 & 0 & 0.6931 & 0 \\ 0.6931 & 0 & -0.6931 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$f(u,v) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 5.5452 & 0 & 5.5452 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Distance measure

$$U = \begin{bmatrix} -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \\ -2 & -1 & 0 & 1 \end{bmatrix} \qquad V = \begin{bmatrix} -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$D(U,V) = \begin{bmatrix} (-2,-2) & (-1,-2) & (0,-2) & (1,-2) \\ (-2,-1) & (-1,-1) & (0,-1) & (1,-1) \\ (-2,0) & (-1,0) & (0,0) & (1,0) \\ (-2,1) & (-1,1) & (0,1) & (1,1) \end{bmatrix}$$

Distance measure

- Now compute the distances
- D(u, v) = sqrt(u.^2 + v.^2)
- D(-2, -2) = (– 2)^2 + (– 2)^2 = (8) ^1/2 = 2.82
- D(-2, -2) = (-1)^2 + (–2)^2 = (5) ^1/2 = 2.23
- D(0, 2) = (0)^2 + (– 2)^2 = (4) ^1/2 = 2
- D(1, -2) = (1)^2 + (– 2)^2 = 2.23
- D(-2, -1) = (– 2)^2 + (– 1)^2 = 1.14
- D(-1, -1) = (-1)^2 + (-1)^2 = 2.23

Distance measure

- D(0, -1) = (0)^2 + (-1)^2 = 1
- D(1, -1) = (1)^2 + (-1)^2 = 1.41
- D(-2, 0) = (– 2)^2 + (0)^2 = 2
- D(-1, 0) = (-1)^2 + (0)^2 = 1
- D(0, 0) = (0)^2 + (0)^2 = 0
- D(1, 0) = (1)^2 + (0)^2 = 1
- D(-2, 1) = (– 2)^2 + (1)^2 = 2.23
- D(-1, 1) = (-1)^2 + (1)^2 = 1.41
- D(0, 1) = (0)^2 + (1)^2 = 1
- D(1, 1) = (1)^2 + (1)^2 = 1.41

$$\boxed{H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[(u^2+v^2)/D_0^2\right]}\right] + \gamma_L}$$

$$D(u,v) = \begin{bmatrix} 2.8284 & 2.2361 & 2 & 2.2361 \\ 2.2361 & 1.4142 & 1 & 1.4142 \\ 2 & 1 & 0 & 1 \\ 2.2361 & 1.4142 & 1 & 1.4142 \end{bmatrix} = \begin{bmatrix} (-2,-2) & (-1,-2) & (0,-2) & (1,-2) \\ (-2,-1) & (-1,-1) & (0,-1) & (1,-1) \\ (-2,\ 0) & (-1,\ 0) & (0,\ 0) & (1,\ 0) \\ (-2,\ 1) & (-1,\ 1) & (0,\ 1) & (1,\ 1) \end{bmatrix}$$

```
H = (gamma_h - gamma_l) .* (1 - exp(-c .* (D_uv.^2 / (D0^2)))) + gamma_l;
```

HMF

$$h(u,v) = \begin{bmatrix} 0.5026 & 0.5016 & 0.5013 & 0.5016 \\ 0.5016 & 0.5006 & 0.5003 & 0.5006 \\ 0.5013 & 0.5003 & 0.5000 & 0.5003 \\ 0.5016 & 0.5006 & 0.5003 & 0.5006 \end{bmatrix}$$

Example 1 homomorphic filters

$$g(u,v) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 5.5452 & 0 & 5.5452 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.5026 & 0.5016 & 0.5013 & 0.5016 \\ 0.5016 & 0.5006 & 0.5003 & 0.5006 \\ 0.5013 & 0.5003 & 0.5000 & 0.5003 \\ 0.5016 & 0.5006 & 0.5003 & 0.5006 \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2.7797 & 0 & 2.7726 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Example 1 homomorphic filters

$$DFT^{-1} = \begin{bmatrix} 5.5523 & 0.0071 & 5.5523 & 0.0071 \\ -5.5523 & -0.0071 & -5.5523 & -0.0071 \\ 5.5523 & 0.0071 & 5.5523 & 0.0071 \\ -5.5523 & -0.0071 & -5.5523 & -0.0071 \end{bmatrix}$$

```
g1 = exp(DFT_Inverse) - 1;
```

$$g_1 = \begin{bmatrix} 256.8220 & 0.0071 & 256.8220 & 0.0071 \\ -0.9961 & -0.0071 & -0.9961 & -0.0071 \\ 256.8220 & 0.0071 & 256.8220 & 0.0071 \\ -0.9961 & -0.0071 & -0.9961 & -0.0071 \end{bmatrix} \Rightarrow$$

```
g2 = g1 ./ max(g1(:));
```

$$\begin{bmatrix} 1.0000 & 0.0000 & 1.0000 & 0.0000 \\ -0.0039 & -0.0000 & -0.0039 & -0.0000 \\ 1.0000 & 0.0000 & 1.0000 & 0.0000 \\ -0.0039 & -0.0000 & -0.0039 & -0.0000 \end{bmatrix}$$

Resultant HMF image

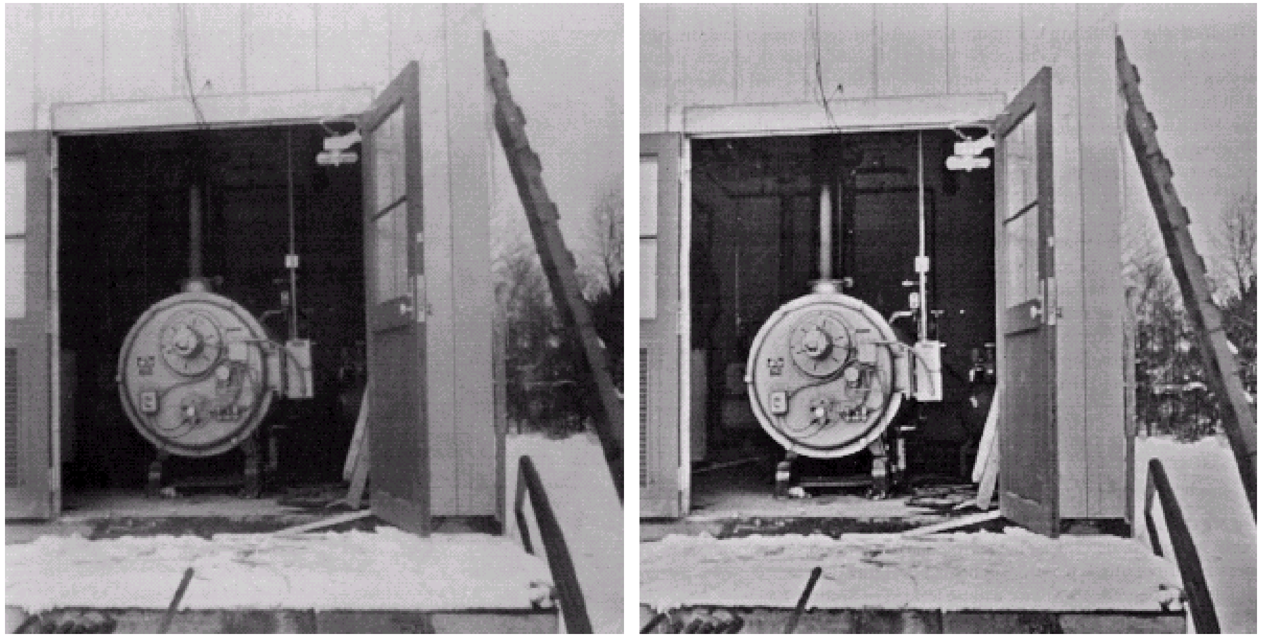
Homomorphic Filtering

- In homomorphic filtering, gamma_h and gamma_l are parameters used to control the high-frequency and low-frequency emphasis of the filter.
- Gamma_h, also known as the high-frequency gamma, is a parameter that controls the emphasis given to the high-frequency components of the image. Increasing gamma_h results in a stronger emphasis on the high-frequency components, making the image appear sharper and more detailed.
- Gamma_l, also known as the low-frequency gamma, is a parameter that controls the emphasis given to the low-frequency components of the image. Increasing gamma_l results in a stronger emphasis on the low-frequency components, making the image appear smoother and less detailed.
- The choice of gamma_h and gamma_l values depends on the specific application and the characteristics of the image being processed. In general, higher values of gamma_h and lower values of gamma_l are used to enhance the edges and details in the image, while lower values of gamma_h and higher values of gamma_l are used to smooth out the image and reduce noise.

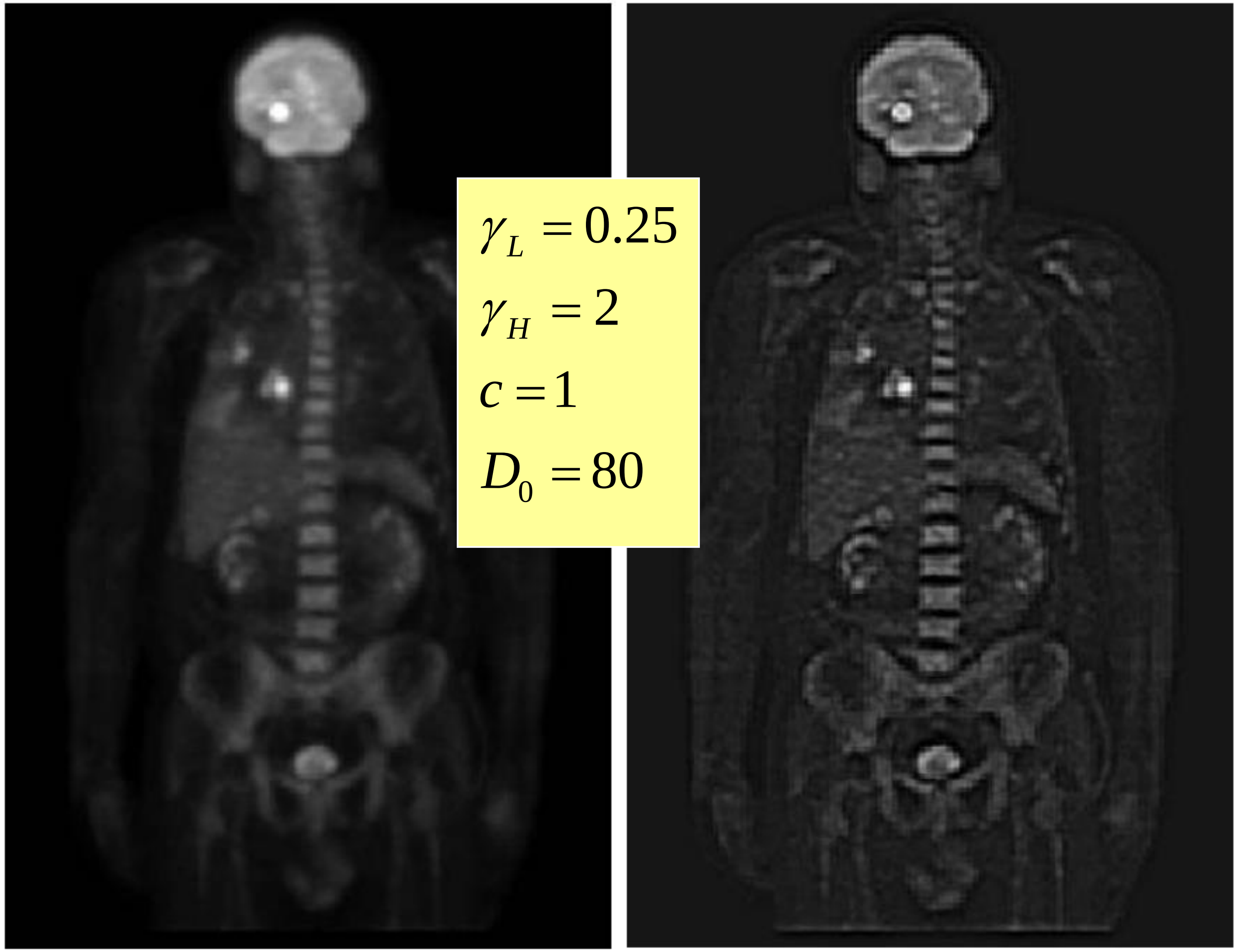
Homomorphic Filtering: Example

a b

FIGURE 4.33
(a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)



Homomorphic Filtering: Example



$$\gamma_L = 0.25$$
$$\gamma_H = 2$$
$$c = 1$$
$$D_0 = 80$$

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.)

Summery of the lecture

- Frequency domain Filters
- Ideal Lowpass and High pass Filters
- Butterworth Lowpass and High pass Filters
- Gaussian Lowpass and High pass Filters
- Homomorphic Filtering