Claire Mantel
09/07/2024

34269
–
#2 Intro to digital image processing 2/2

Outline

Spatial filtering

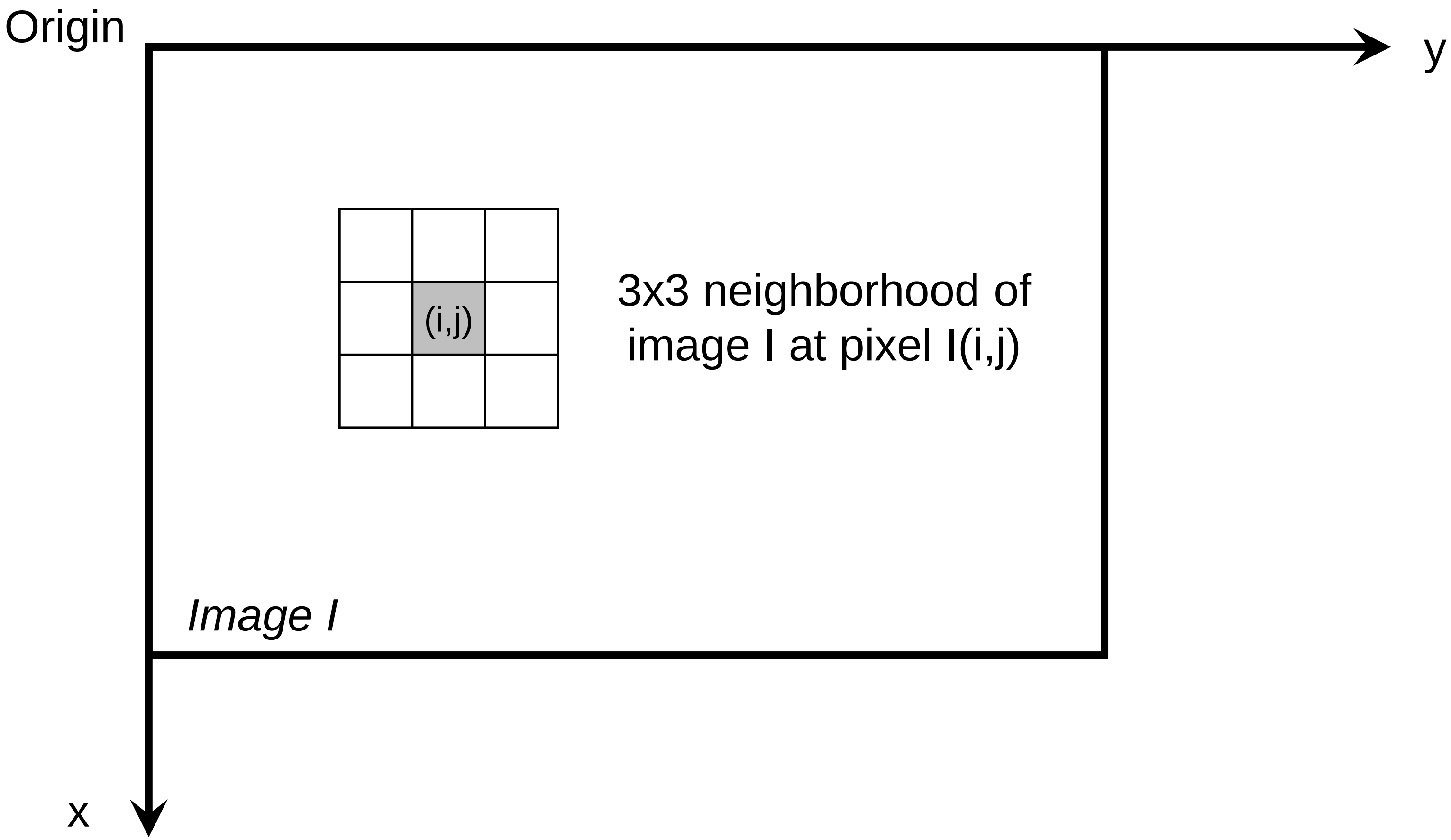
- Principle
- Process
- Applications: smoothing and sharpening with spatial filter

Frequency domain filtering

- Principle and process
- Classic design for smoothing/low-pass filtering, high-pass filtering, band-pass/band-reject filters

Principle

- Neighbourood



Principle

- Neighbourood
- Linear combination

(i-1, j-1)	(i-1, j)	(i-1, j+1)
(i, j-1)	(i,j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

3x3 neighborhood of
image I at pixel I(i,j)

$$g(i,j) = w(-1, -1)I(i-1, j-1) + \ldots$$
$$w(0, -1)I(i, j-1)+ w(0, 0)I(i, j) + \ldots$$
$$\ldots + w(1, 1)I(i+1, j+1)$$

Principle

- Neighbourood
- Linear combination

$$g(i,j) = w(-1,-1)I(i-1,j-1) + \ldots$$
$$w(0,-1)I(i,j-1) + w(0,0)I(i,j) + \ldots$$
$$\ldots + w(1,1)I(i+1,j+1)$$

(i-1, j-1)	(i-1, j)	(i-1, j+1)
(i, j-1)	(i,j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

3x3 neighborhood
of image I at (i,j)

w(-1,-1)	*w(-1,0)*	*w(-1,1)*
w(0, -1)	*w(0,0)*	*w(0,1)*
w(1,-1)	*w(1, 0)*	*w(1,1)*

Mask, or kernel,
or filter

Principle

- Neighbourood
- Linear combination

(i-1, j-1)	(i-1,j)	(i-1, j+1)
(i, j-1)	(i,j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

3x3 neighborhood
of image I at (i,j)

w(-1,-1)	*w(-1,0)*	*w(-1,1)*
w(0, -1)	*w(0,0)*	*w(0,1)*
w(1,-1)	*w(1, 0)*	*w(1,1)*

Mask
(or kernel or filter)

g(i,j) = w(-1, -1)I(i-1, j-1) + ...
w(0, -1)I(i, j-1)+ w(0, 0)I(i, j) + ...
... + w(1, 1)I(i+1, j+1)

$$g(i,j) = \sum_{-1}^{1}\sum_{-1}^{1} w(k,l)I(i+k,j+l)$$

Principle

w(-1,-1)	*w(-1,0)*	*w(-1,1)*
w(0, -1)	*w(0,0)*	*w(0,1)*
w(1,-1)	*w(1, 0)*	*w(1,1)*

Mask
(or kernel or filter)

Correlation $$g(i,j) = \sum_{-1}^{1}\sum_{-1}^{1} w(k,l)I(i+k,j+l)$$

$$g(i,j) = \sum_{-1}^{1}\sum_{-1}^{1} w(-k,-l)I(i-k,j-l)$$

Convolution $$g(i,j) = \sum_{-1}^{1}\sum_{-1}^{1} w_{180}(k,l)I(i-k,j-l)$$

Process

1	2	3
4	5	6
7	8	9

Mask w

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Padded image

Mask size m x n
Padding m-1 rows at top and bottom
and n-1 colomns at left and right

Process - correlation

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Padded image

1	2	3	0	0	0	0	0	0
4	5	6	0	0	0	0	0	0
7	8	9	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

First location

Process - correlation

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f (2D discrete unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Padded image

0	1	2	3	0	0	0	0	0
0	4	5	6	0	0	0	0	0
0	7	8	9	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Second location

Process - correlation

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	2	3	0	0	0	0
0	0	4	5	6	0	0	0	0
0	0	7	8	9	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

First non-zero location

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Output padded image

Process - correlation

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Second non-zero location

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Output padded image

Process - correlation

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	9	8	7	0	0	0
0	0	0	6	5	4	0	0	0
0	0	0	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Output padded image

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

Output filtered image
(cropped)

Process - convolution

Mask w

1	2	3
4	5	6
7	8	9

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Input image f
(2D discrete
unit impulse)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	1	2	3	0	0	0
0	0	0	4	5	6	0	0	0
0	0	0	7	8	9	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Output padded image

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Output filtered image
(cropped)

Convolution - Properties

- Notation $$g(i,j) = \sum_{-1}^{1}\sum_{-1}^{1} m(k,l)I(i-k,j-l) = M \star I$$

- Commutative $M \star I = I \star M$ (not true for correlation)

- Associative $M1 \star (M2 \star I) = (M1 \star M2) \star I$ (not true for correlation)

  $\Rightarrow$ Apply several filters in sequence

- Linearity $$M \star (I + J) = (M \star I) + (M \star J)$$ $$(kM) \star I = k(M \star I) = M \star (kI)$$

- Fully characterized by its impulse response

Usage - smoothing

- Box filter (averaging)

$$g(i,j) = \frac{1}{MN} \sum_{-a}^{a} \sum_{-b}^{b} I(i+k, j+l)$$

with $M = 2a + 1$
and $N = 2b + 1$

For $M = N = 3$,
the convolution mask is

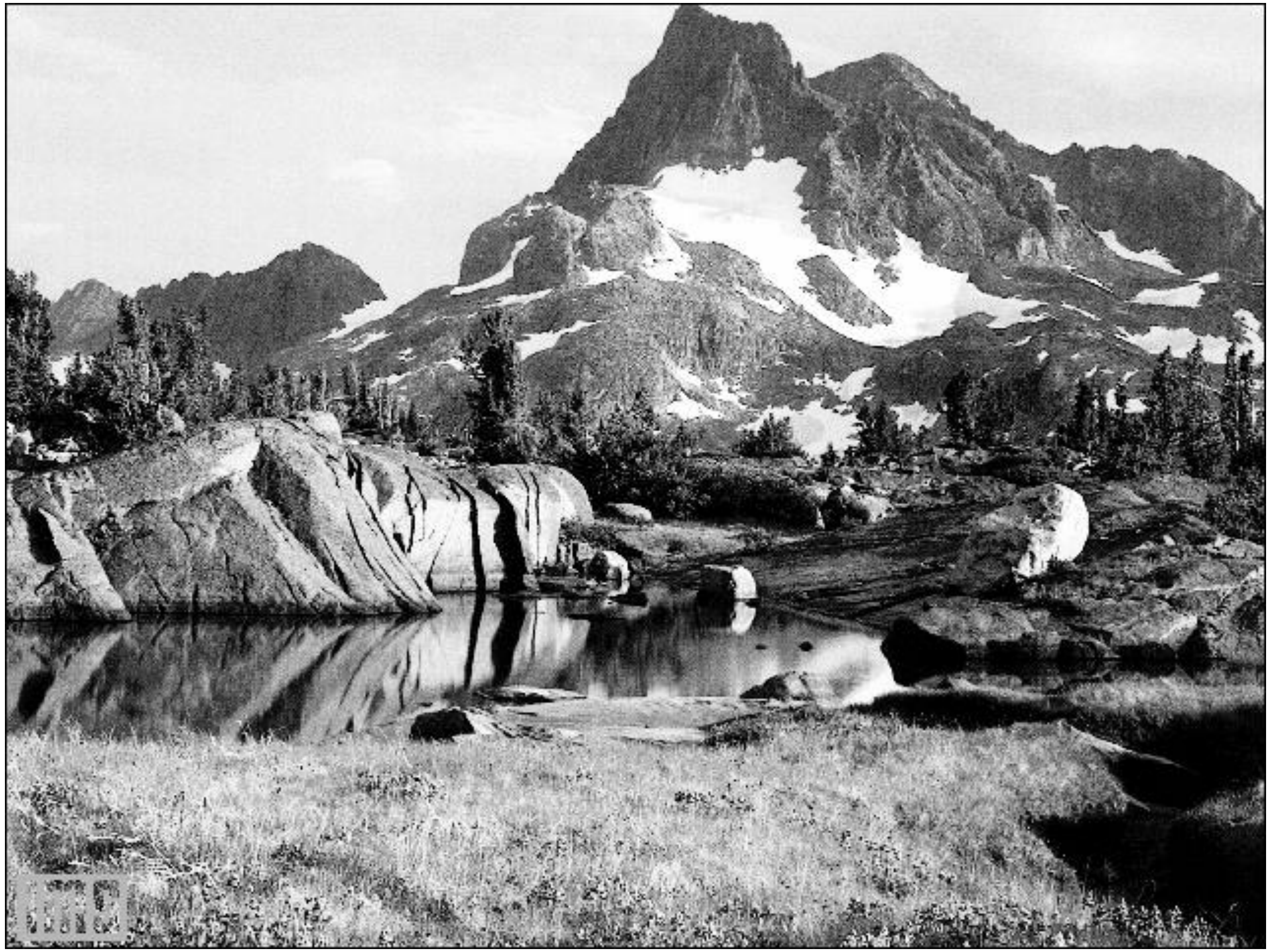
$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Usage - smoothing

- Box filter (averaging)

$$g(i,j) = \frac{1}{MN} \sum_{k=-a}^{a} \sum_{l=-b}^{b} I(i+k, j+l)$$

with M = 2a + 1
and N = 2b + 1



Usage - smoothing

- Box filter (averaging)

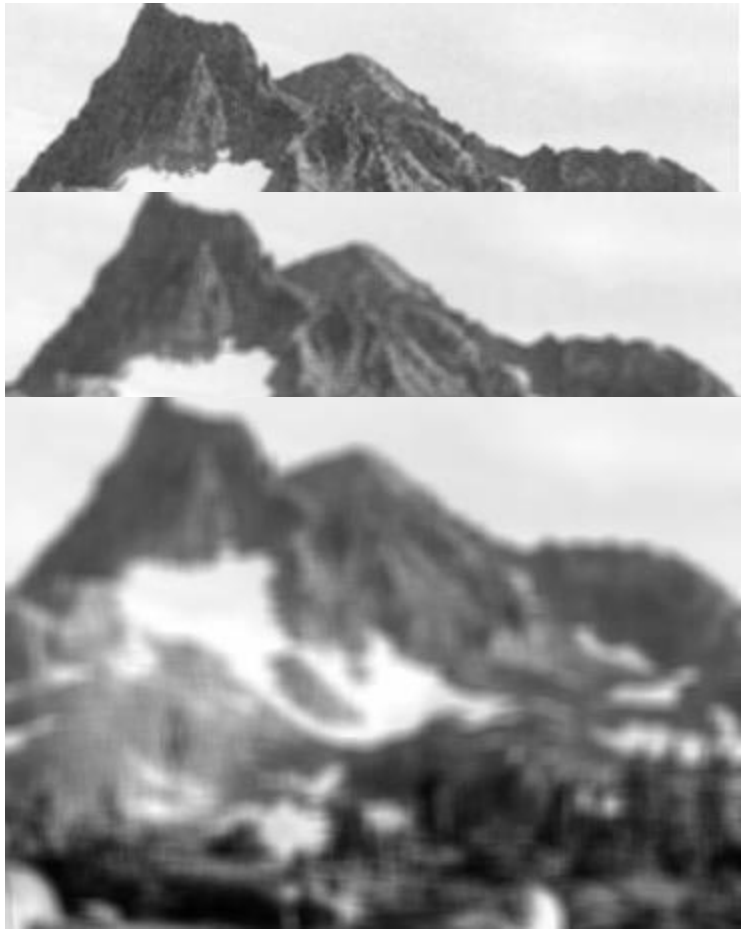
with M = N = 3

with M = N = 5

$$g(i,j) = \frac{1}{MN}\sum_{k=-a}^{a}\sum_{l=-b}^{b} I(i+k, j+l)$$ *with M = 2a + 1 and N = 2b + 1*

Usage - smoothing

- Box filter (averaging)

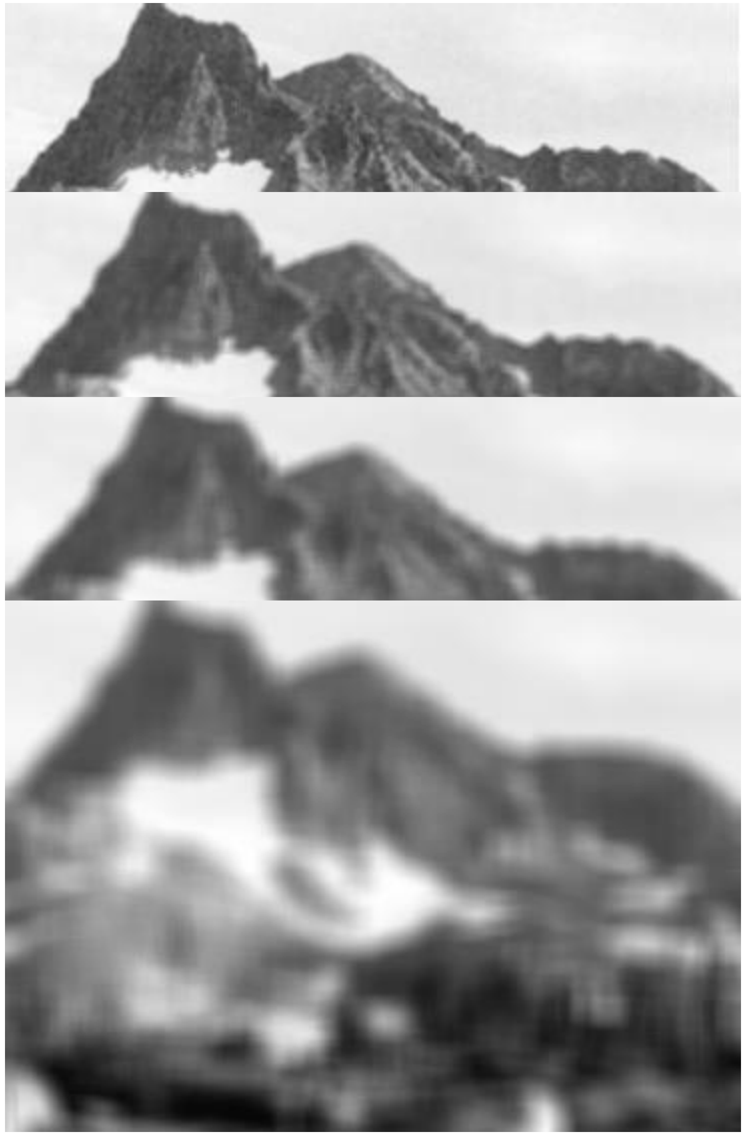


with M = N = 9

$$g(i,j) = \frac{1}{MN}\sum_{k=-a}^{a}\sum_{l=-b}^{b} I(i+k, j+l)$$ *with M = 2a + 1 and N = 2b + 1*

Usage - smoothing

- Box filter (averaging)

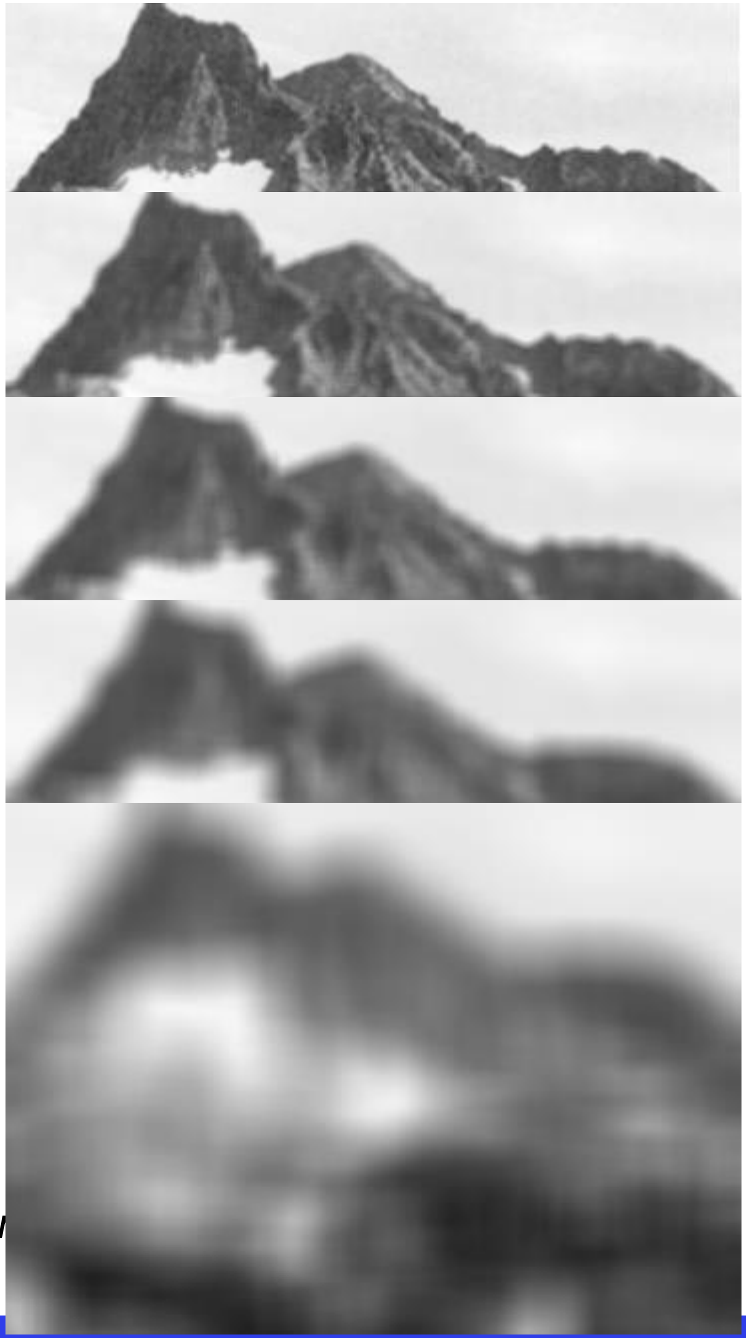


with M = N = 15

$$g(i,j) = \frac{1}{MN} \sum_{k=-a}^{a} \sum_{l=-b}^{b} I(i+k, j+l)$$ *with M = 2a + 1 and N = 2b + 1*

Usage - smoothing

- Box filter (averaging)



with M = N = 35

$$g(i,j) = \frac{1}{MN} \sum_{k=-a}^{a} \sum_{l=-b}^{b} I(i+k, j+l)$$

with M = 2a + 1 ar

Usage - smoothing

- Box filter
- Gaussian filter

$$g(i,j) = \frac{1}{\sum_{-a}^{a}\sum_{-b}^{b} e^{-\frac{k^2+l^2}{\sigma^2}}} \sum_{-a}^{a}\sum_{-b}^{b} I(i+k, j+l)e^{-\frac{k^2+l^2}{\sigma^2}}$$

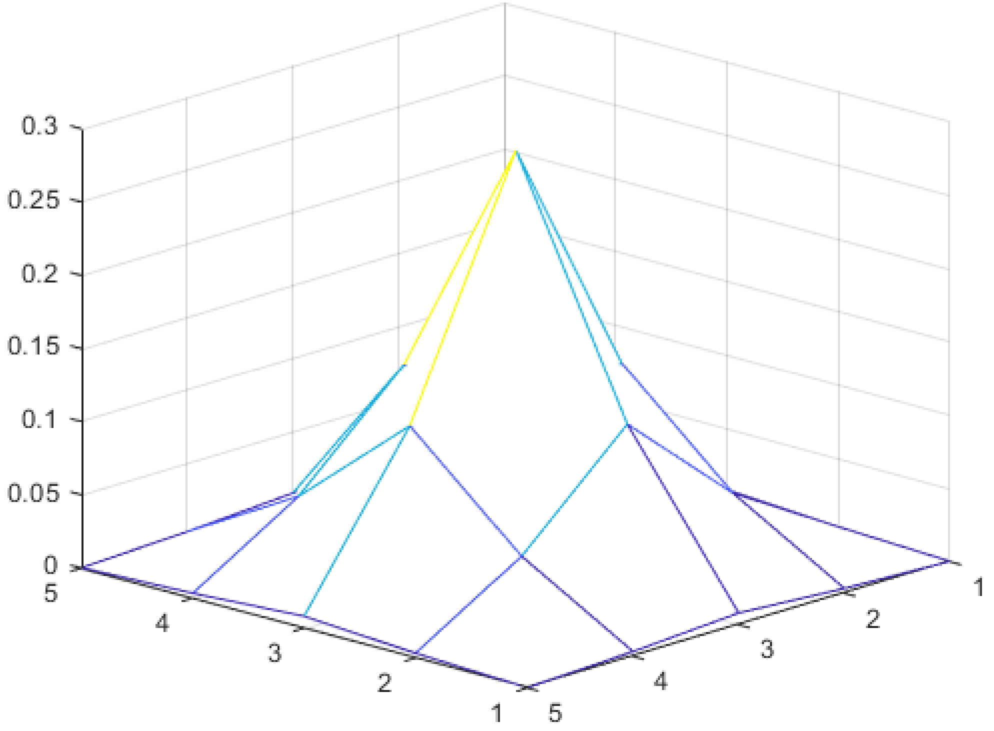
Usage - smoothing

- Box filter
- Gaussian filter

$$g(i,j) = \frac{1}{\sum_{-a}^{a}\sum_{-b}^{b} e^{-\frac{i^2+j^2}{\sigma^2}}} \sum_{-a}^{a}\sum_{-b}^{b} I(i+k, j+l) e^{-\frac{i^2+j^2}{\sigma^2}}$$

For σ = 0.75 and a filter size of 5x5

0.0002	0.0033	0.0081	0.0033	0.0002
0.0033	0.0479	0.1164	0.0479	0.0033
0.0081	0.1164	0.2831	0.1164	0.0081
0.0033	0.0479	0.1164	0.0479	0.0033
0.0002	0.0033	0.0081	0.0033	0.0002

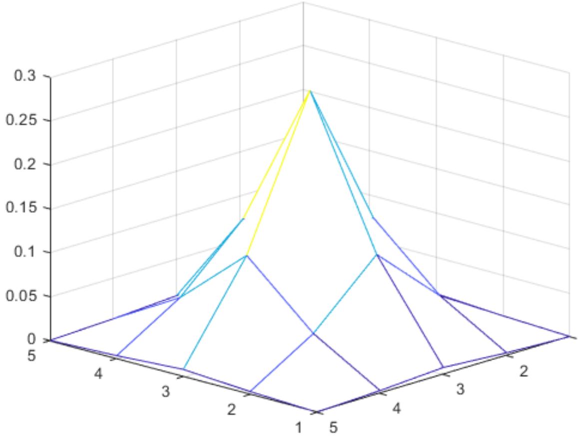


Usage - smoothing

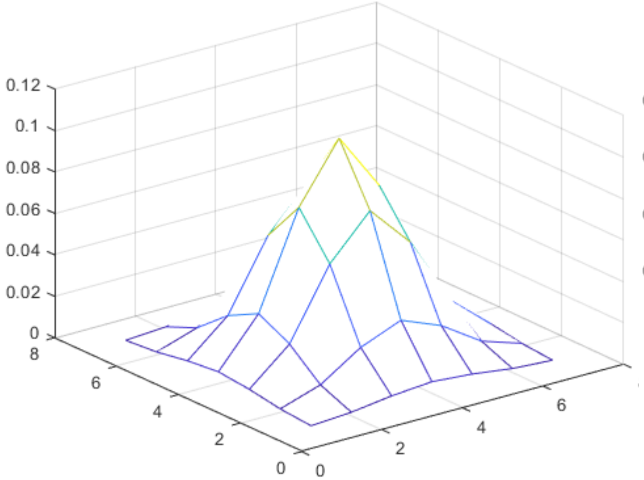
- Box filter
- Gaussian filter

$$g(i,j) = \frac{1}{\sum_{-a}^{a}\sum_{-b}^{b} e^{-\frac{i^2+j^2}{\sigma^2}}} \sum_{-a}^{a}\sum_{-b}^{b} I(i+k,j+l)\, e^{-\frac{i^2+j^2}{\sigma^2}}$$

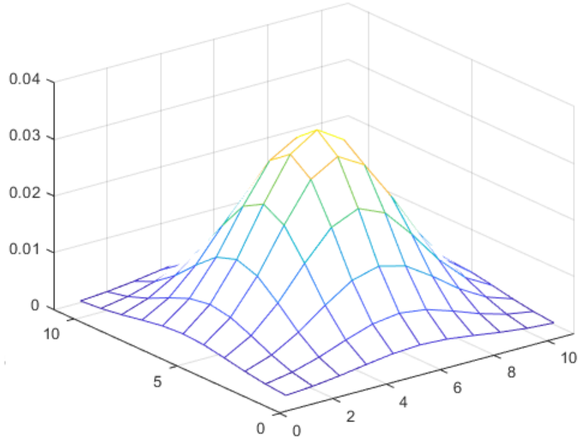
σ = 0.75,
filter size of 5x5



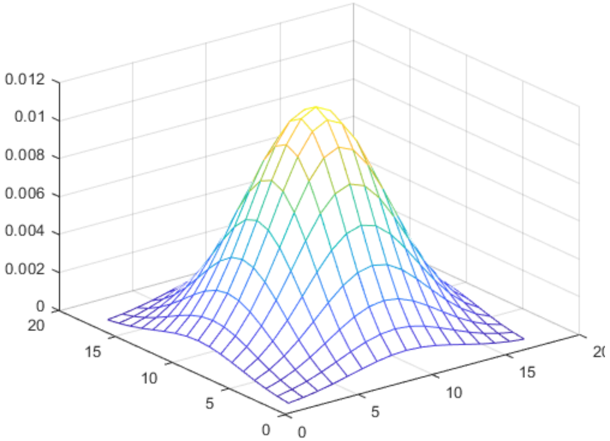
σ = 1.25,
filter size of 7x7



σ = 2.25,
filter size of 11x11

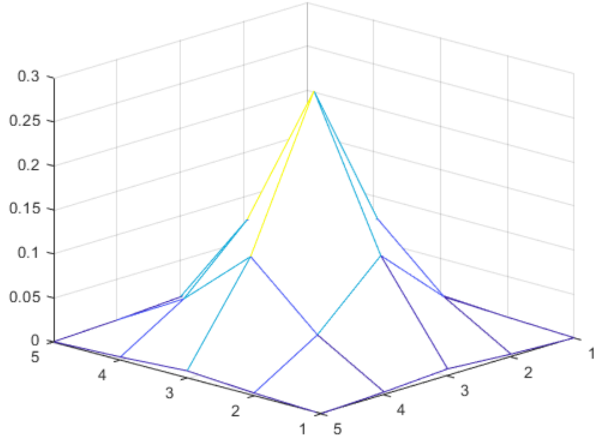


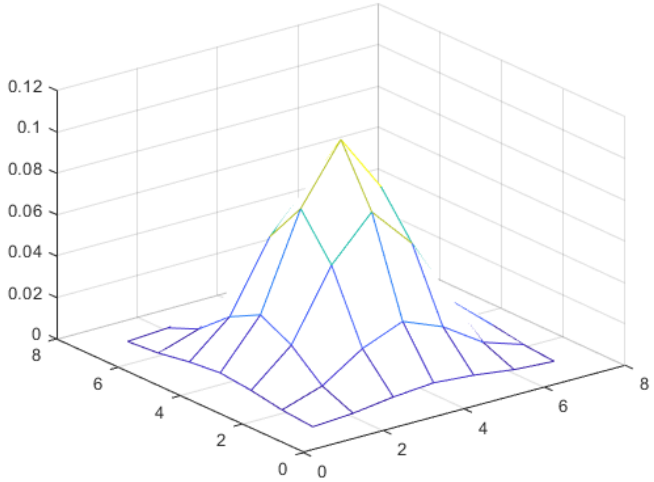
σ = 3.75,
filter size of 17x17



Usage - smoothing

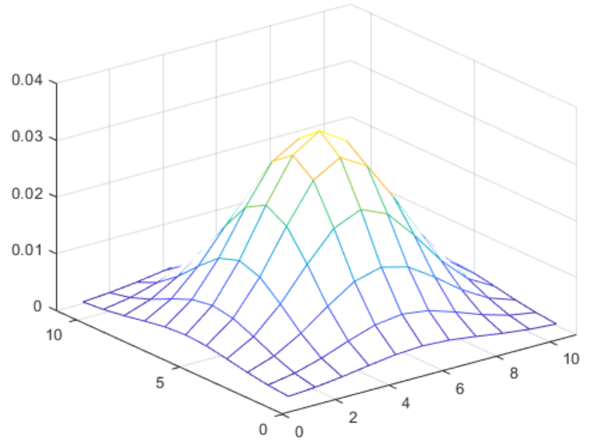
- Box filter
- Gaussian filter

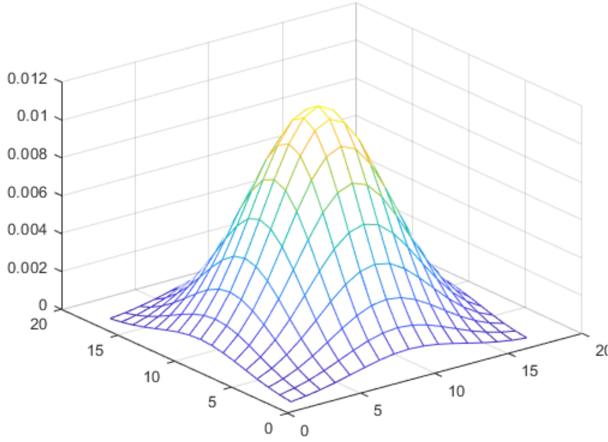




Usage - smoothing

- Box filter
- Gaussian filter





Usage - smoothing

- Box filter
- Gaussian filter

Usage - sharpening

- Second derivative - Laplacian

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

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

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

Usage - sharpening

- Second derivative - Laplacian

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

0	1	0
1	-4	1
0	1	0

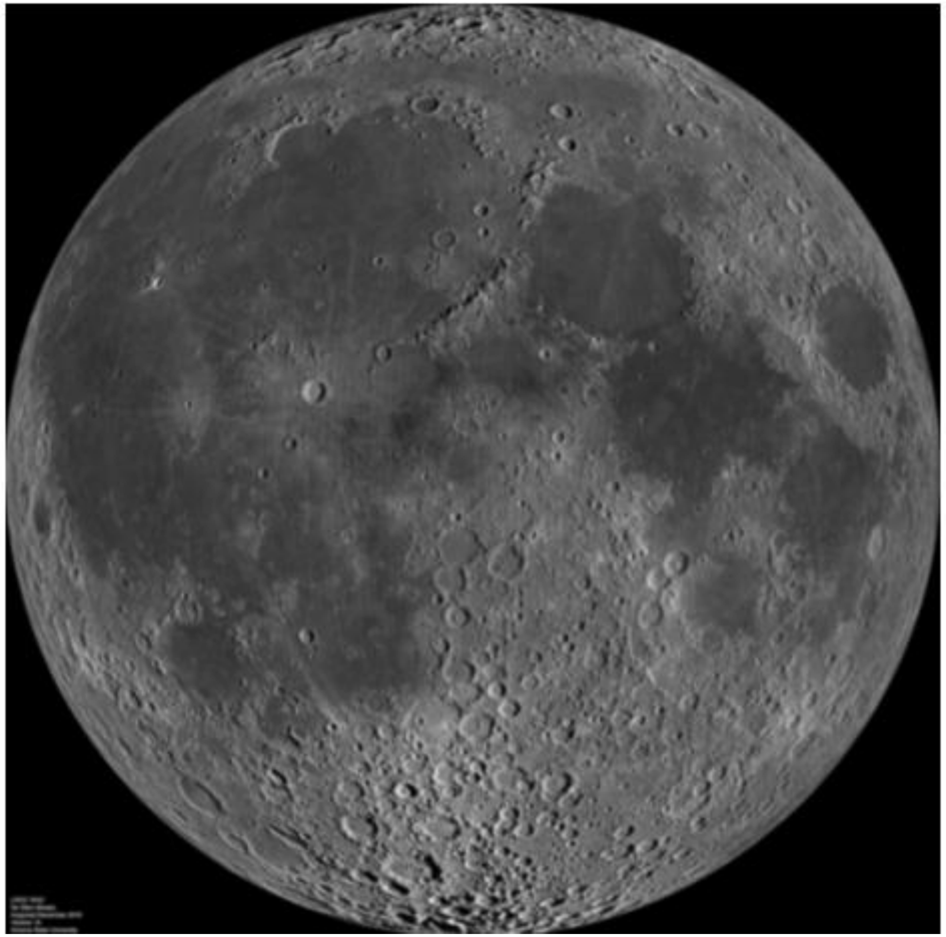
1	1	1
1	-8	1
1	1	1

Anisotropic

Usage - sharpening

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

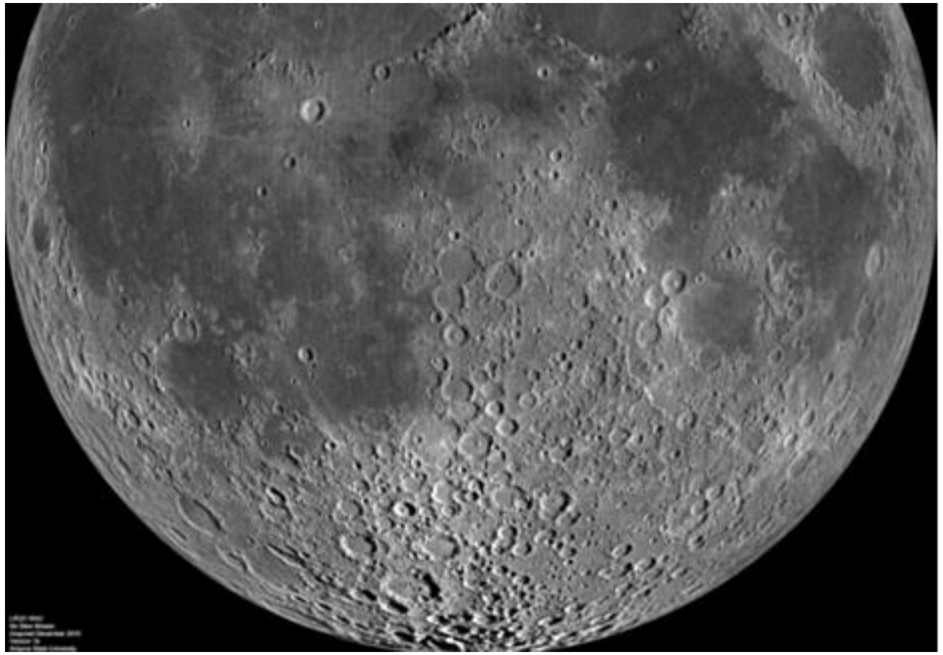
0	1	0
1	-4	1
0	1	0



Usage - sharpening

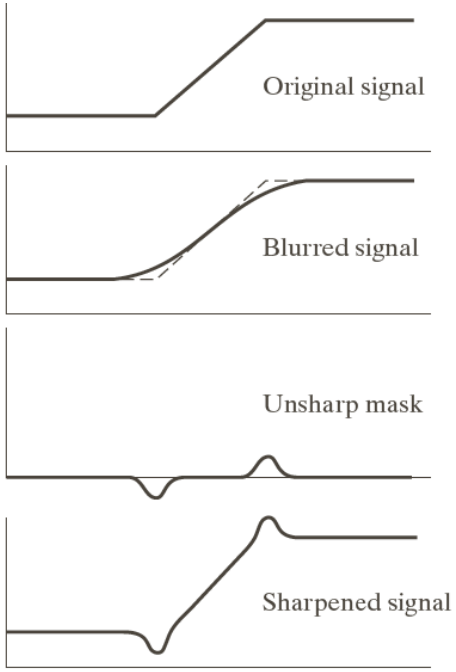
- Second derivative - Laplacian

$$g(x, y) = f(x, y) + \nabla^2 f(x, y)$$



Usage - sharpening

- Second derivative - Laplacian
- Unsharp masking & highboost



Source: Digital Image Processing, Gonzales & Woods

$$g_{mask}(x,y) = f(x,y) - B_f(x,y)$$

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

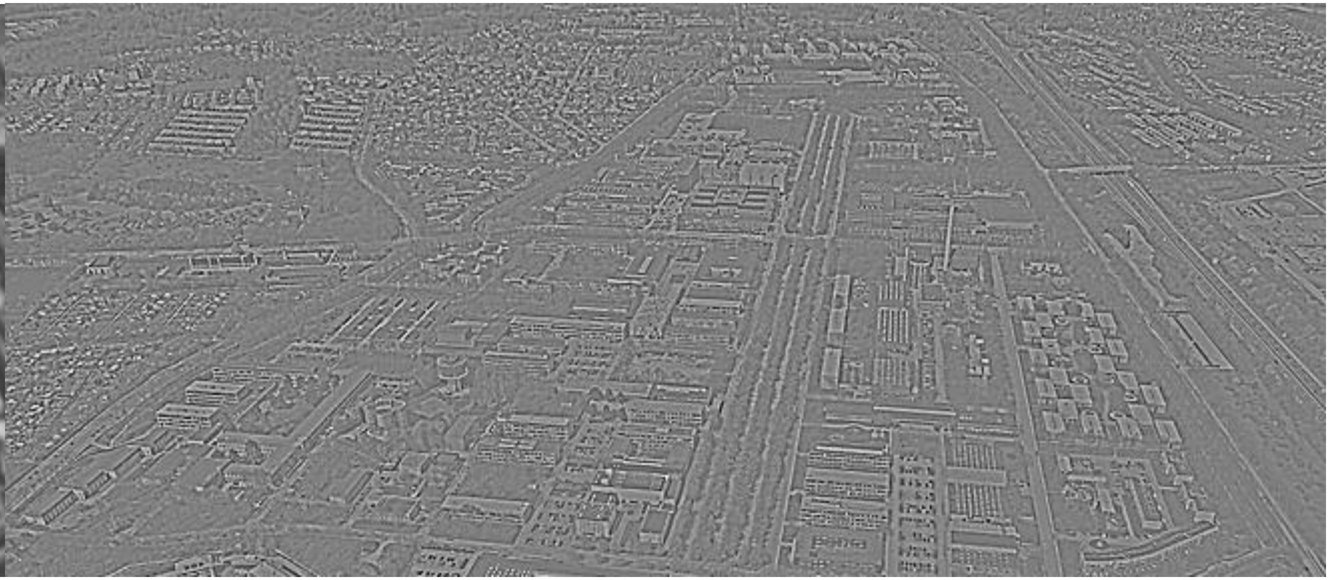
Unsharp masking for k=1
Highboost for k>1

Usage - sharpening

- Second derivative - Laplacian
- Unsharp masking & highboost

Gaussian blur (sigma = 1.25)

Mask



Usage - sharpening

- Second derivative - Laplacian
- Unsharp masking & highboost

End of spatial filtering part

-

Pause ?

Outline - Frequency domain filtering

- Principle
- Smoothing/low-pass filtering
- High-pass filtering
- Band-pass/band-reject filters

Principle

- Discrete Fourier Transform

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

$$with\ u \in [\![0, M-1]\!]\ and\ v \in [\![0, N-1]\!]$$

f(x,y) input image,

u and v spatial frequencies

- Inverse transform

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

$$with\ x \in [\![0, M-1]\!]\ and\ y \in [\![0, N-1]\!]$$

Principle

Complex result of DFT

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

With Spectrum

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

And phase angle

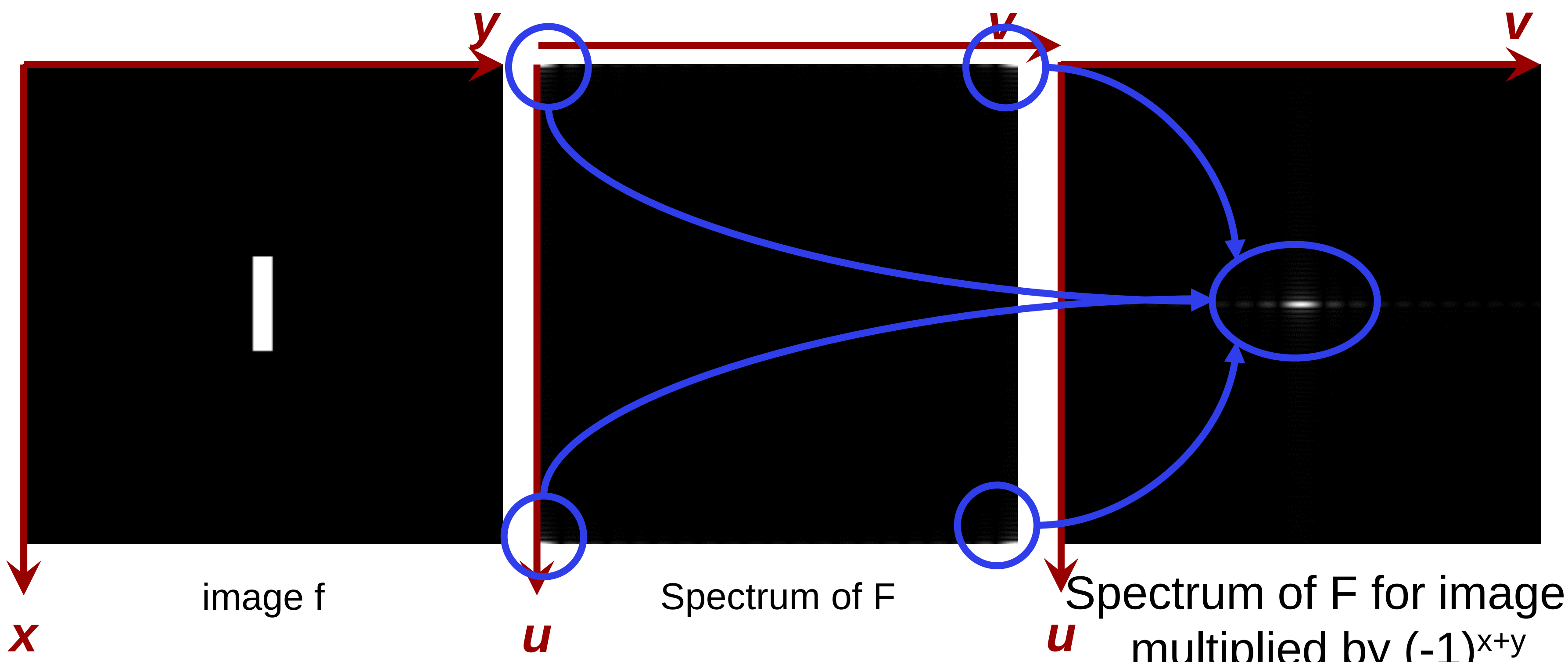
$$\varphi(u,v) = tan^{-1}\left(\frac{R(u,v)}{I(u,v)}\right)$$

Principle

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

$$with\ u_0 = \frac{M}{2} and\ v_0 = \frac{N}{2} :\ f(x,y)(-1)^{(x+y)} \Leftrightarrow F(u-M/2, v-N/2)$$

- Discrete Fourier Transform



Spectrum of F for image f multiplied by $(-1)^{x+y}$

→ Equivalent to shifting the quadrants of F

Principle

- Discrete Fourier Transform

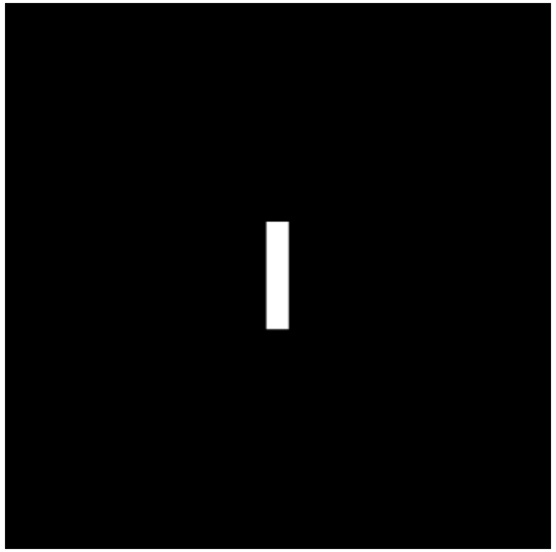
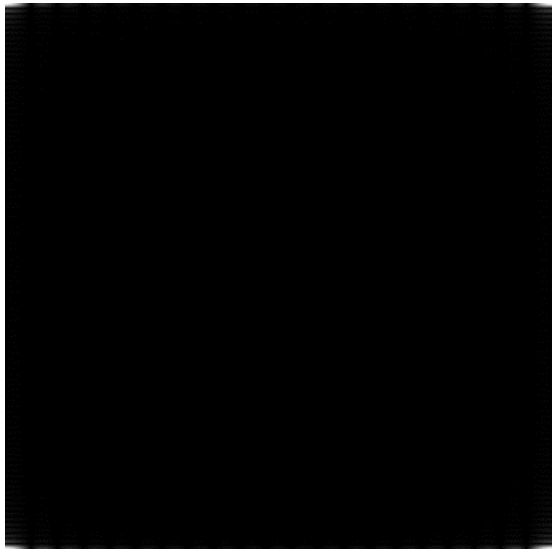
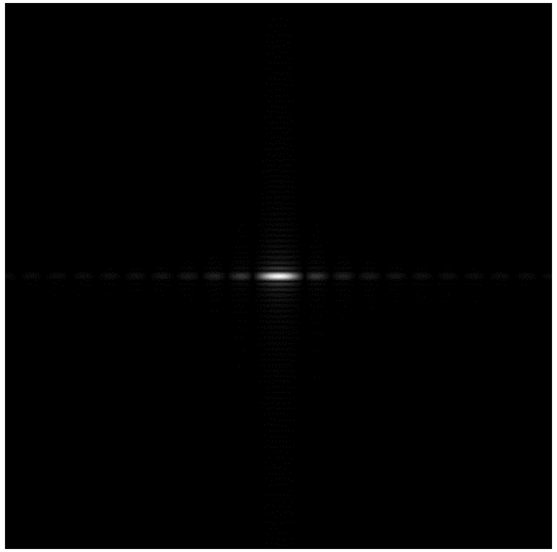


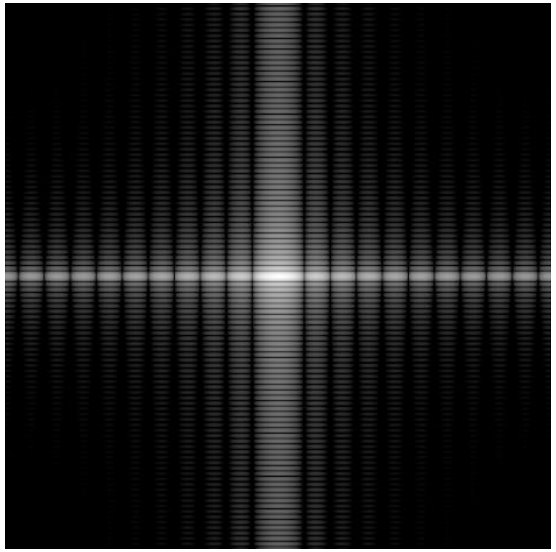
image f



Spectrum of F



Shifted spectrum of F

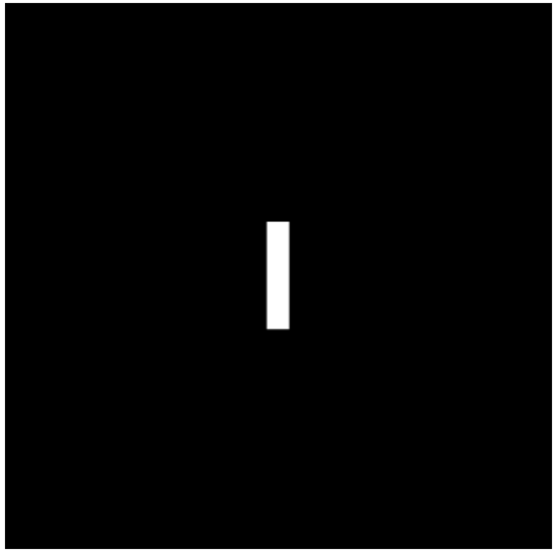


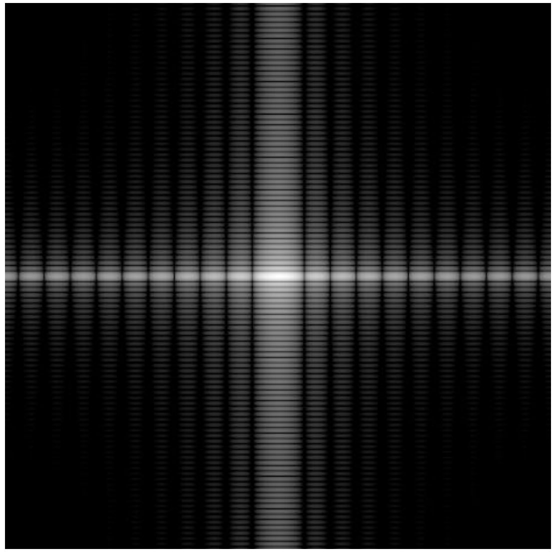
Log of shifted spectrum

$$rect(a,b) \Leftrightarrow ab \frac{\sin \pi ua}{\pi ua} \frac{\sin \pi vb}{\pi vb} e^{-j\pi(ua+vb)}$$

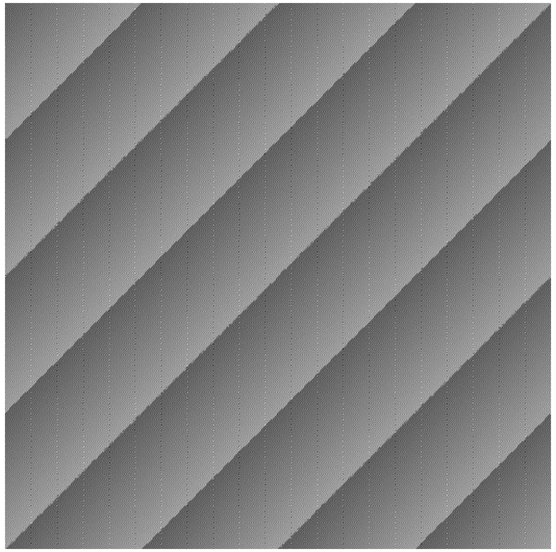
Principle

- Discrete Fourier Transform





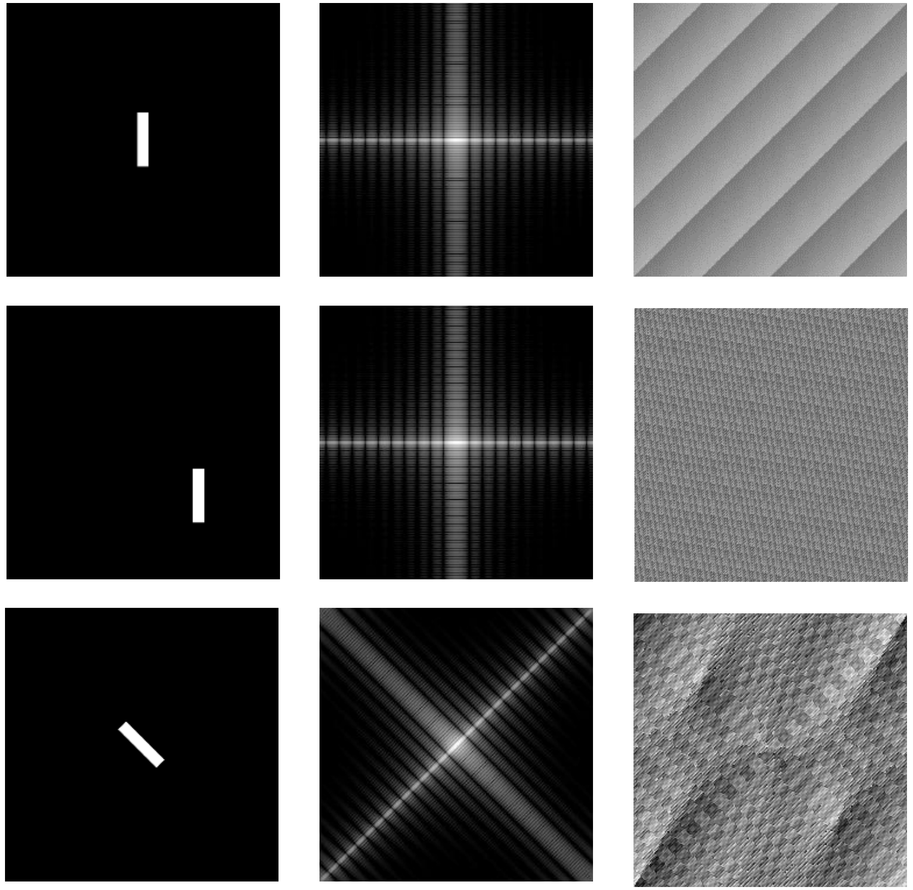
log of centered spectrum



Phase

$$rect(a,b) \Leftrightarrow ab\frac{\sin \pi ua}{\pi ua}\frac{\sin \pi vb}{\pi vb}e^{-j\pi(ua+vb)}$$

Principle



spatial translation

spatial rotation

image f

log of shifted spectrum

phase

$$rect(a,b) \Leftrightarrow ab\frac{\sin \pi ua}{\pi ua}\frac{\sin \pi vb}{\pi vb}e^{-j\pi(ua+vb)}$$

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

$$f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$$

$$with\ x = r\cos\theta\,, y = r\sin\theta$$

$$and\ u = \omega\cos\varphi\,, v = \omega\sin\varphi$$

Principle

- DFT
- Frequency resolution and padding

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

$$with\ u \in [\![0, M-1]\!]\ and\ v \in [\![0, N-1]\!]$$

The distance between the frequency samples of the DFT are function of the image resolution and spatial sampling

$$\Delta u = \frac{1}{M\Delta X}, \Delta v = \frac{1}{N\Delta Y}$$

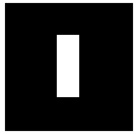
$$with\ M\ the\ number\ of\ rows, N\ number\ of\ columns$$
$$and\ \Delta X, \Delta Y\ the\ intervals\ between\ spatial\ samples$$

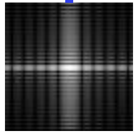
Zero padding adds no extra information, only a smoother version of the spectrum

Principle

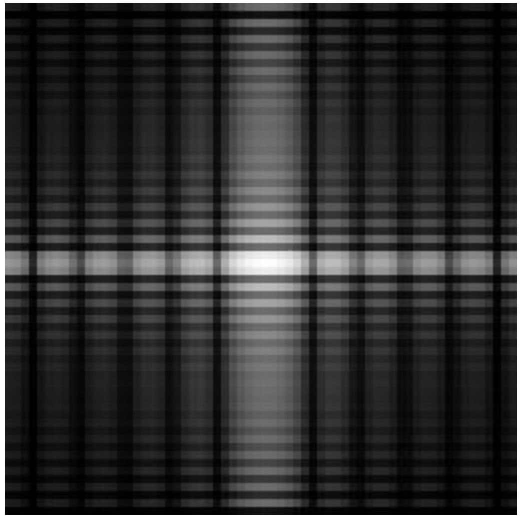
- DFT
- Frequency resolution and padding

64 x 64 image

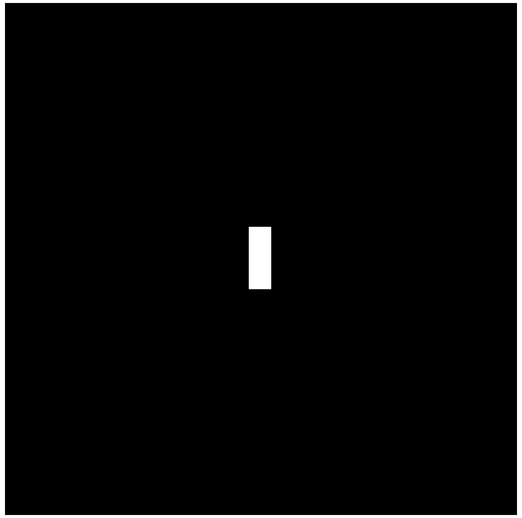




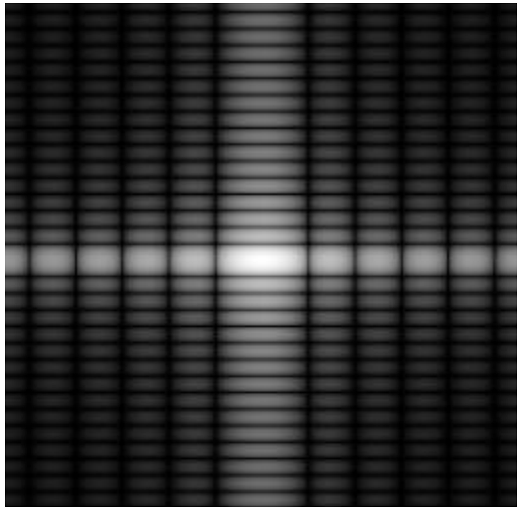
Upscaling
(no interpolation)



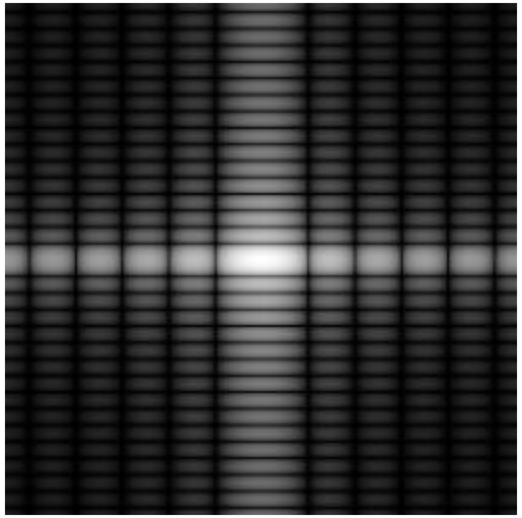
256 x 256 image
Same as
64 x 64 version
with 0 padding



Image



Log of shifted spectrum of DFT



Log of shifted spectrum of DFT

Principle

- DFT
- Convolution theorem

$$f(x,y) * h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m, y-n)$$

$$f(x,y) * h(x,y) \overset{\text{DFT}}{\Leftrightarrow} F(u,v)H(u,v)$$

$$f(x,y)h(x,y) \overset{\text{DFT}}{\Leftrightarrow} F(u,v) * H(u,v)$$

Principle

- DFT
- Convolution theorem
- Link with spatial filtering

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

Principle

- DFT
- Convolution theorem
- Link with spatial filtering

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

$$g(x,y) = \mathcal{F}^{-1}[H(u,v)R(u,v) + jH(u,v)I(u,v)]$$

$$\varphi(u,v) = tan^{-1}\left(\frac{R(u,v)}{I(u,v)}\right)$$

→ Zero-phase-shift filter

Principle

- DFT
- Convolution theorem
- Link with spatial filtering

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

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

$$if\ f(x,y) = \delta(x,y), then\ F(u,v) = 1 \ \Rightarrow\ g(x,y) = \mathcal{F}^{-1}[H(u,v)] = h(x,y)$$

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

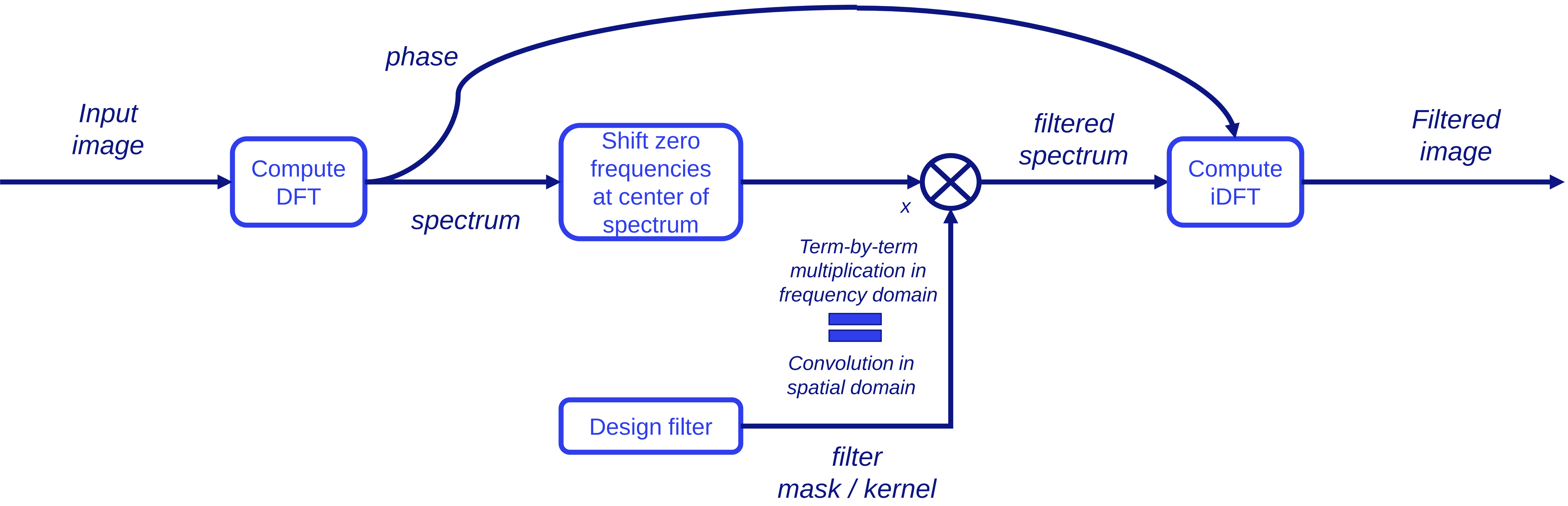
$$if\ F(u,v) = \delta(x,y), then\ f(x,y) = 1 \Rightarrow F(u,v) * H(u,v) = H(u,v) = \mathcal{F}[h(x,y)]$$

Principle - summary

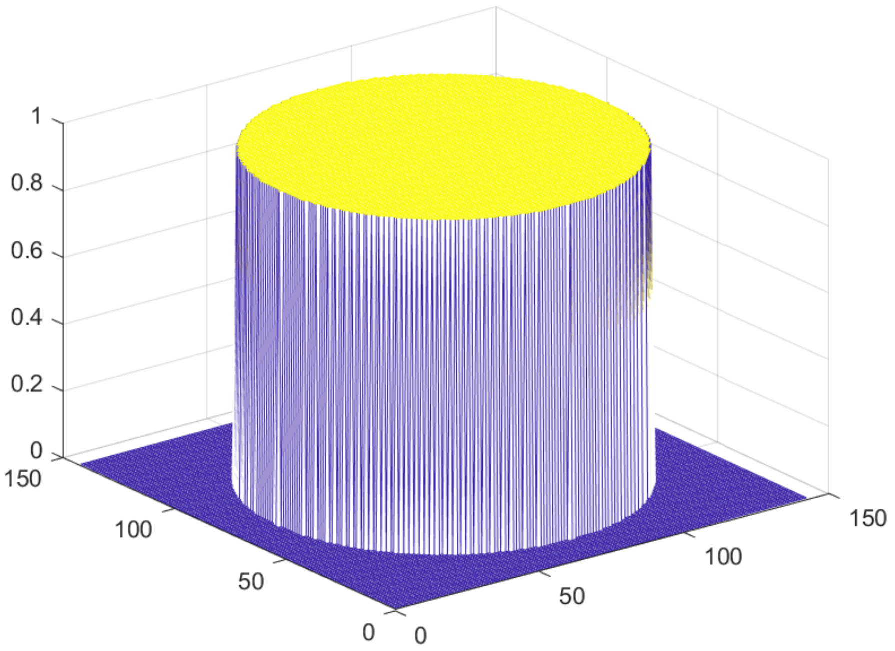
- DFT is a discrete transform used to perform Fourrier analysis. It transforms image content from spatial domain f(x,y) to frequency domain F(u,v), separated into spectrum (magnitude) and phase
- The relation between the image content and its spectrum is straightforward, not the relation with its phase.
- The spectrum can be used for filtering because
  - a term-by-term multiplication in the frequency domain = convolution in spatial domain
  - filters are more intuitive to design in frequency domain

    Filters modifying only the spectrum are called zero-phase-shift filter

Process


phase
Input image
Compute DFT
spectrum
Shift zero frequencies at center of spectrum
x
Term-by-term multiplication in frequency domain
=
Convolution in spatial domain
filtered spectrum
Compute iDFT
Filtered image
Design filter
filter mask / kernel

Low-pass filtering - Ideal low-pass (”tube” filter)



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

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

$$D_0 \text{: } cutoff \; frequency$$

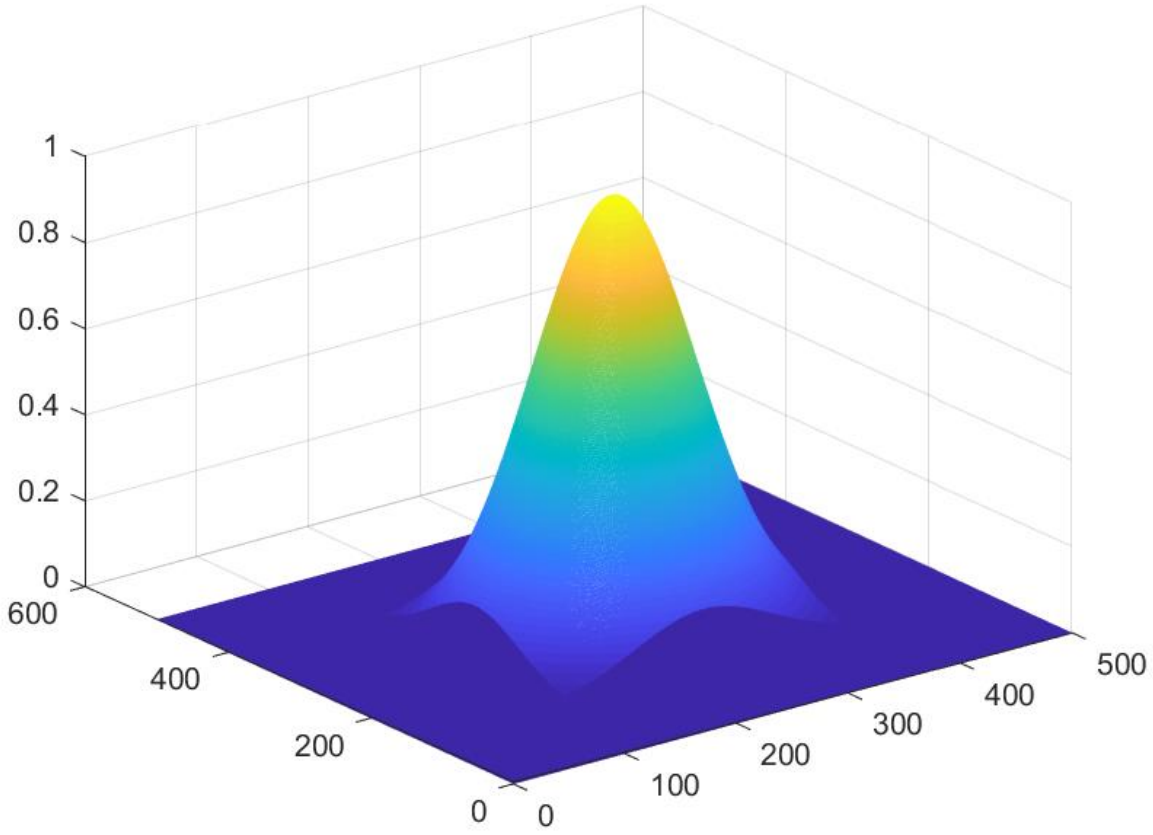
Low-pass filtering

- Ideal low-pass (”tube” filter)

image size
500x500

Cutoff (Radius): 199 161 121 57 28

Low-pass filtering – Gaussian filter



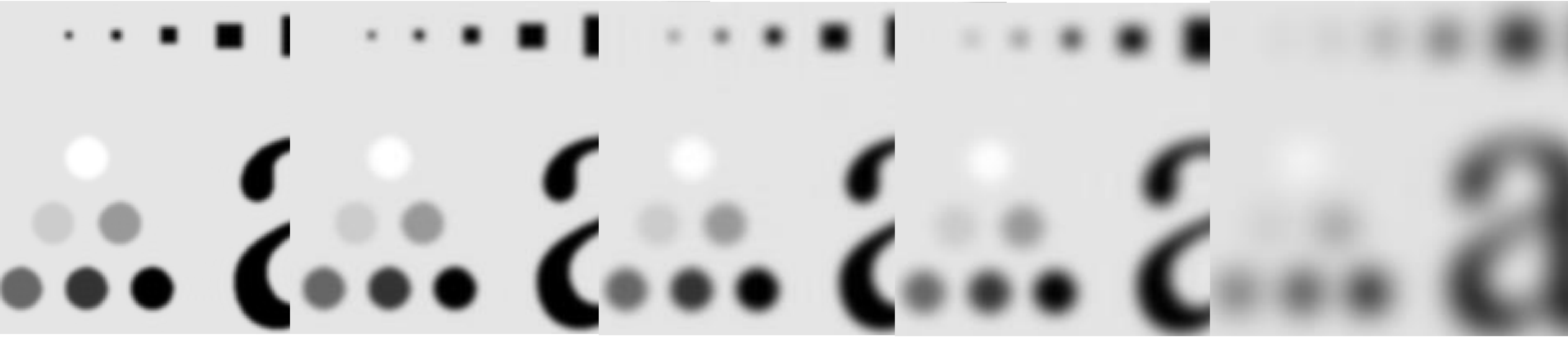
$$H(u,v) = e^{-\left(D(u,v)\right)^2/2\sigma^2}$$

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

$$\sigma: \ cutoff\ frequency$$

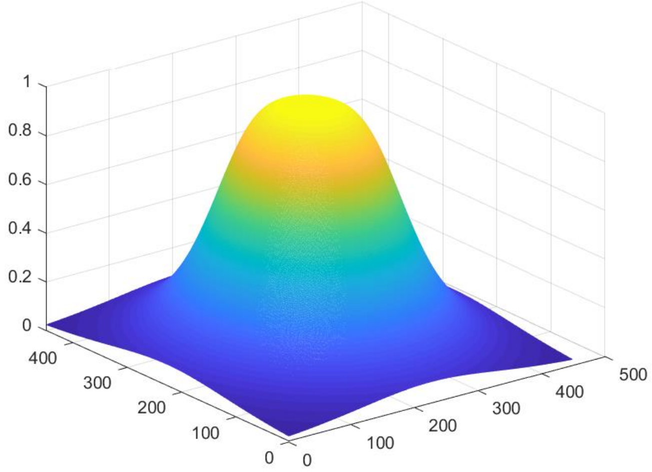
Low-pass filtering

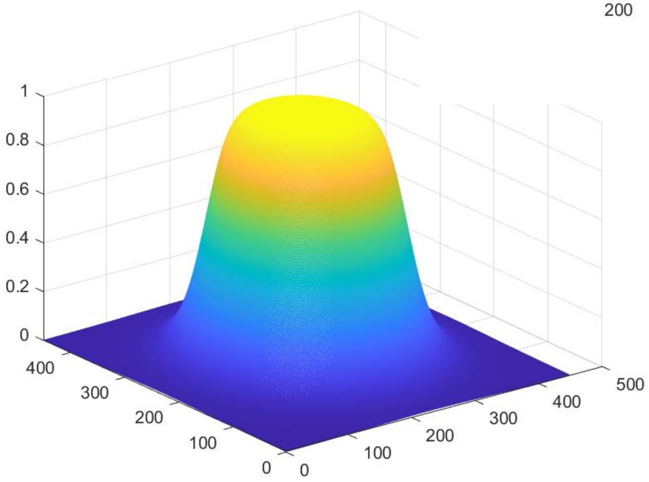
- Gaussian filter



Low-pass filtering – Butterworth filter

Order 2





Order 4

Filter with maximally flat frequency response

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

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

$D_0: \ cutoff \ frequency$

n: order of the filter (steepness of the slope)

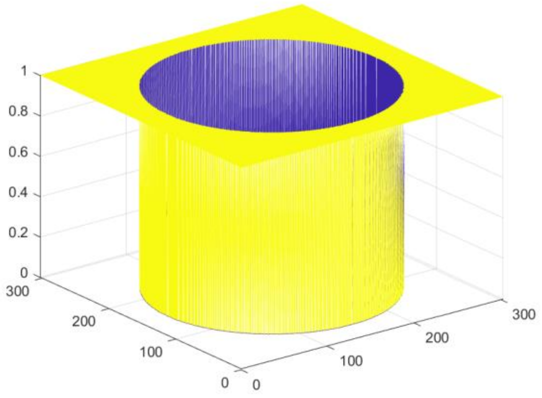
Low-pass filtering

- Butterworth filter

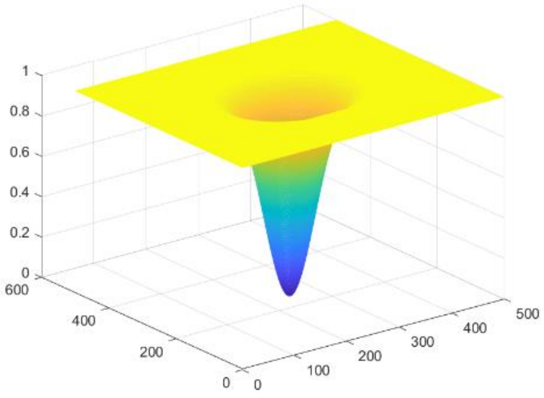
High-pass filtering

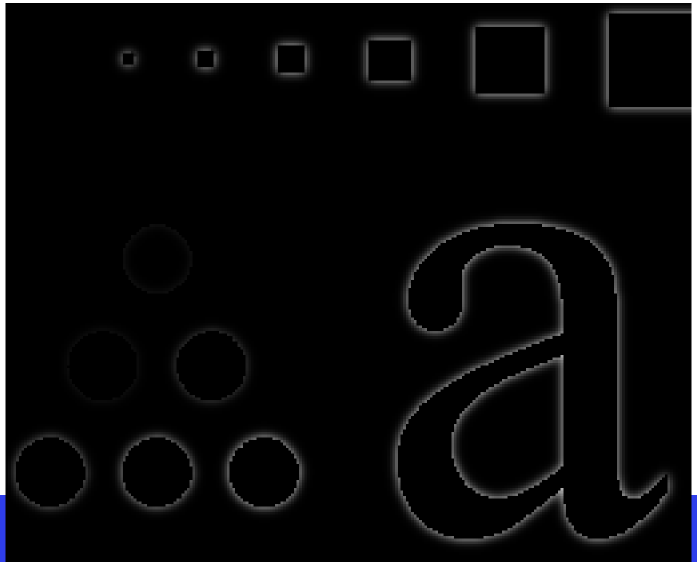
$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

ideal

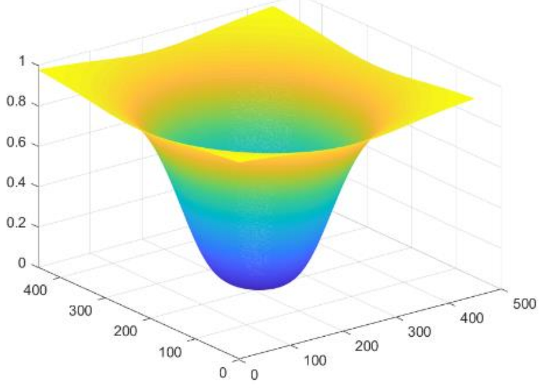


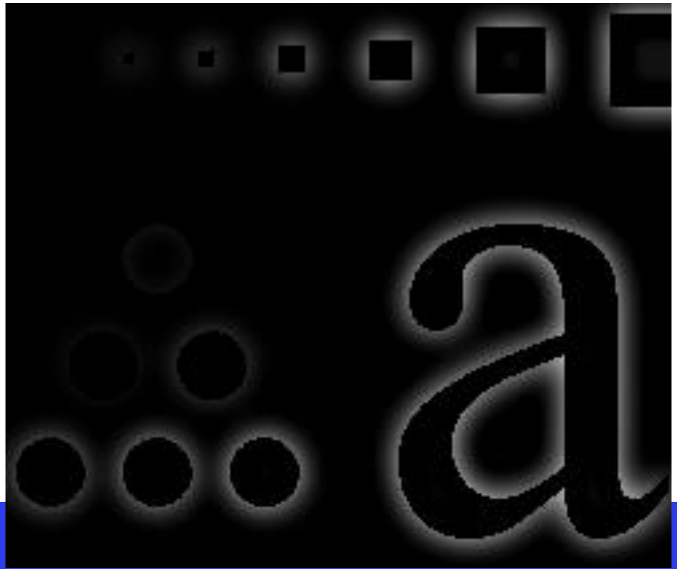
gaussian





Butterworth





Selective filters

- Band-reject

$$H(u,v) = \begin{cases} 0 \; if \, D_0 - W/2 \leq D(u,v) \leq D_0 + W/2 \\ 1 \; otherwise \end{cases}$$

Ideal

$$H(u,v) = e^{-\left(\frac{(D(u,v))^2 - D_0^2}{D(u,v)W}\right)^2 / 2\sigma^2}$$

Gaussian

With D_0 cutoff frequency

and W the width of the band

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

Butterworth based

- Band-pass

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

Selective filters

- Band-reject/Band-pass
- Notch filters

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

High pass Butterworth centered in (u_k, v_k)

$$H_k(u,v) = \frac{1}{1 + \left(D_{0k}/D_k(u,v)\right)^{2n}}$$

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

Coming up:

- Now: Exercise on image filtering
- This afternoon: Topic 3 Sparse and redundant recovery and representation

Thank you!