Image filtering

- Pixelwise mappings
  - Non-contextual - Global

$$f_{out}(x,y) = F(f_{in}(x,y))$$

- Filtering
  - Contextual - Local

$$f_{out}(x,y) = F(\{f_{in}(u,v) \mid (u,v) \in W(x,y)\})$$

Enhance low frequencies

Interesting: coarse structure

Enhance high frequencies

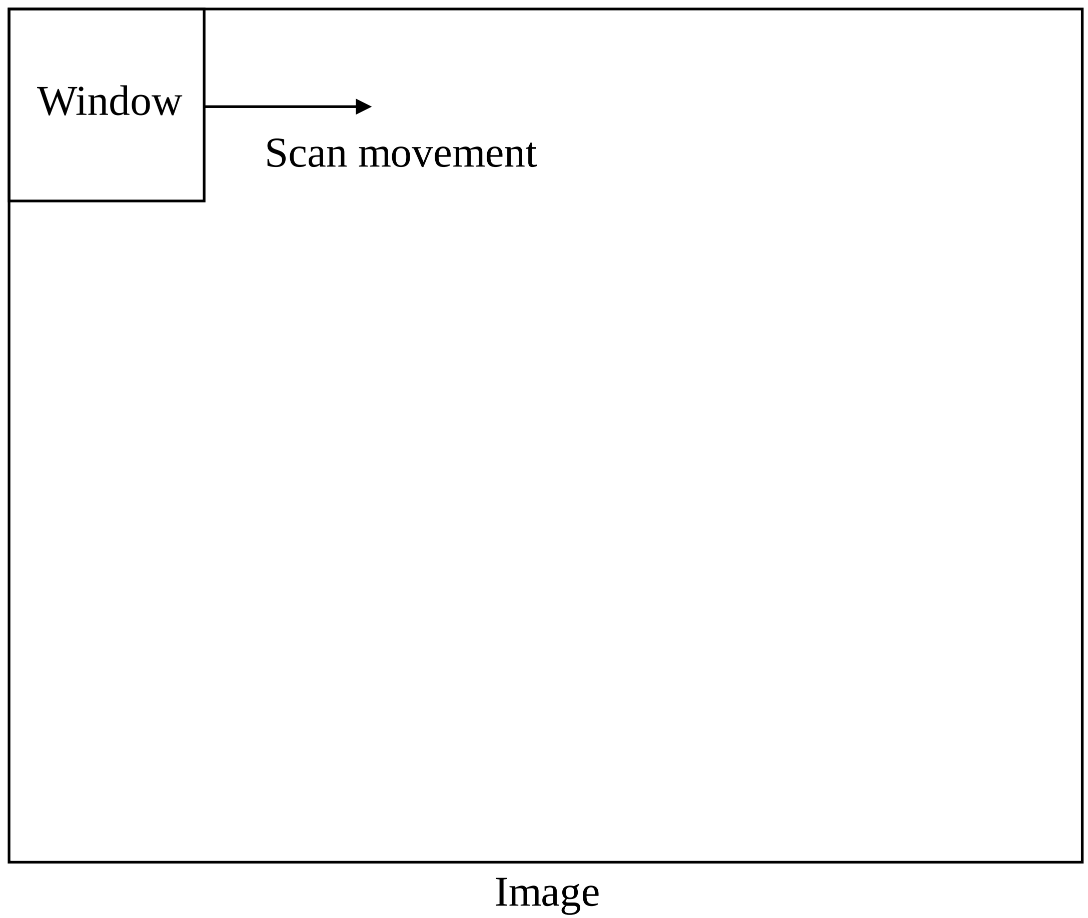
Interesting: fine structure

Image filtering

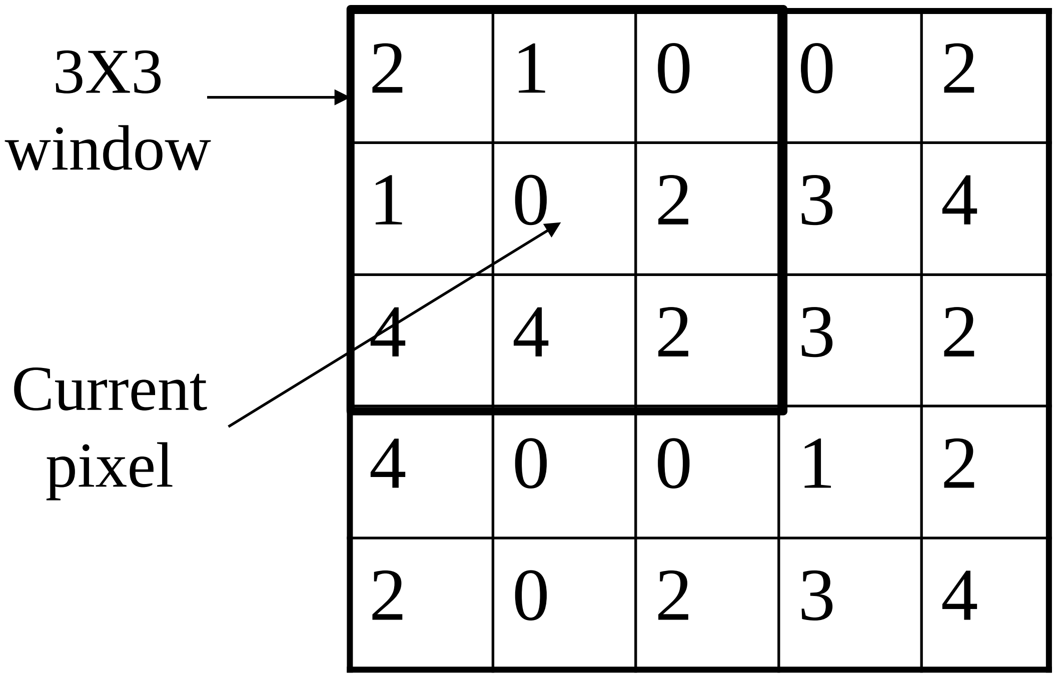
- Image processing – image in/image out
- Terminology from signal processing
  - Convolution
  - Analysis in frequency domain
- Moving window operations
- Lowpass/highpass/bandpass
- Preprocessing
- Image enhancement

DTU

Moving window operation



Filtering



New pixel value is computed from the 9 window values

Usually non-recursive update

Sum

2	1	0	0	2
1	0	2	3	4
4	4	2	3	2
4	0	0	1	2
2	0	2	3	4

16	15	18
17	15	19
18	15	19

2+1+0+1+0+2+4+4+2=16

Maximum

2	1	0	0	2
1	0	2	3	4
4	4	2	3	2
4	0	0	1	2
2	0	2	3	4

4	4	4
4	4	4
4	4	4

Max(2,1,0,1,0,2,4,4,2)=4

DTU

Boundary conditions

- Valid, results shrink for each operation
- Free boundary
- Fixed boundary
- Even reflection
- Odd reflection
- Periodic boundary (torus)
- Simulated boundary

Fixed boundary

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	2	1	0	0	2	0	0
0	0	1	0	2	3	4	0	0
0	0	4	4	2	3	2	0	0
0	0	4	0	0	1	2	0	0
0	0	2	0	2	3	4	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Even reflection

0	1	1	0	2	3	4	4	3
1	2	2	1	0	0	2	2	0
1	2	2	1	0	0	2	2	0
0	1	1	0	2	3	4	4	3
4	4	4	4	2	3	2	2	3
0	4	4	0	0	1	2	2	1
0	2	2	0	2	3	4	4	3
0	2	2	0	2	3	4		
0	4	4	0	0	1	2		

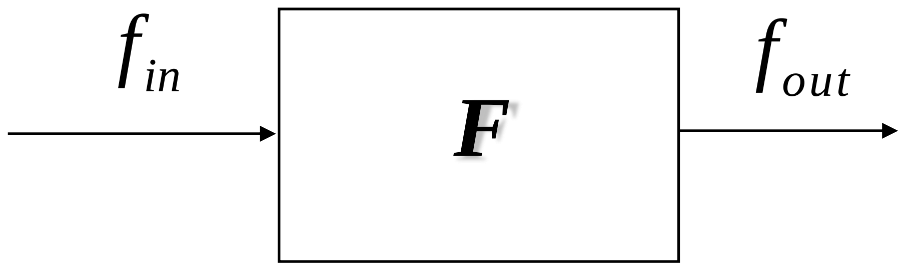
Odd reflection

2	4	4	4	2	3	2	3	2
2	0	1	0	2	3	4	3	2
0	1	2	1	0	0	2	0	0
2	0	1	0	2	3	4	3	2
2	4	4	4	2	3	2	3	2
0	0	4	0	0	1	2	1	0
2	0	2	0	2	3	4	3	2
0	0	4	0	0	1	2		
2	4	4	4	2	3	2		

Periodic boundary

1	2	4	0	0	1	2	4	0
3	4	2	0	2	3	4	2	0
0	2	2	1	0	0	2	2	1
3	4	1	0	2	3	4	1	0
3	2	4	4	2	3	2	4	4
1	2	4	0	0	1	2	4	0
3	4	2	0	2	3	4	2	0
0	2	2	1	0	0	2		
3	4	1	0	2	3	4		

Filter operators



- Linear operator

$$F(\alpha \cdot f + \beta \cdot g) = \alpha \cdot F(f) + \beta \cdot F(g)$$

$$f_{out}(u,v) = \int_x \int_y f_{in}(x,y) h(x,y,u,v) dxdy$$

- Shift-invariant

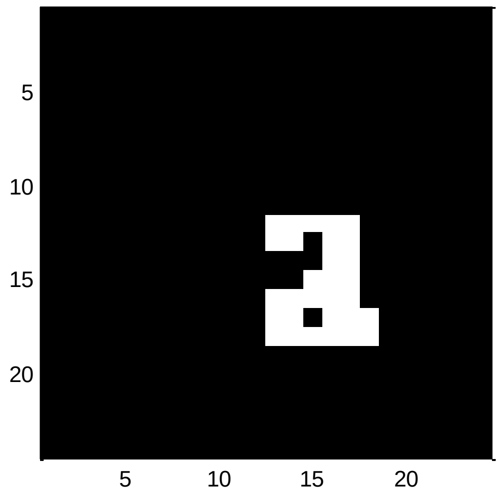
$$h(x,y,u,v) = h(x-u, y-v)$$

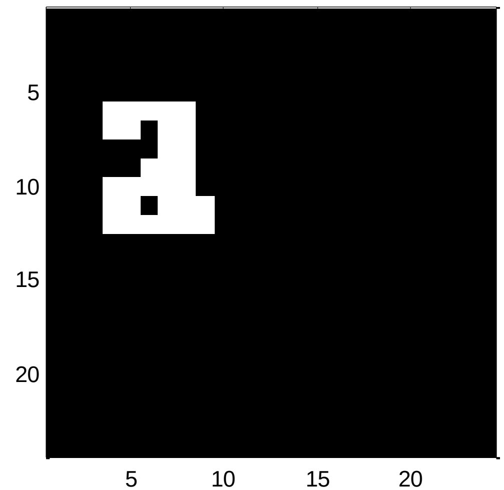
Similarity and convolution

- Compare two continuous images by computing

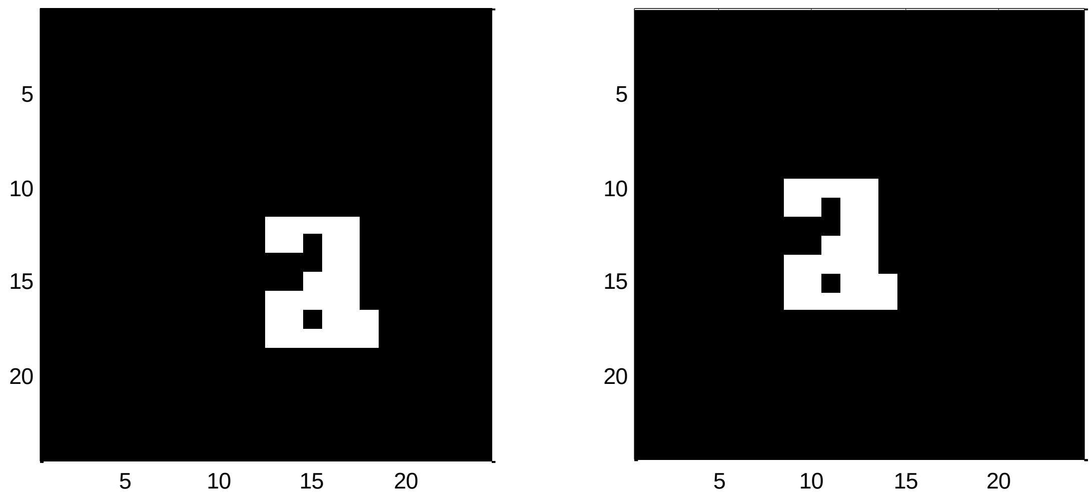
$$Sim_{f,h} = \int_x \int_y f(x,y)h(x,y)dxdy$$

Sum of product image=0

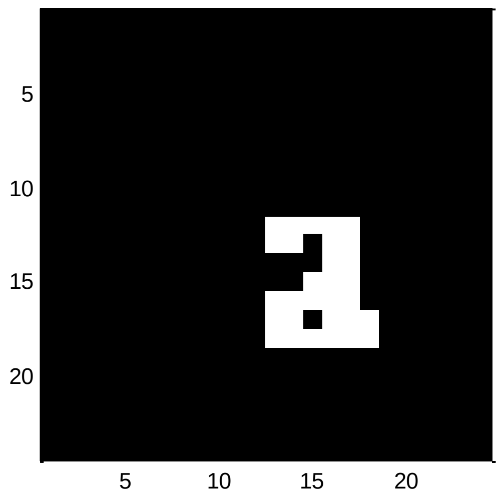


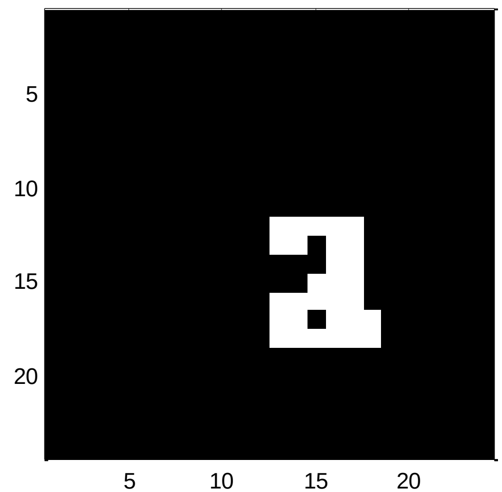


Sum of product image=4



Sum of product image=30





Similarity and convolution

- Compare two continuous images by computing

$$Sim_{f,h} = \int_x \int_y f(x, y)h(x, y)dxdy$$

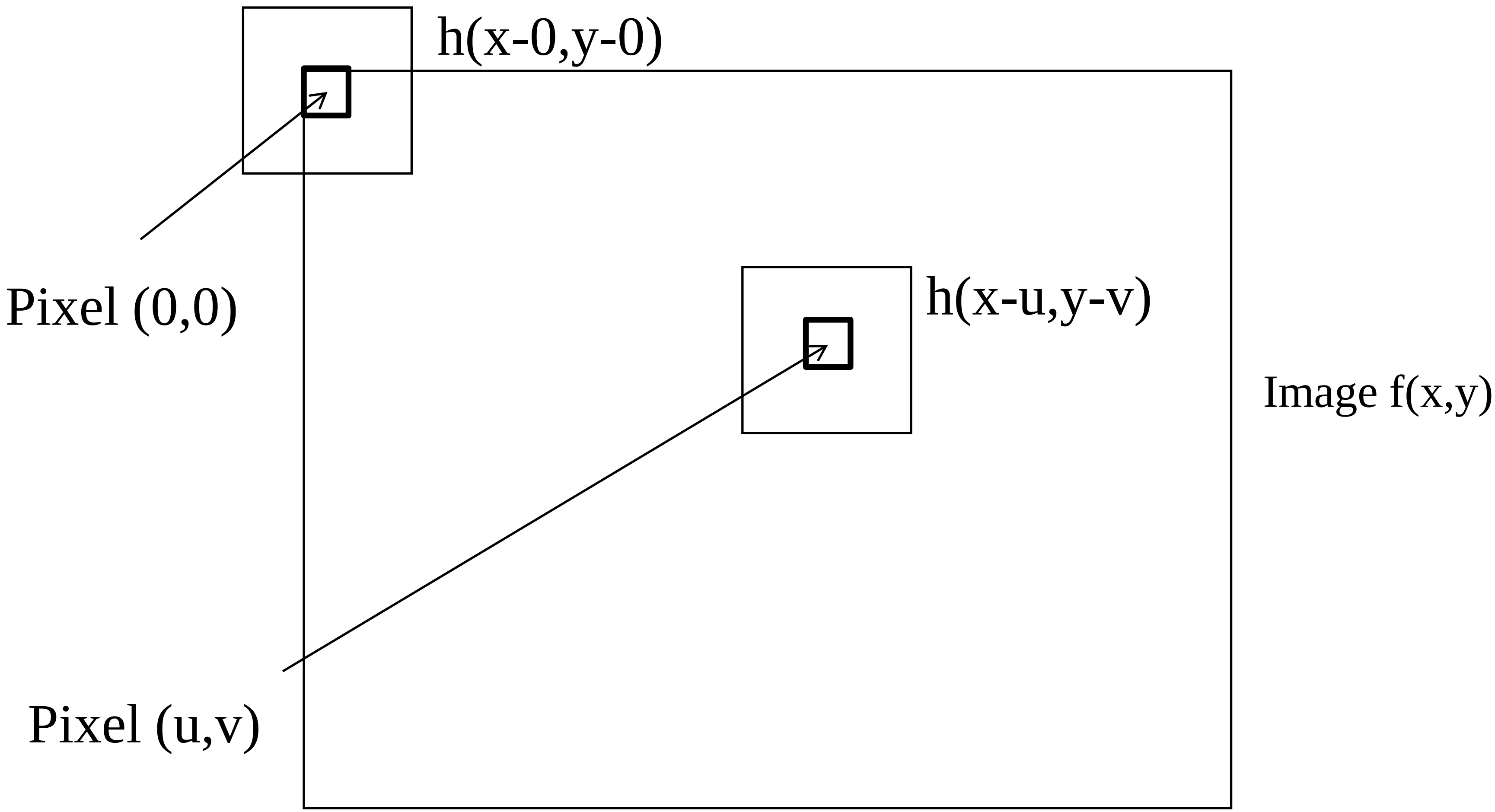
- Compute similarity for any displacement of h

$$Sim_{f,h}(u,v) = \int_x \int_y f(x, y)h(x-u, y-v)dxdy$$

- Compare with convolution

$$(f * h)(u,v) = \int_x \int_y f(x, y)h(u-x, v-y)dxdy$$

Moving comparison window



$$Sim_{f,h}(u,v) = \int_x \int_y f(x,y)h(x-u,y-v)dxdy$$

Linear shift-invariant filters

Linear shift-invariant filters can be expressed through convolution as

$$(f * h)(u,v) = \int_x \int_y f(x,y)\breve{h}(x-u, y-v)\,dx\,dy$$

In digital form

$$(f * h)(i,j) = \sum_k \sum_l f(k,l)\breve{h}(k-i, l-j)$$

DTU

LSI filter example

f

2	1	0	0	2
1	0	2	3	4
4	4	2	3	2
4	0	0	1	2
2	0	2	3	4

$\breve{h}$

-1	-1	-1
0	0	0
1	1	1

result

7	8	5
1	-4	-6
-6	-4	2

2*(-1)+1*(-1)+0*(-1)+1*0+0*0+

2*0+4*1+4*1+2*1=7

DTU

Point spread function *h*

f

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

$\breve{h}$

-1	-1	-1
0	0	0
1	1	1

result

1	1	1
0	0	0
-1	-1	-1

$= h$

DTU

Operators $\breve{h}$ and h

- h is called convolution mask, impulse response or point spread function
- $\breve{h}$ is called the kernel or filter mask
- $\breve{h}$ is separable if

$$\breve{h}(i, j) = h_1(i) \cdot h_2(j)$$

i.e.

$$(f * \breve{h})(i, j) = \sum_k h_1(k-i) \sum_l f(k,l) h_2(l-j)$$

Matlab filter2 (1)

>> help filter2

FILTER2 Two-dimensional digital filter.

Y = FILTER2(B,X) filters the data in X with the 2-D FIR filter in the matrix B. The result, Y, is computed using 2-D correlation and is the same size as X.

Matlab filter2 (2)

Y = FILTER2(B,X,'shape') returns Y computed via 2-D correlation with size specified by 'shape':

'same' - (default) returns the central part of the correlation that is the same size as X.

'valid' - returns only those parts of the that are computed without the zero-padded edges, size(Y) < size(X).

'full' - returns the full 2-D correlation, size(Y) > size(X).

Matlab filter2 (3)

FILTER2 uses CONV2 to do most of the work. 2-D correlation is related to 2-D convolution by a 180 degree rotation of the filter matrix.

See also FILTER, CONV2.

>>

Lowpass filtering

- Smoothing
- Removes high-frequency noise
- Blurs fine structure e.g. lines and edges

Mean filters

- Mean
- Weighted mean
- Trimmed mean
- Mean with adaptive neighborhood

Mean filter example

f

2	1	0	0	2
1	0	2	3	4
4	4	2	3	2
4	0	0	1	2
2	0	2	3	4

$\breve{h}$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

result

16/9	15/9	18/9
17/9	15/9	19/9
18/9	15/9	19/9

DTU

Trimmed mean filter example

f

2	1	0	0	2
1	0	2	3	4
4	4	2	3	2
4	0	0	1	2
2	0	2	3	4

result

8/5	8/5	11/5
9/5	8/5	11/5
10/5	8/5	11/5

44% trimmed mean

Rank filtering

Based on first sorting the pixels in the window

- Median
- Minimum/maximum
- Range
- Linear combination based on rank

Rank linear combination

6	8	5
2	3	3
0	5	0

Rank

0	0	2	3	3	5	5	6	8

Median coefficients

0	0	0	0	1	0	0	0	0

Range coefficients

-1	0	0	0	0	0	0	0	1

Trimmed mean coefficients

0	0	0.2	0.2	0.2	0.2	0.2	0	0

Exercise 00.4

2	1	3	4	5
1	1	0	2	3
2	0	**0**	1	2
5	1	2	3	1
4	3	1	2	0

What is the value of the marked pixel after a 5 x 5 median filter?

DTU

High-pass filtering

- Edge and line enhancement
- Both magnitude and direction
- Enhances high-frequency noise
- Optimality based on edge or line profile

Derivatives

- Gradient

$$\nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right)$$

- Hessian

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix}$$

Hessian parameters

- Laplacian (trace of Hessian)

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

- Determinant of Hessian (DoH)

$$DoH(f)(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} \cdot \frac{\partial^2 f(x,y)}{\partial y^2} - \left(\frac{\partial^2 f(x,y)}{\partial x \partial y} \right)^2$$

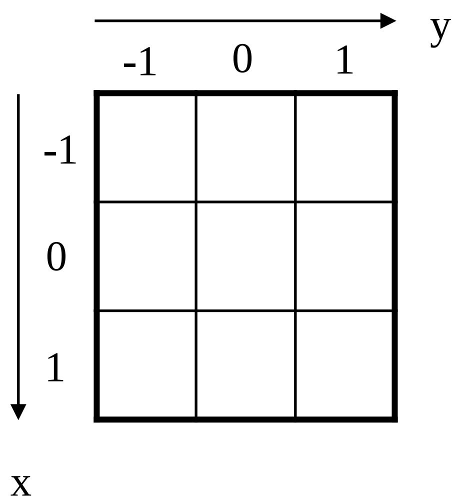
DTU

LSI filter design

Fit a 2D polynomial function

$$f\ (x,y) = p_1(x,y) + \varepsilon(x,y) = a \cdot x + b \cdot y + c + \varepsilon(x,y)$$

To the image function within the window



9 linear equations, 3 unknown

$$\begin{bmatrix} -1 & -1 & 1 \\ -1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \boldsymbol{\varepsilon} = \begin{bmatrix} f(-1,-1) \\ f(-1,0) \\ f(-1,1) \\ f(0,-1) \\ f(0,0) \\ f(0,1) \\ f(1,-1) \\ f(1,0) \\ f(1,1) \end{bmatrix} \Leftrightarrow \mathbf{x} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \boldsymbol{\varepsilon} = \mathbf{f}$$

DTU

Least squares solution

$$
\begin{bmatrix} a \\ b \\ c \end{bmatrix} = (\mathbf{x}^{\mathrm{T}}\mathbf{x})^{-1}\mathbf{x}^{\mathrm{T}}\mathbf{f} = \begin{bmatrix} -1/6 & -1/6 & -1/6 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 \\ -1/6 & 0 & 1/6 & -1/6 & 0 & 1/6 & -1/6 & 0 & 1/6 \\ 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 \end{bmatrix} \begin{bmatrix} f(-1,-1) \\ f(-1,0) \\ f(-1,1) \\ f(0,-1) \\ f(0,0) \\ f(0,1) \\ f(1,-1) \\ f(1,0) \\ f(1,1) \end{bmatrix}
$$

$\frac{1}{6}$

-1	-1	-1
0	0	0
1	1	1

a

$\frac{1}{6}$

-1	0	1
-1	0	1
-1	0	1

b

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

c

Estimating the gradient

$$\nabla f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right)$$

Estimating the gradient in the center pixel gives us

$$\nabla \hat{f}(0,0) = \left(\frac{\partial \hat{f}}{\partial x}(0,0), \frac{\partial \hat{f}}{\partial y}(0,0) \right) = \left(\frac{\partial p_1}{\partial x}(0,0), \frac{\partial p_1}{\partial y}(0,0) \right) = (a,b)$$

DTU

Fitting a plane with weights

Weights

¼	½	¼
½	1	½
¼	½	¼

$\frac{1}{8}$

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

a

$\frac{1}{8}$

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

b

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

c

Digital approximations to gradient

- Prewitt
- Sobel
- Roberts
- Robinson
- Kirsh

Prewitt filter

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- Direct computation of gradient vector
- Magnitude and orientation computed from vector

DTU

Sobel filter

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

- Direct computation of gradient vector
- Magnitude and orientation computed from vector

Exercise

2	1	3	4	5
1	1	0	2	3
2	0	**[0]**	1	2
5	1	2	3	1
4	3	1	2	0

What is the value of the marked pixel after a 3 x 3 Prewitt filter?

What is the value of the marked pixel after a 3 x 3 Sobel filter?

DTU

Roberts filter

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

- Direct computation of gradient vector
- Magnitude and orientation computed from vector
- Origin shift

DTU

Robinson filter

$$\begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix} \quad \ldots$$

- 8 filters in 8 gradient directions
- Magnitude is maximum response
- Orientation is corresponding direction

Kirsh filter

$$\begin{pmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{pmatrix} \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{pmatrix} \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix} \quad \ldots$$

- 8 filters in 8 gradient directions
- Magnitude is maximum response
- Orientation is corresponding direction

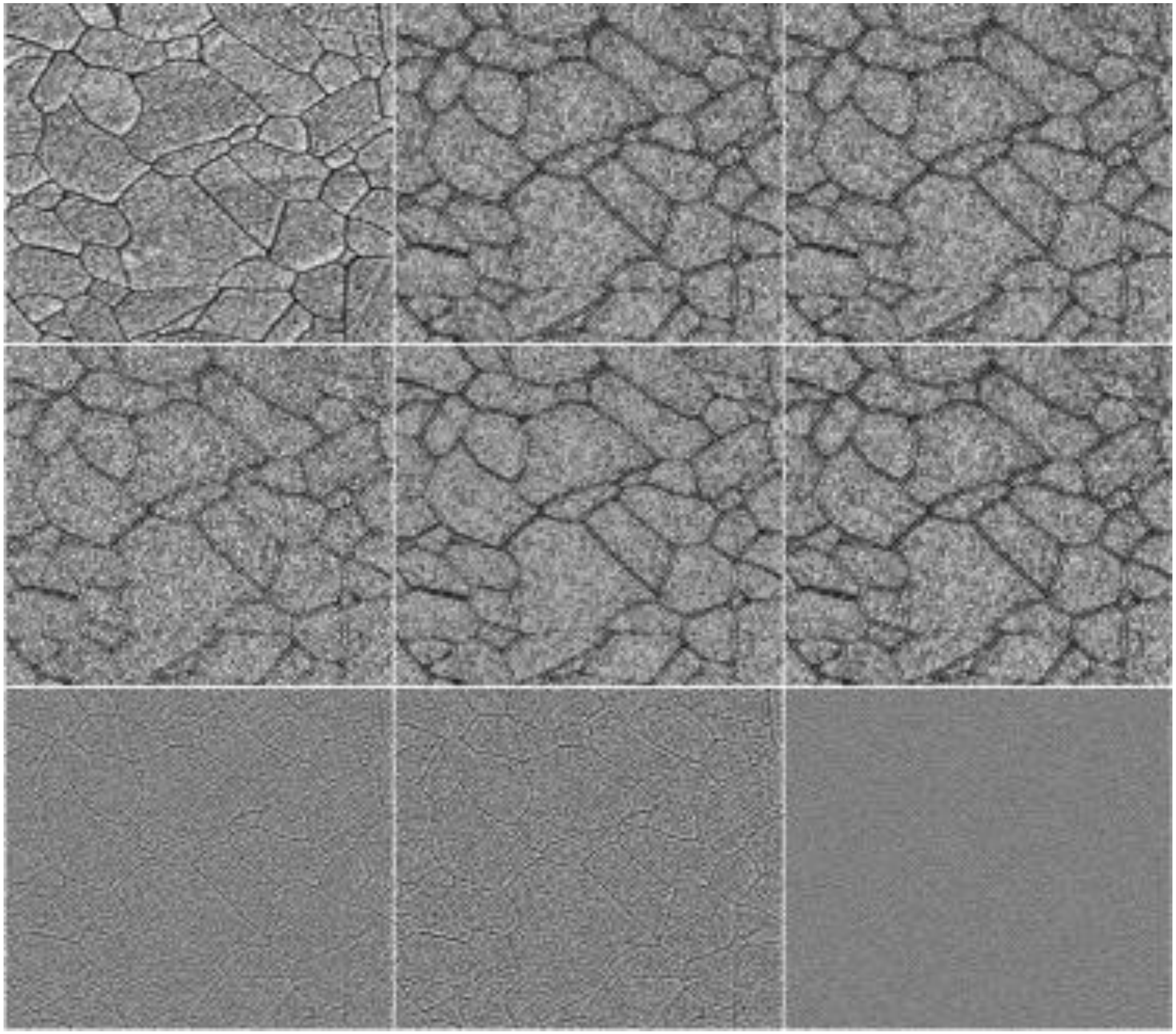
Digital approximations to (negative) laplacian

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad \frac{1}{3}\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad \frac{1}{3}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

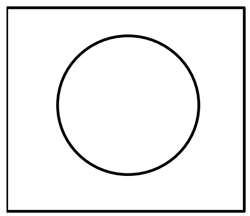
- No orientation
- Sensitive to noise
- Sharpness enhancement

DTU

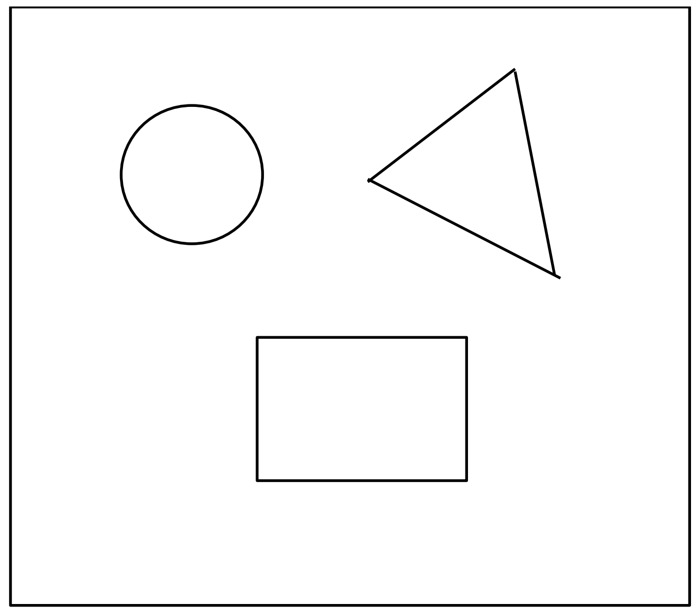
Edge filtering



Matched filters

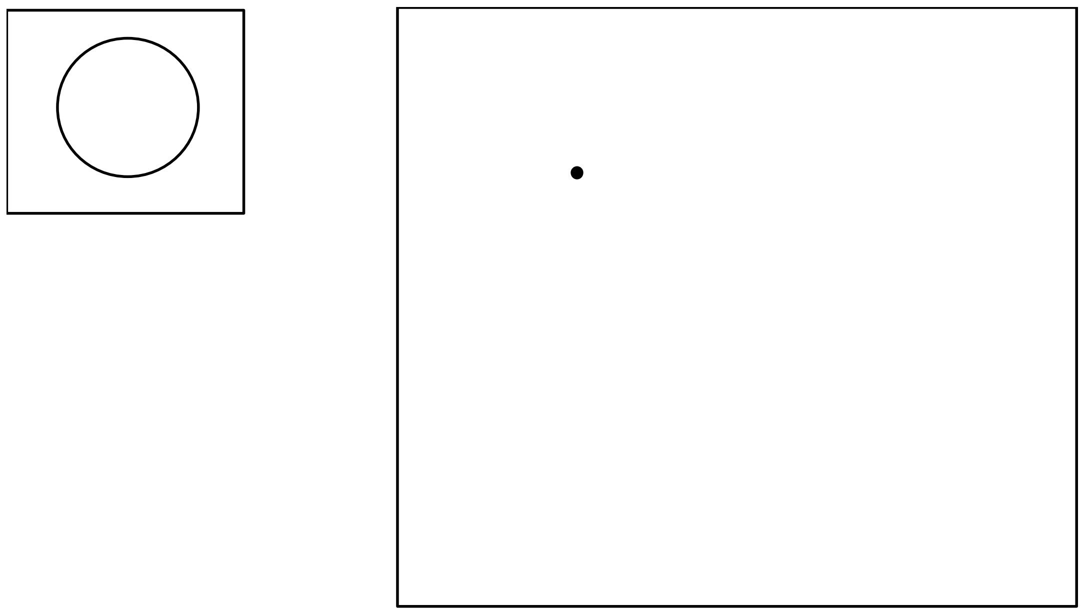


Filter



Find the circle

Matched filters



Circle found