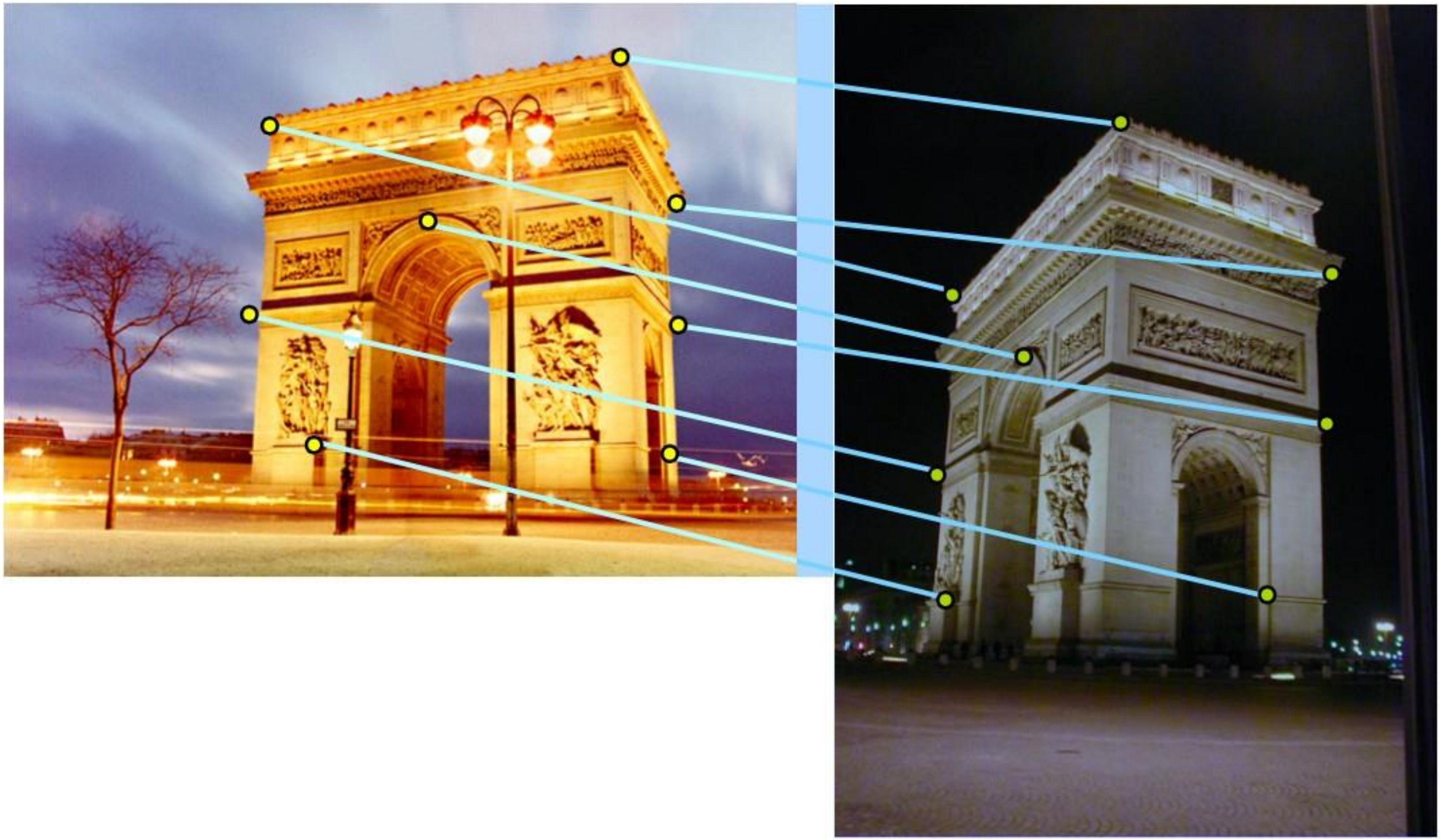
Image Features

03

Image Processing
Dr. Márton Szemenyei
Associate Professor
2024

Position/pose



Detection

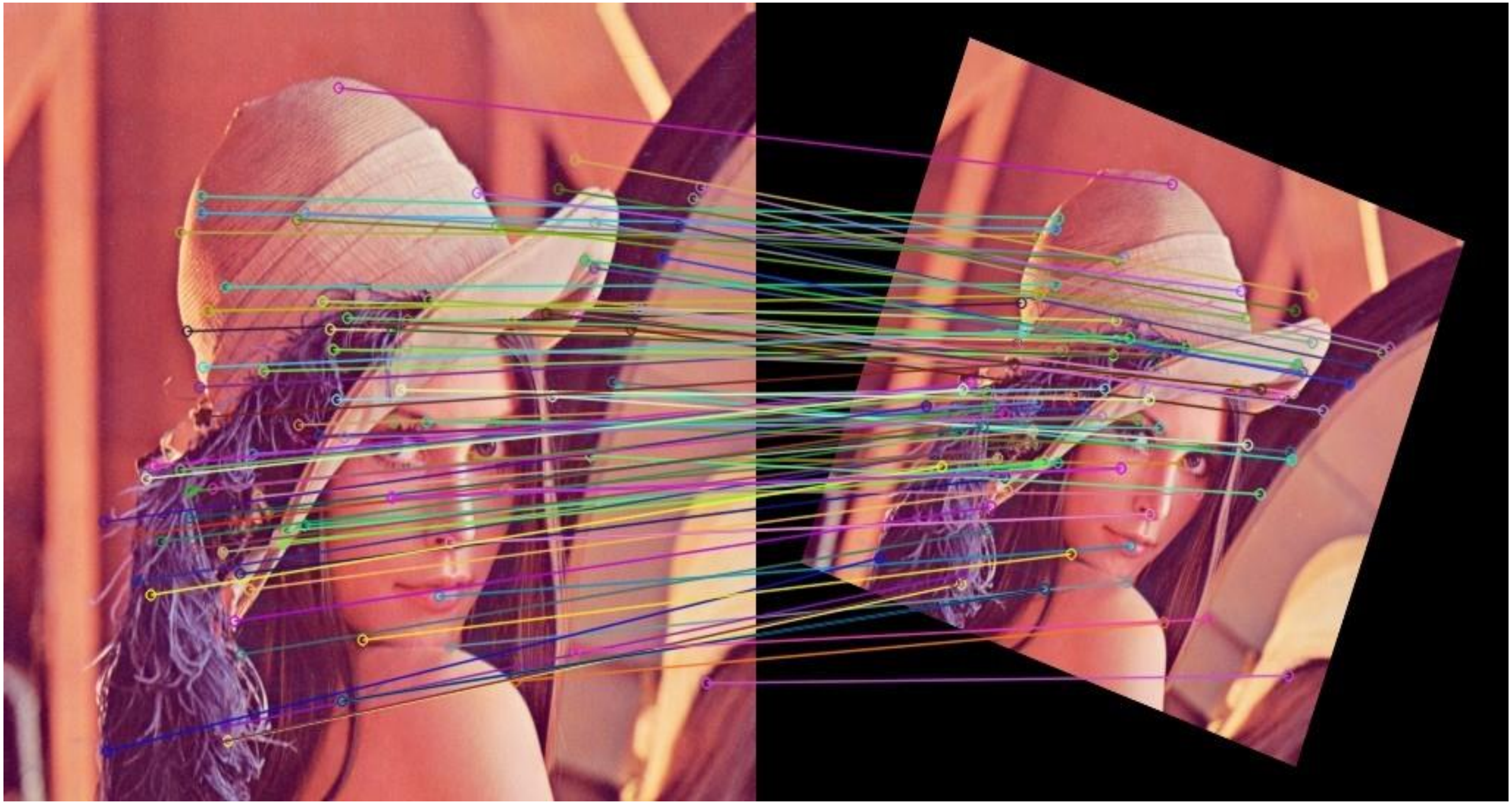
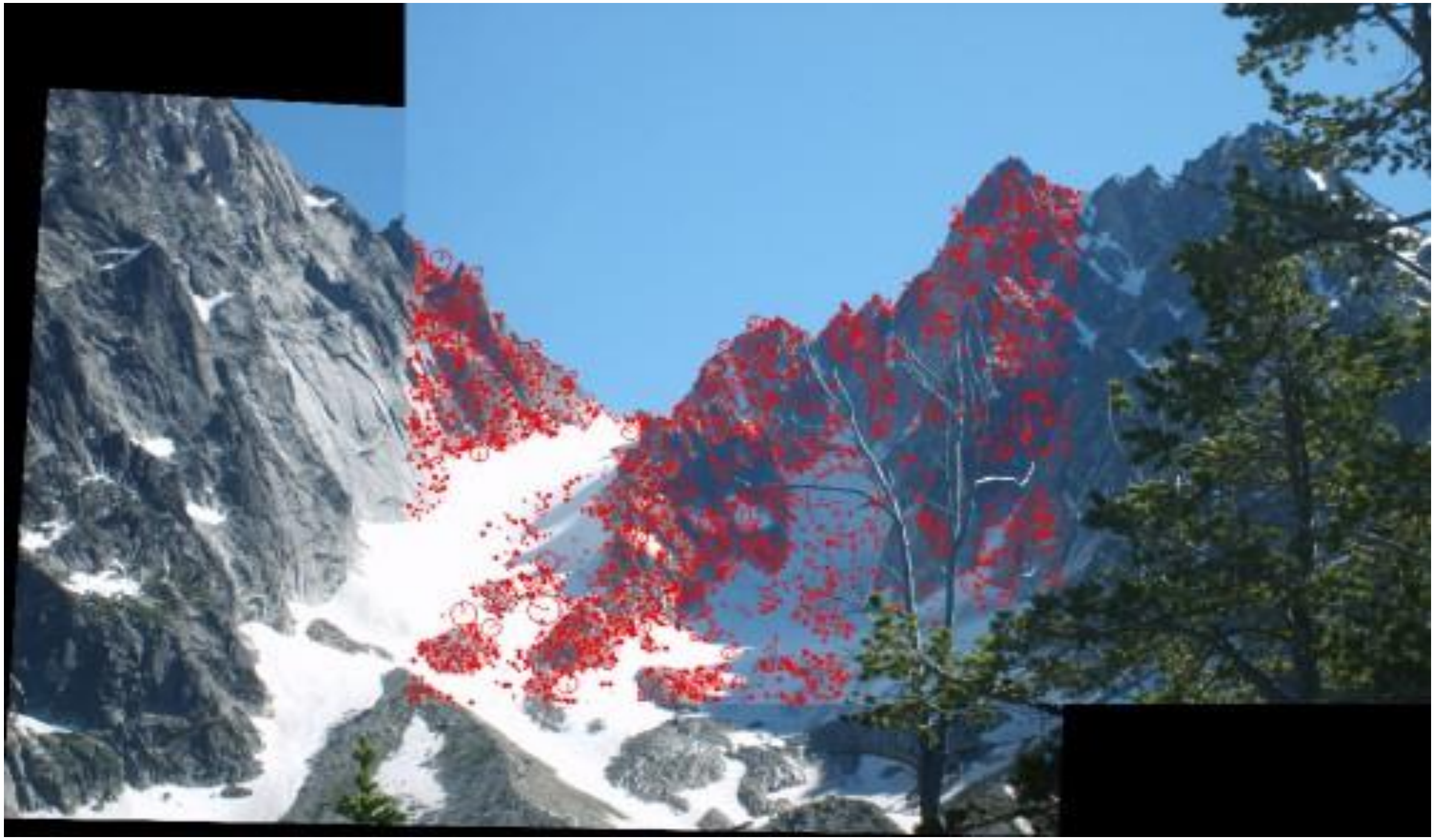
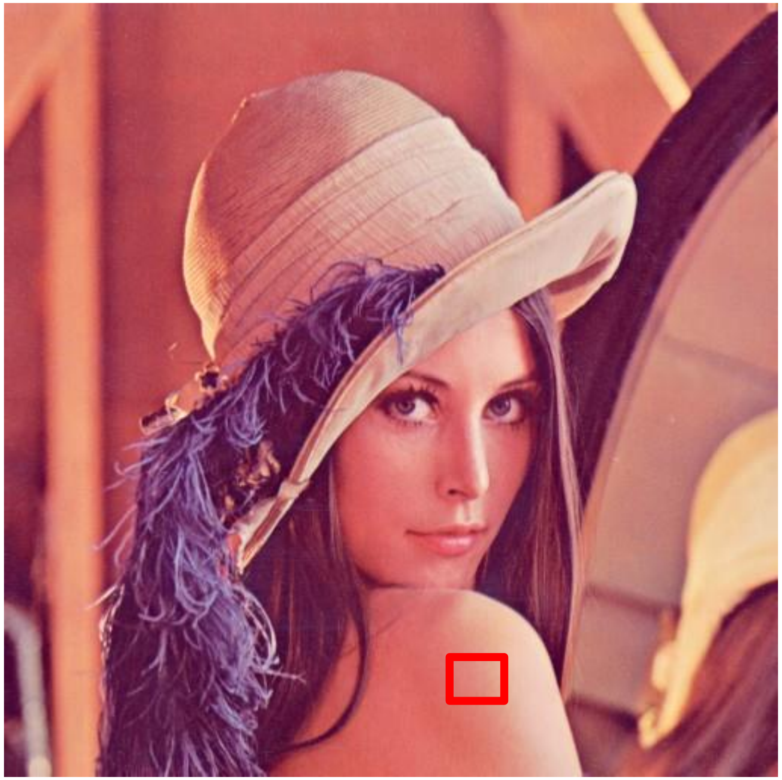


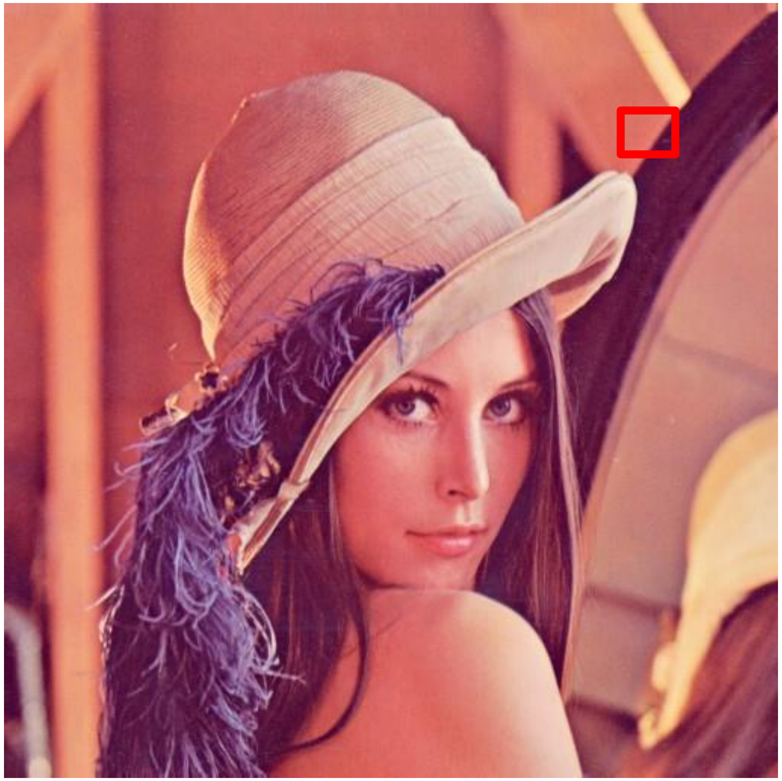
Image stitching

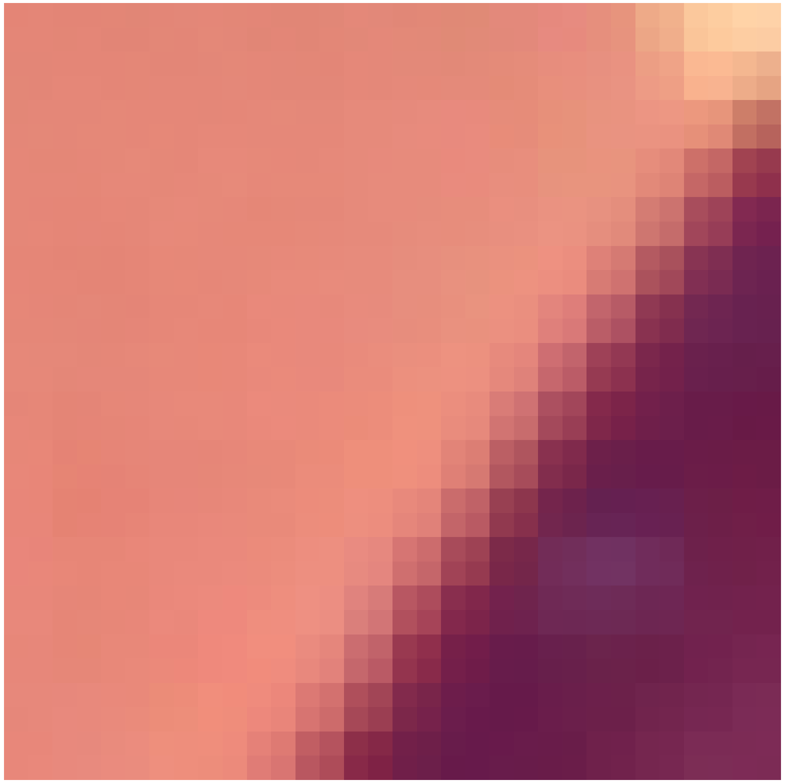


Intensity, colour

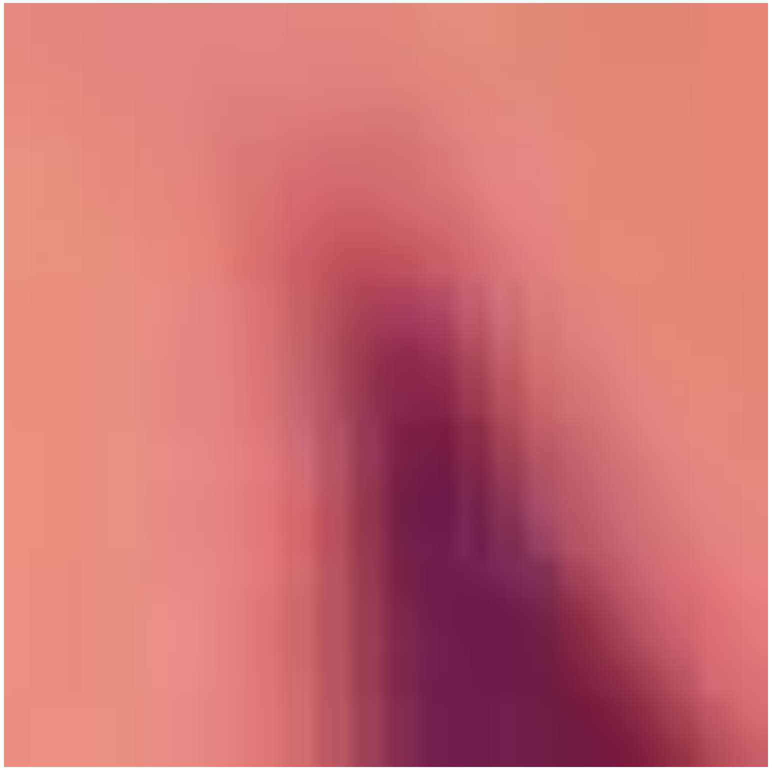


Edges

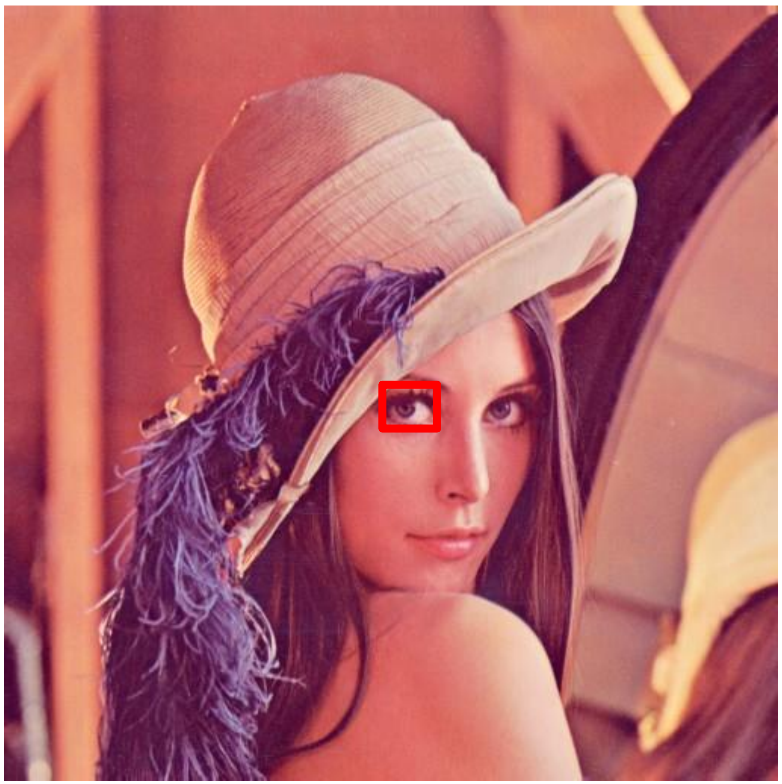


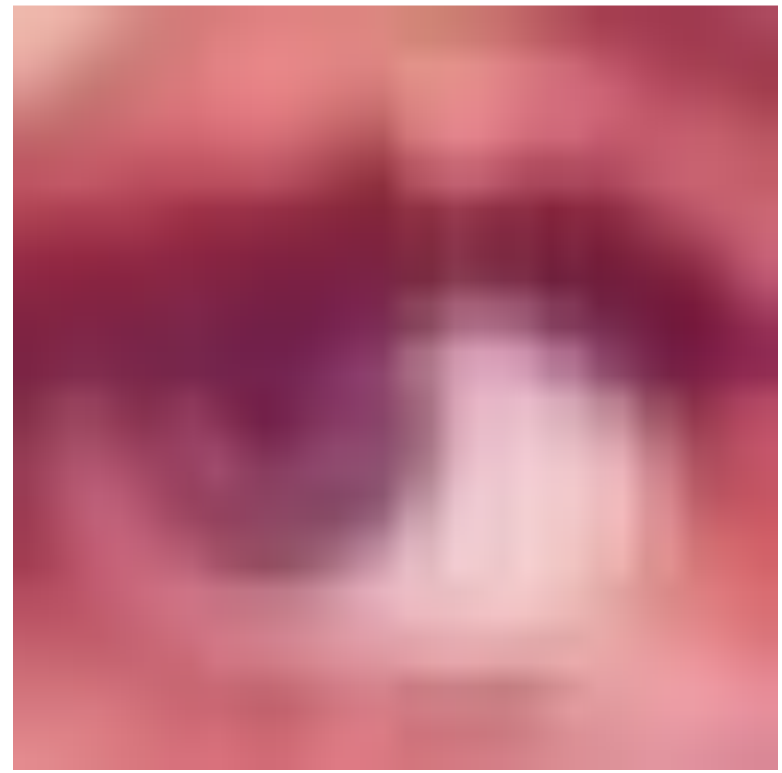


Corners



Regions

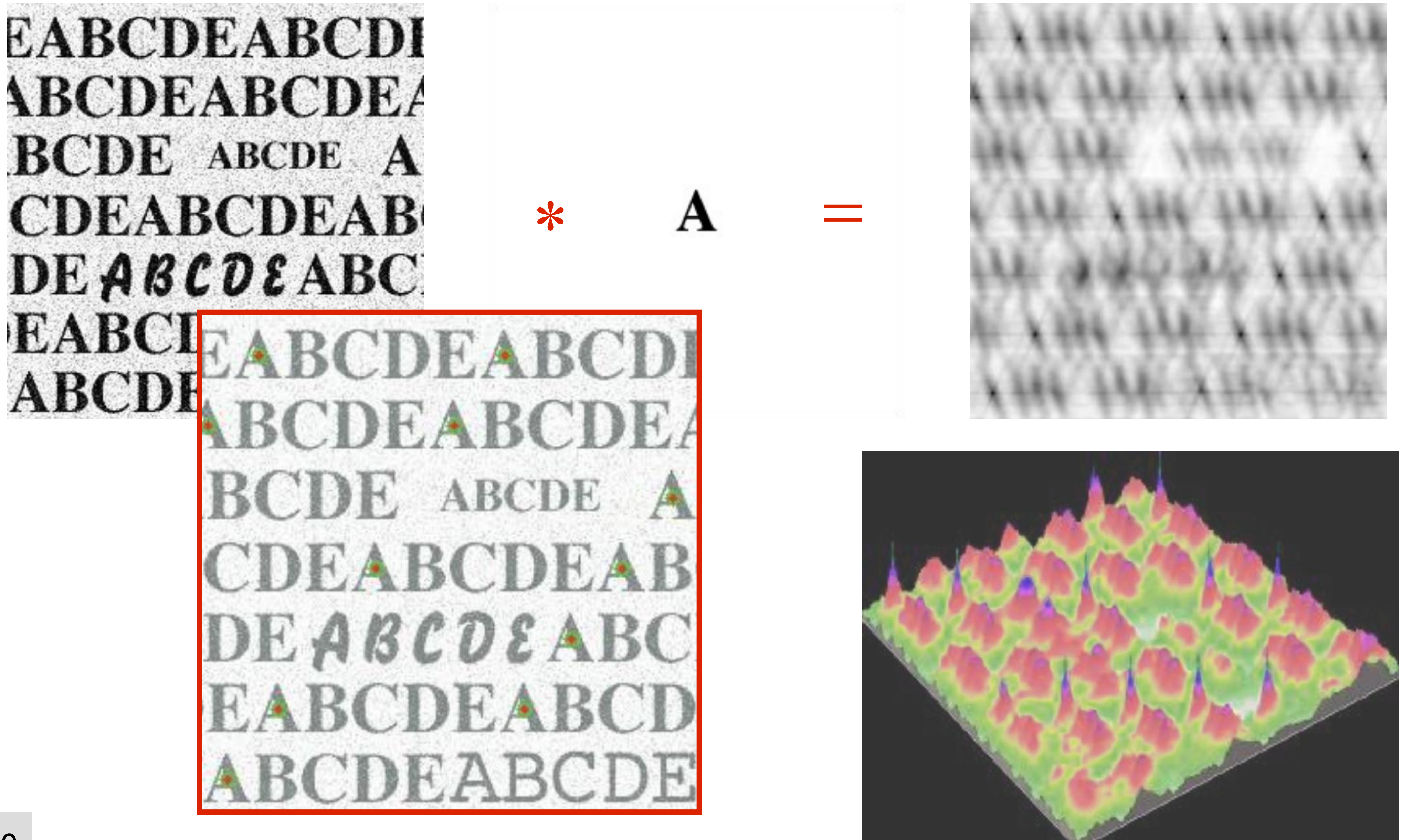




Usable features

1. Intensity (colour)
2. Gradients (edges)
3. Binary objects (later)
4. Corner points
5. Regions

Template matching



Template Matching

Template: convolutional filter

Error function

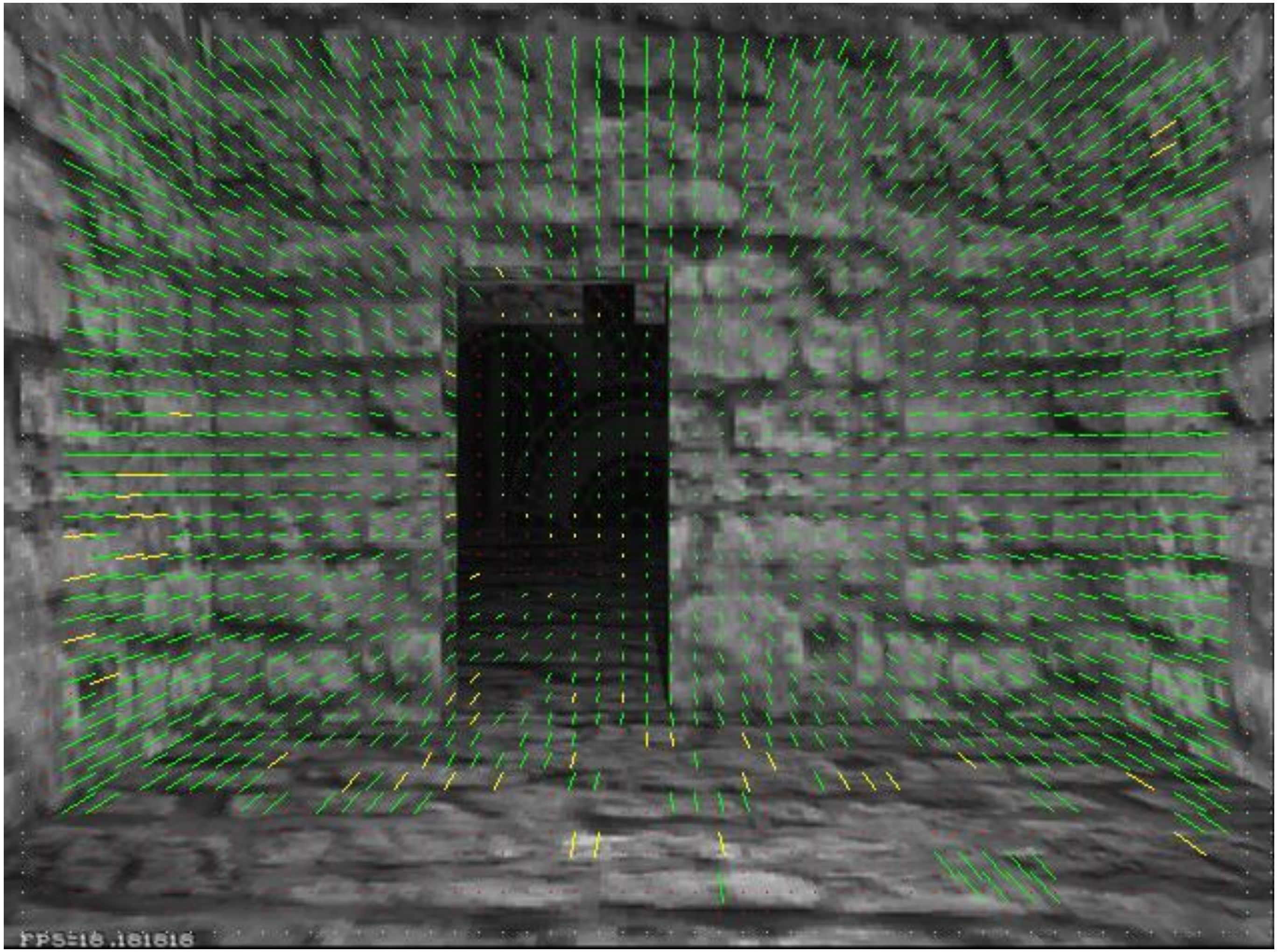
Correlation

$$E_{CC}(x,y) = \sum_{x'} \sum_{y'} I\,(x + x', y + y')T(x', y')$$

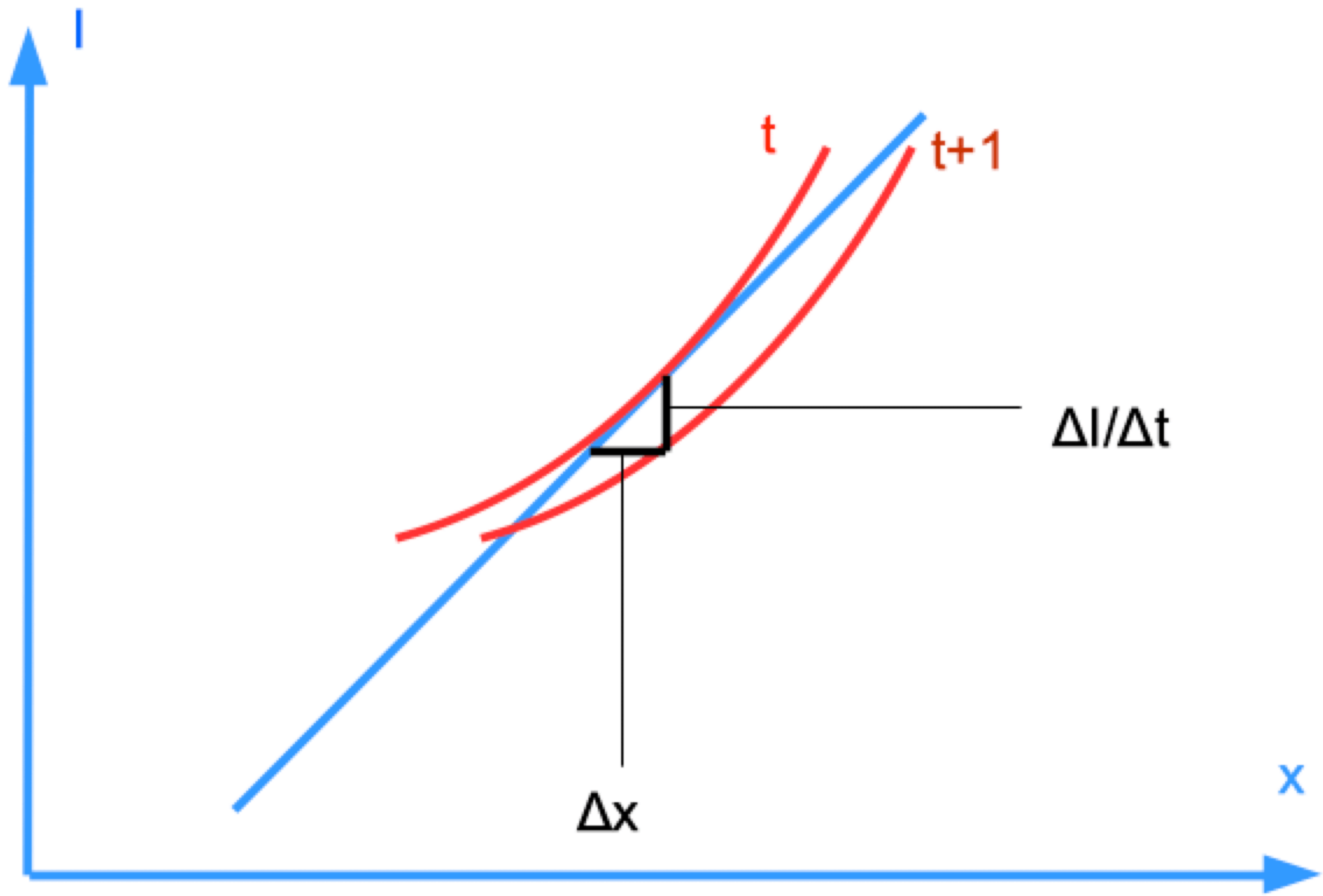
SSD

$$E_{SSD}(x,y) = \sum_{x'} \sum_{y'} \big(I(x + x', y + y') - T(x', y')\big)^2$$

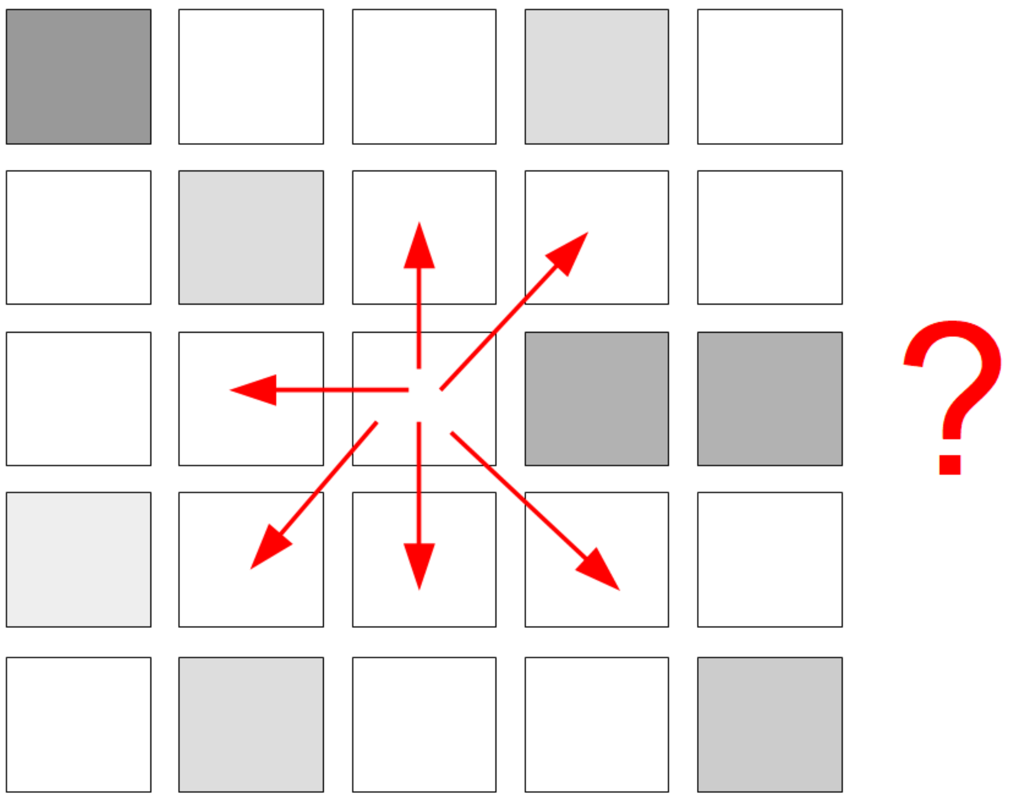
The OF field



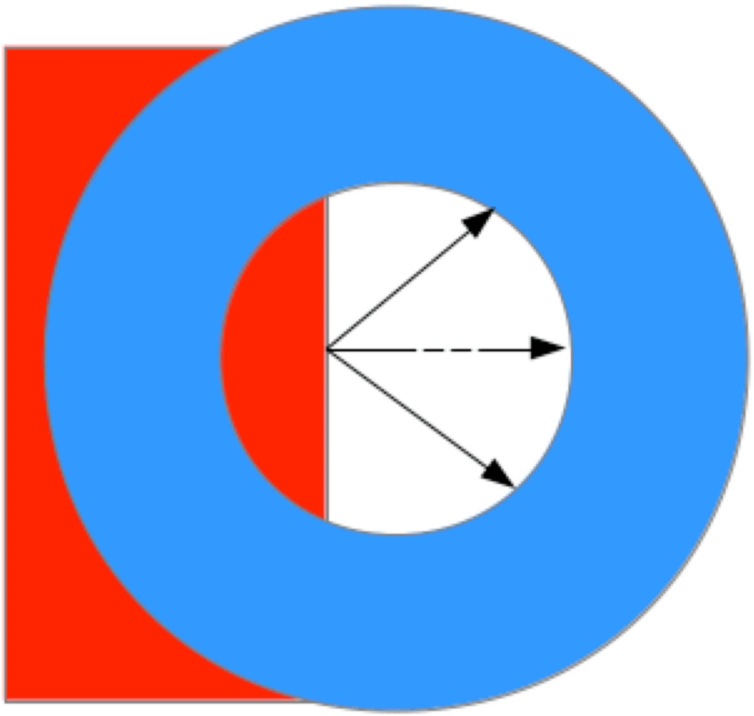
OF principle



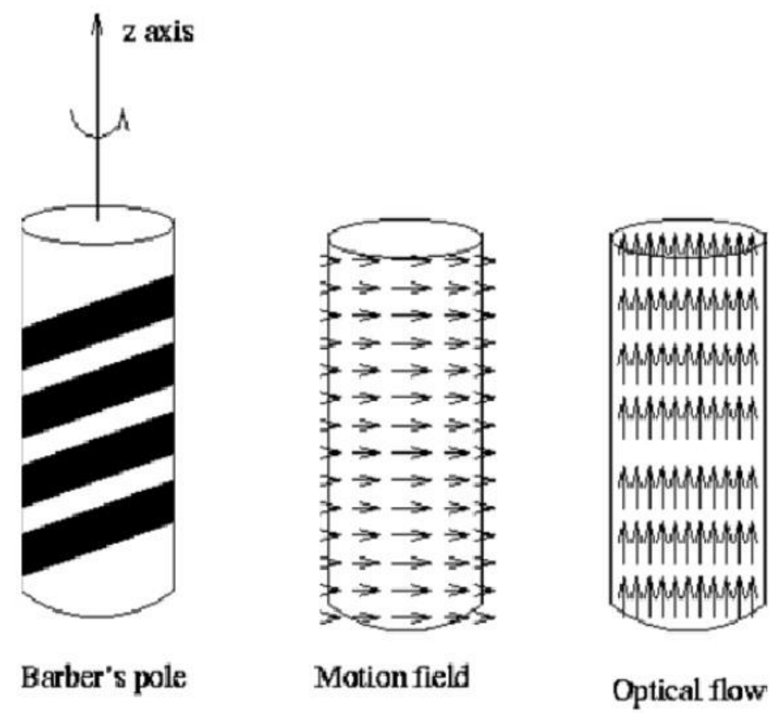
Homogeneous areas



The aperture problem



Barber pole illusion



Assumptions of the OF

The intensity of each object is constant over time

The displacement between two frames is small

We will need it later:

- Pixels close to each other move in a similar way

The intensity flow equation

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

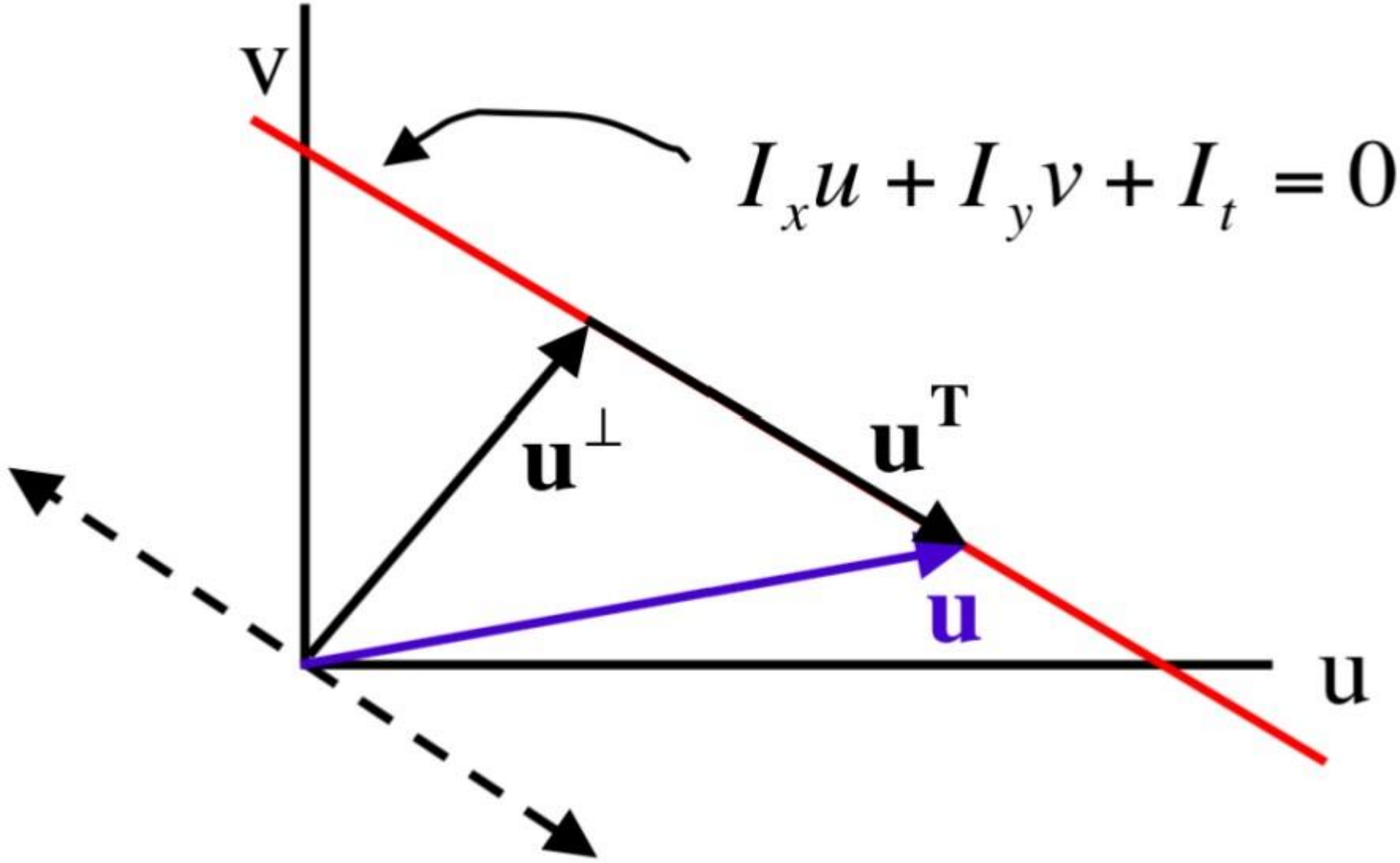
$$f(x + dx) = f(x) + f'(x)dx + \cancel{f''(x)\frac{dx^2}{2}} + \cdots$$

$$\cancel{I(x, y, t)} = \cancel{I(x, y, t)} + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = I_x dx + I_y dy + I_t dt$$

$$I_x \frac{dx}{dt} + I_y \frac{dy}{dt} + I_t = 0 \;\rightarrow\; I_x u + I_y v = -I_t$$

$$v = -u\frac{I_x}{I_y} - \frac{I_t}{I_y}$$

The solvability



$$d = \frac{I_t}{\sqrt{I_x^2 + I_y^2}}$$

Lucas-Kanade

for N pixels

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xN} & I_{yN} \end{bmatrix} \binom{u}{v} = \begin{pmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tN} \end{pmatrix}$$

$$X\vec{u} = Y \;\rightarrow\; \vec{u} = (X^T X)^{-1} X^T Y$$

$$X^T X : H$$

Local structure matrix: covariance matrix of derivatives

Farneback Optical Flow

We approximate the image with a quadratic function

$$I_1(x) = x^T A_1 x + b_1^T x + c_1 \; I_2(x) = x^T A_2 x + b_2^T x + c_2$$

The two images are the same only shifted:

$$I_2(x) = I_1(x-d) = (x-d)^T A_1 (x-d) + b_1^T (x-d) + c_1 = \cdots$$
$$\ldots = x^T A_1 x + (b_1 - 2A_1 d)^T x + d^T A_1 d - b_1^T d + c_1$$

Finally:

$$b_2 = b_1 - 2A_1 d \; \rightarrow \; d = \frac{-1}{2} A_1^{-1}(b_2 - b_1)$$

Farneback in practice

The picture is not a quadratic function

- Estimate polynomials locally, not globally

Estimates per pixel: too noisy

- Use LS estimate in the neighbourhood

$$d = \left(\sum_{p \in N(x)} w_p \, A_p^T A_p \right)^{-1} \sum_{p \in N(x)} w_p \, A_p^T (b_2 - b_1)$$

Farneback vs LK

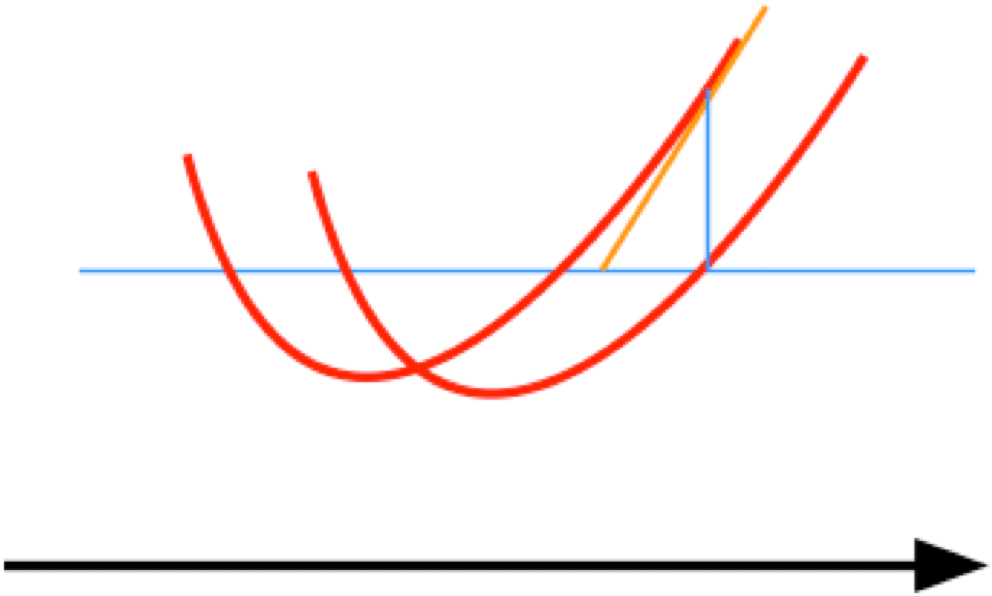
Dense optical flow: Farneback

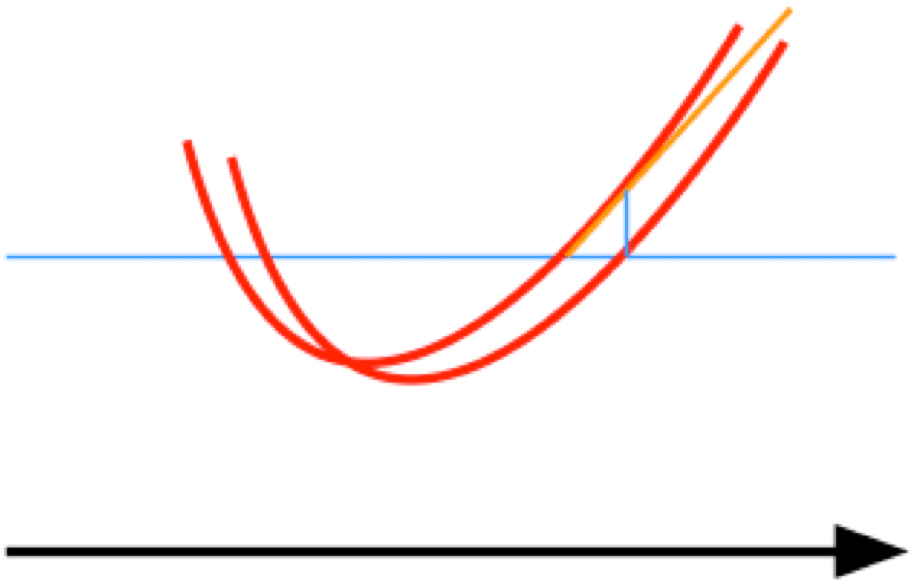
- For each pixel position we calculate the movement

Rare optical flow: LK

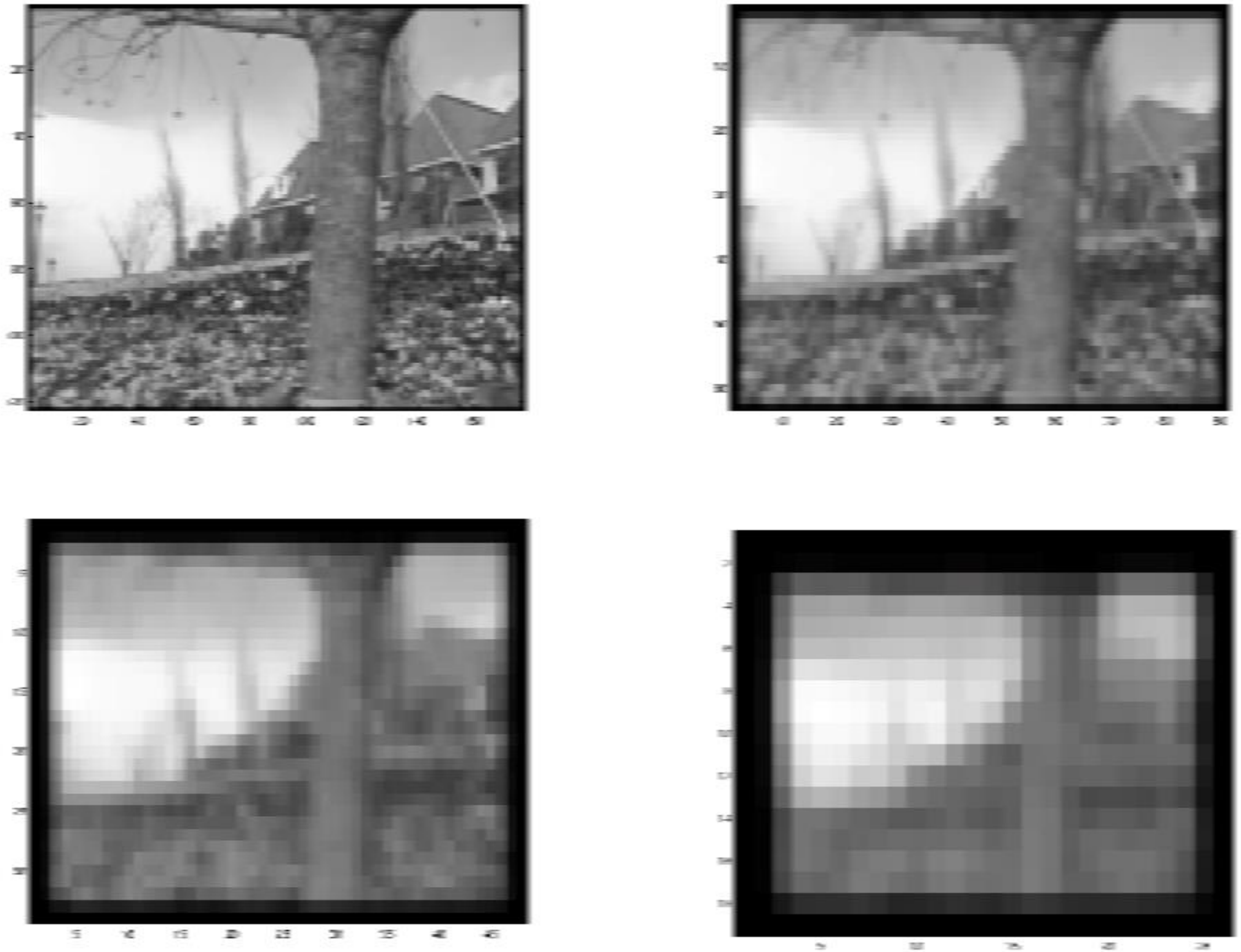
- We count movements only at a few selected points

Iterative OF in 1D

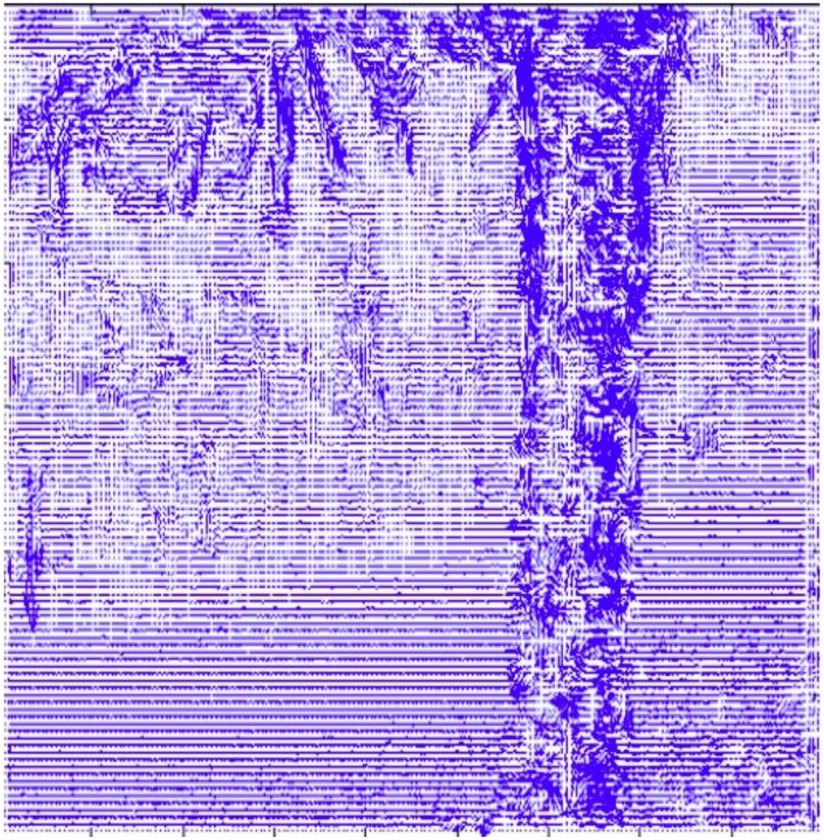


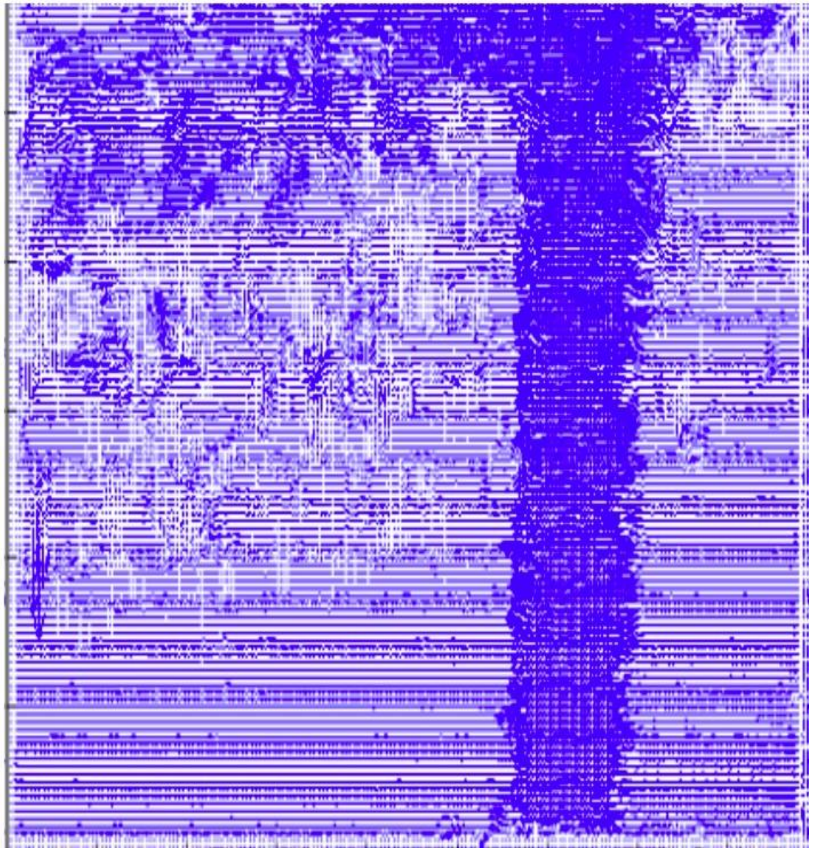


Pyramid OF



Optical Flow with pyramid





Difficulties

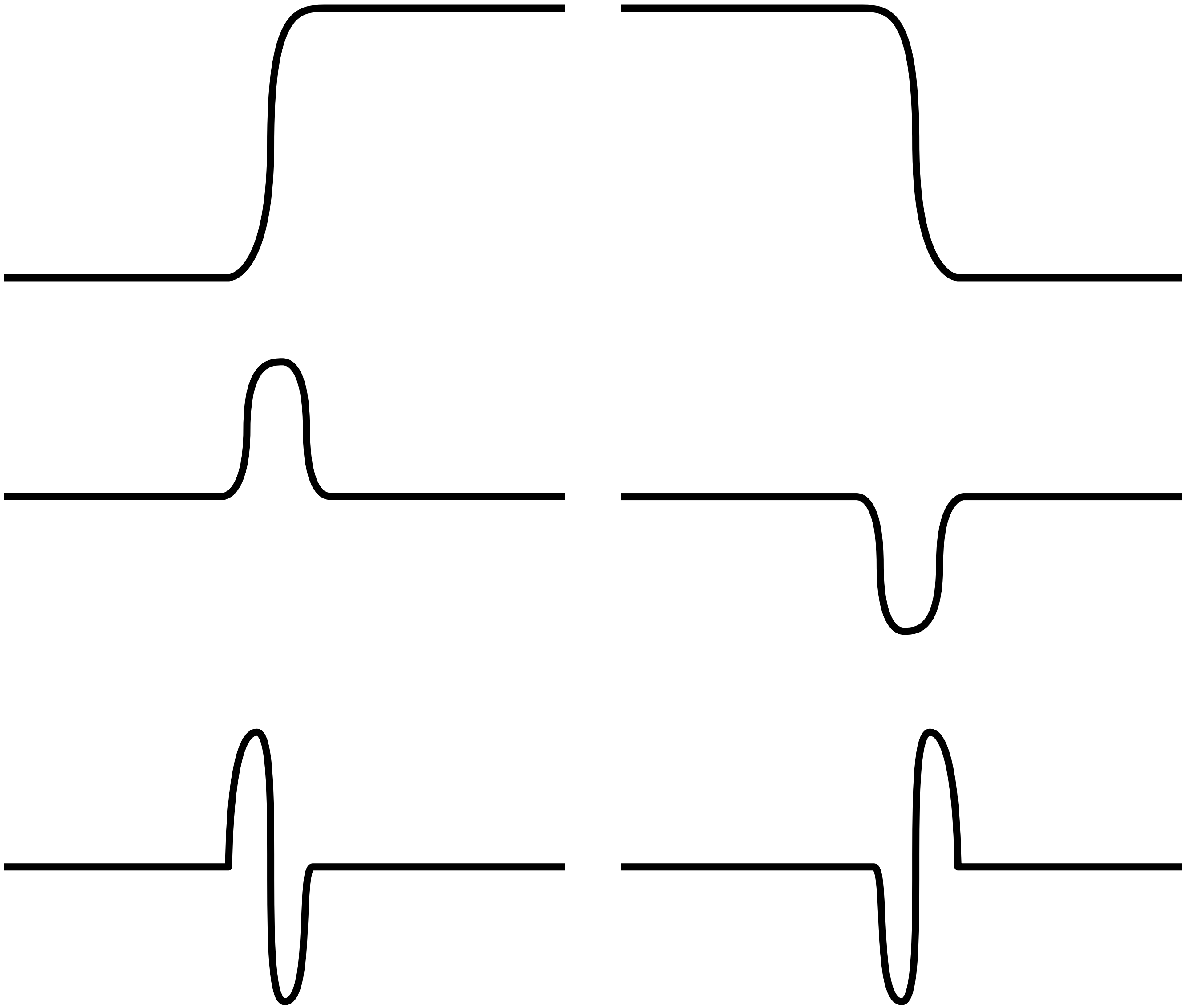
Object changes (rotation, scale, etc.)

Occlusion

Non-linear motion

Similar objects

Edge search with derivatives



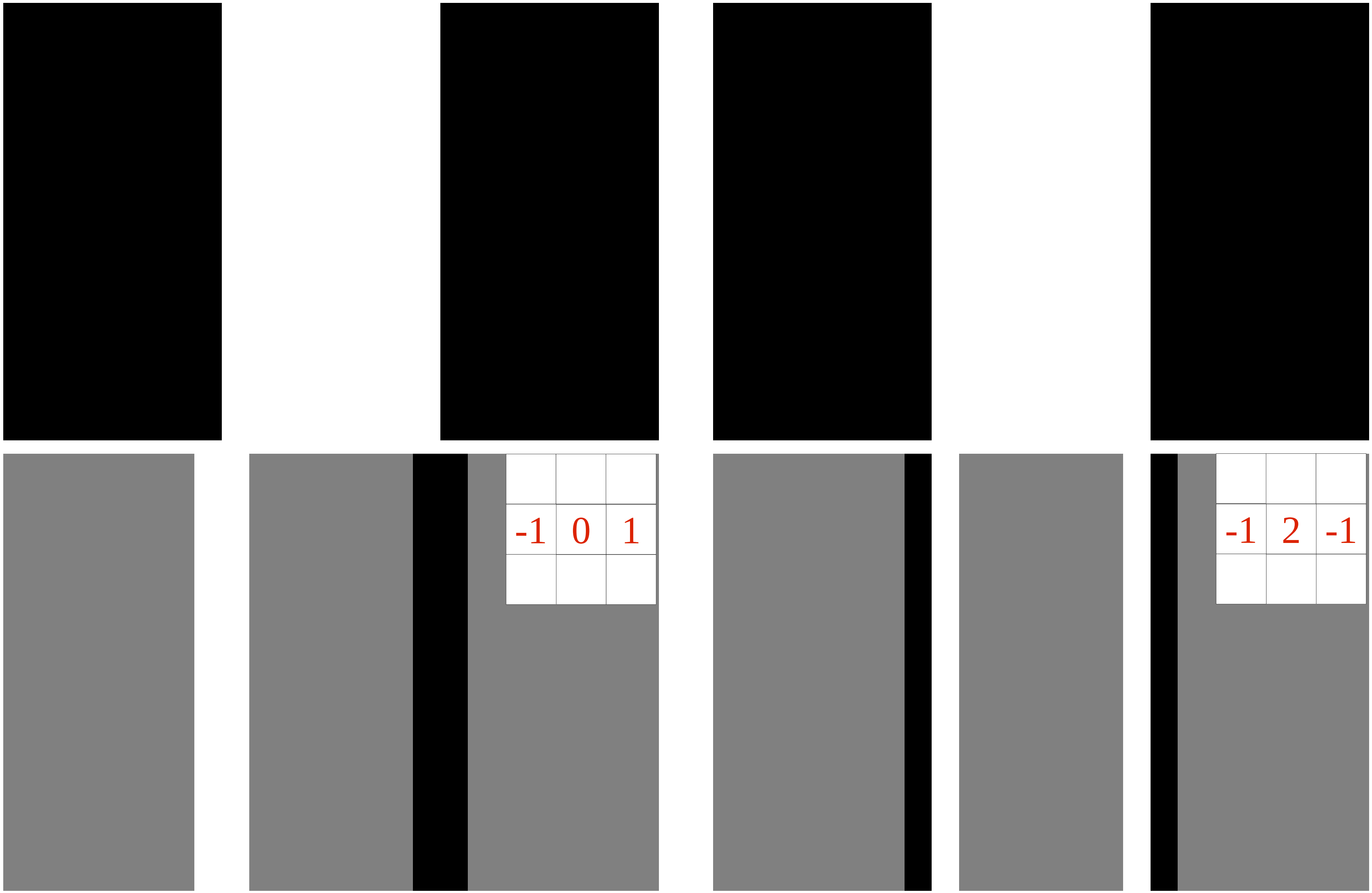
-1	1

-1	
	1

-1	0	1

-1		
	0	
		1

Edge search with derivatives



Edge search with derivatives

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

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

1	-1	-1
2	1	-1
1	-1	-1

5	-3	-3
5	0	-3
5	-3	-3

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

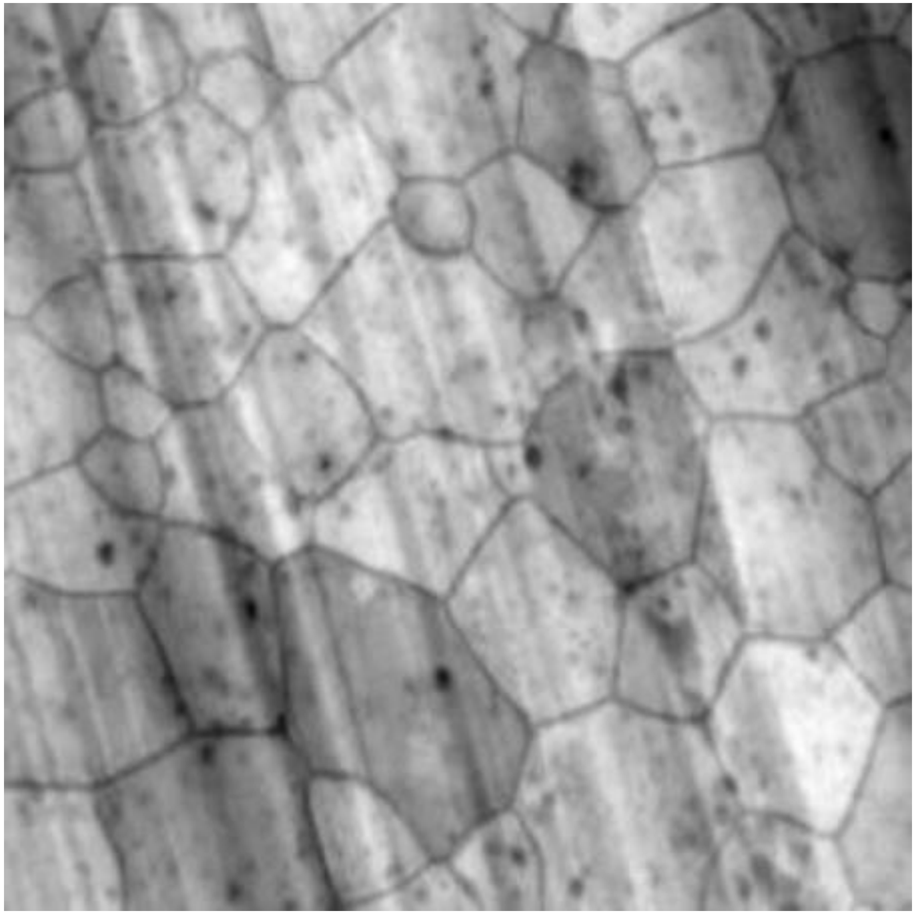
2	1	0
1	0	-1
0	-1	-2

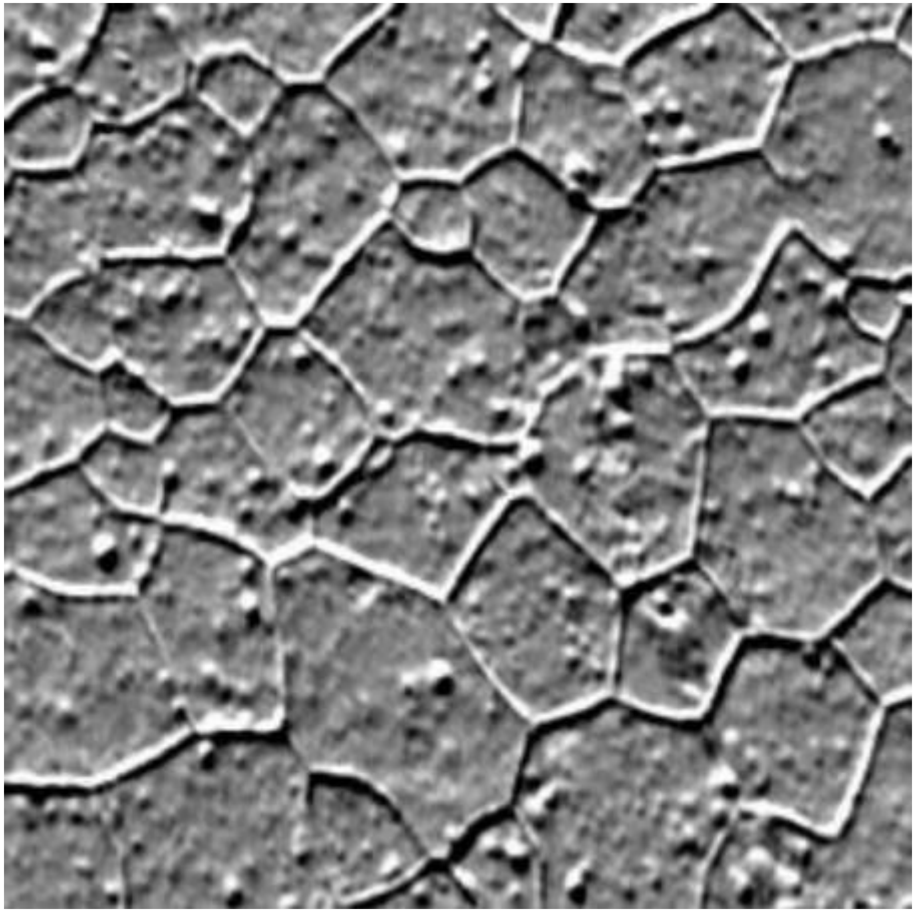
2	1	-1
1	1	-1
-1	-1	-1

5	5	-3
5	0	-3
-3	-3	-3

• • •

Derivative filter

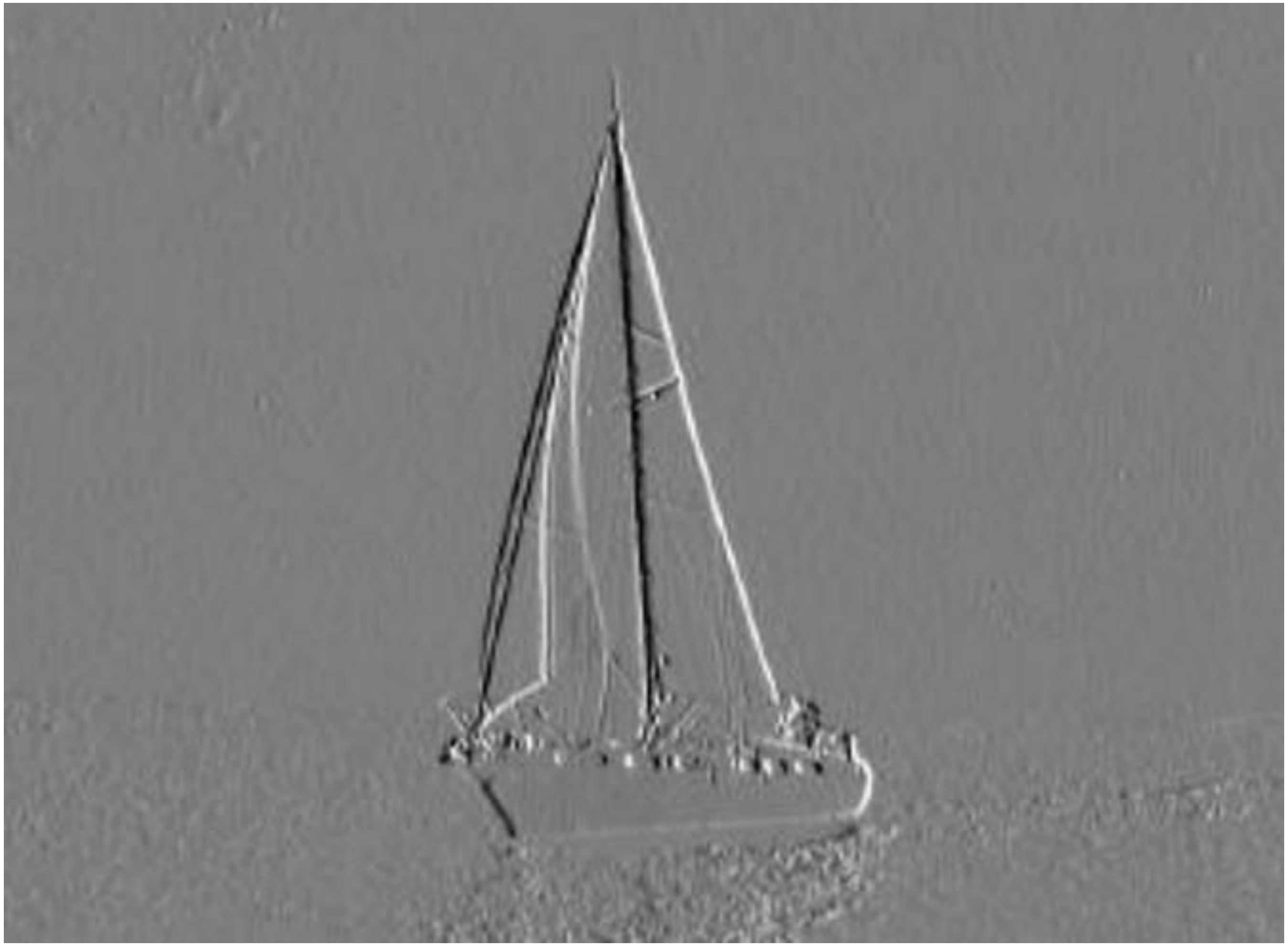




Complex example

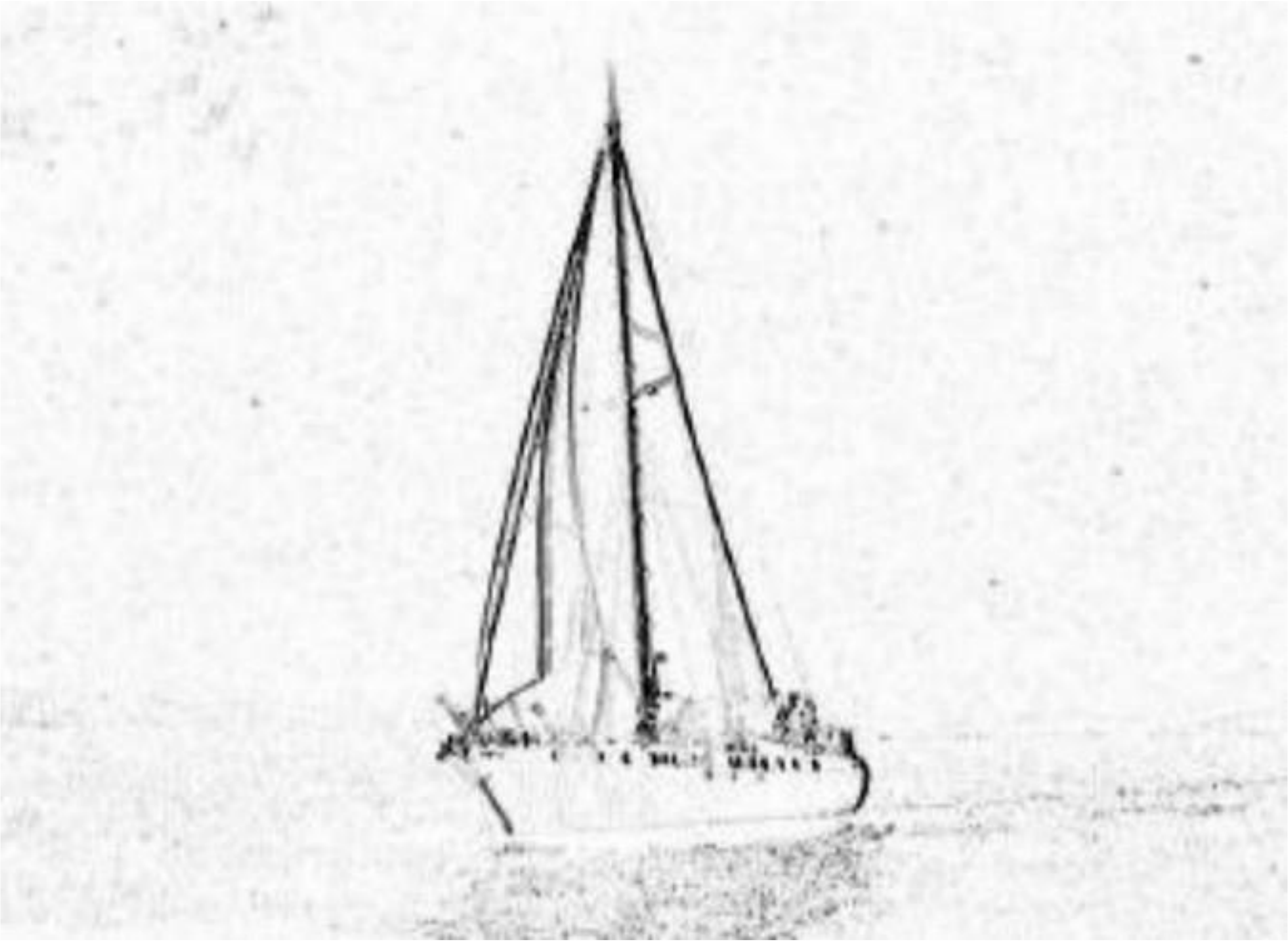
Initial image

Complex example



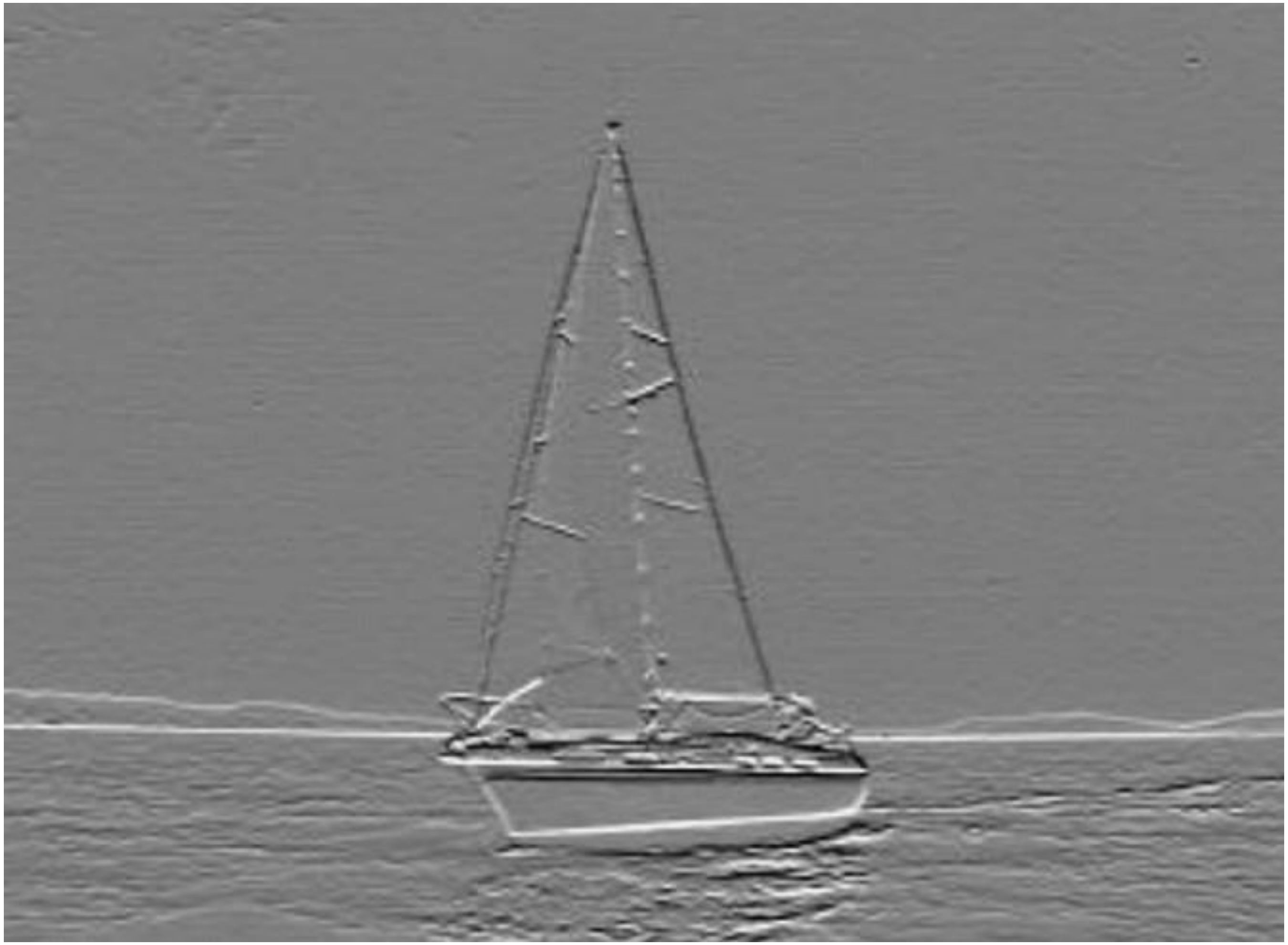
Horizontal derivative

Complex example



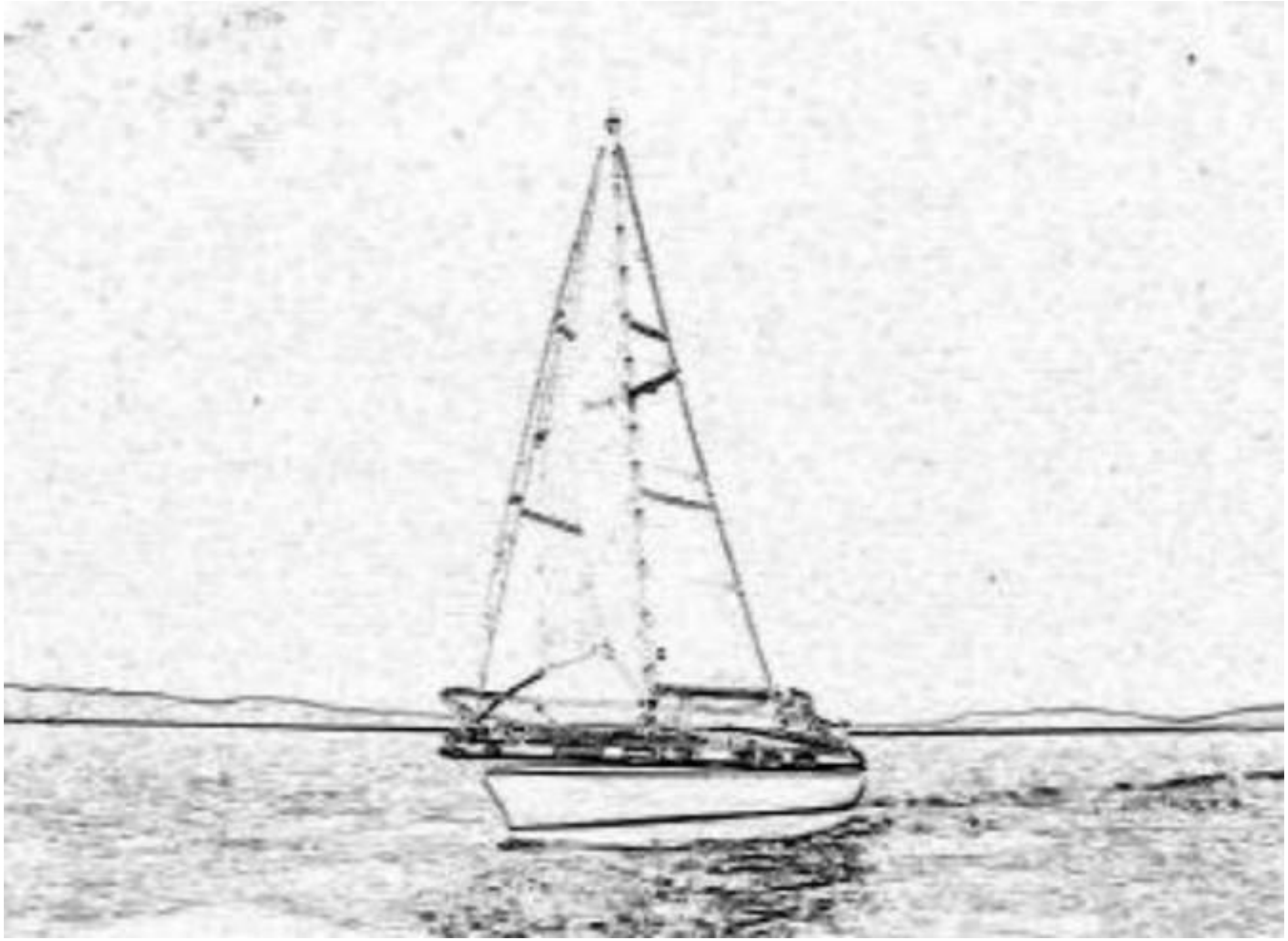
Horizontal derivative absolute value

Complex example



Vertical derivative

Complex example

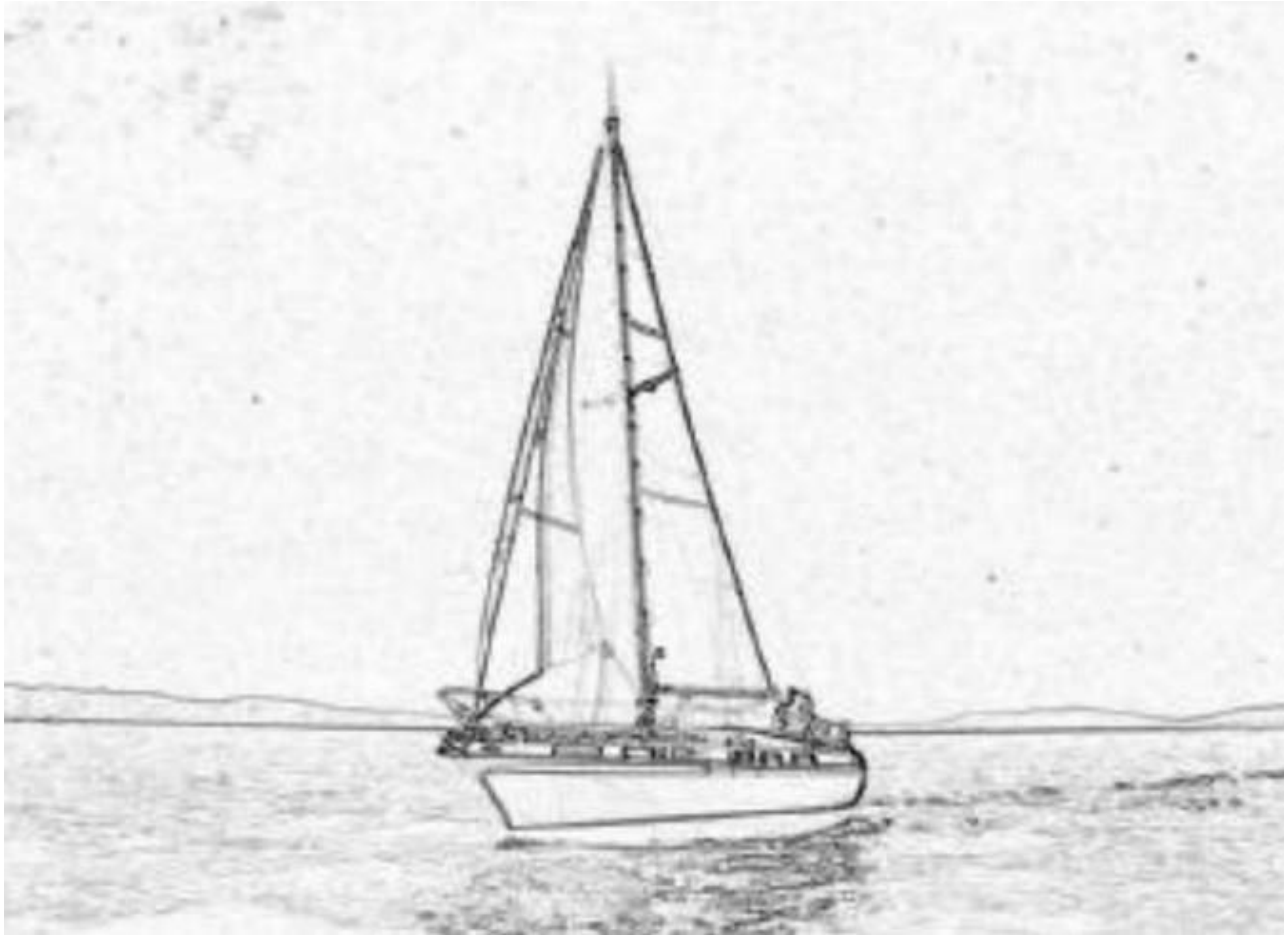


Absolute value of vertical derivative

Complex example

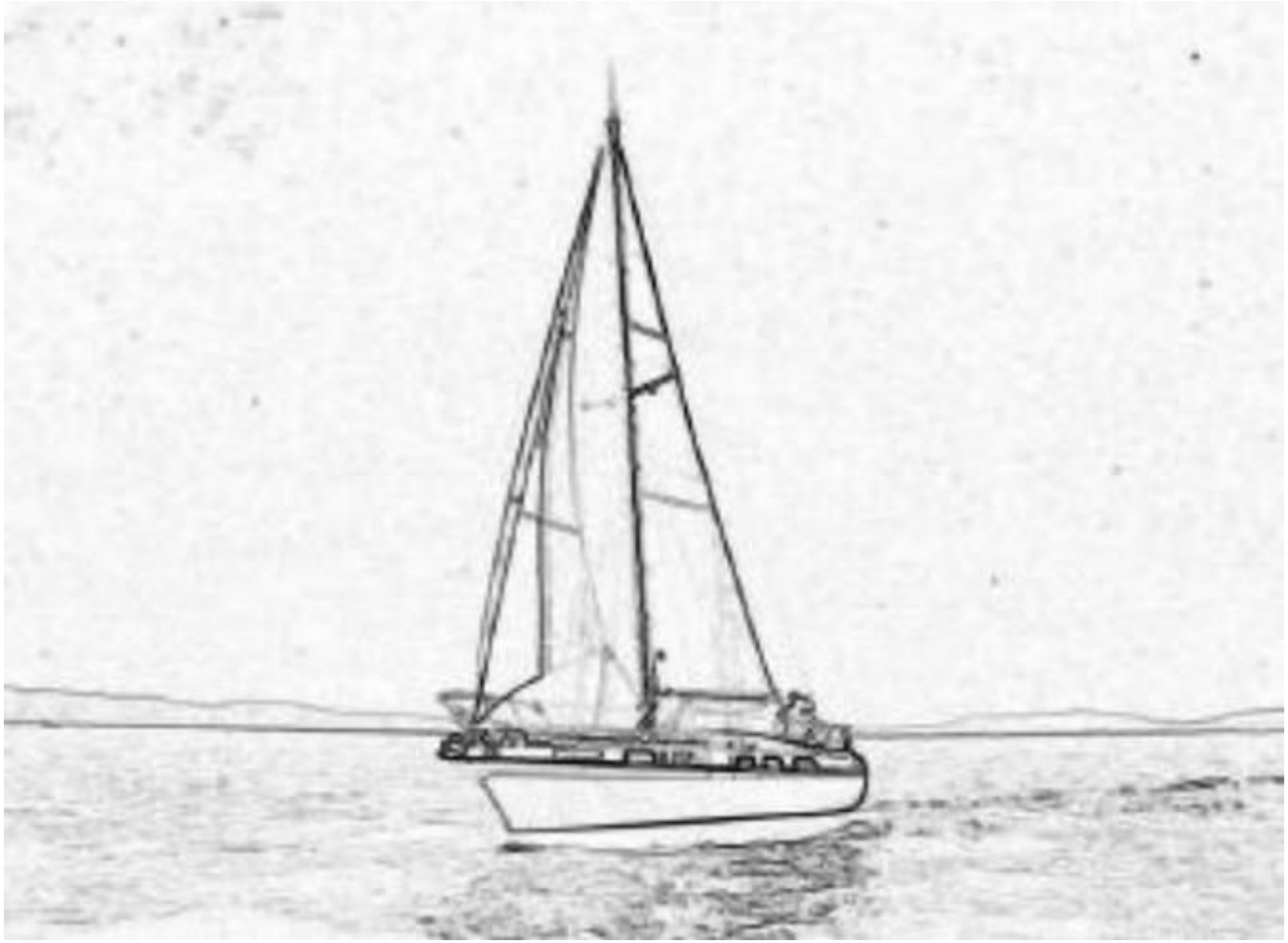
Max of absolute values

Complex example



Sum of absolute values

Complex example



Euclidean norm of absolute values

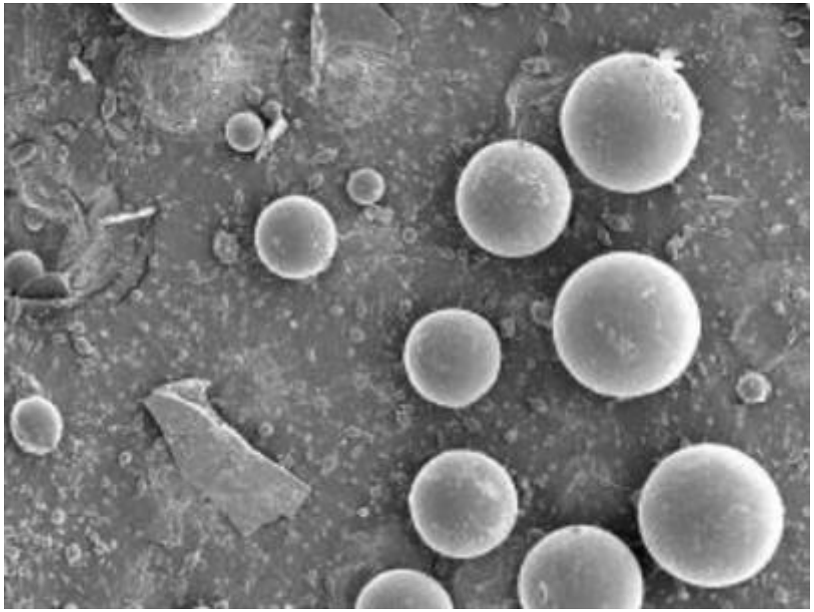
Difference of Gaussians (DoG)

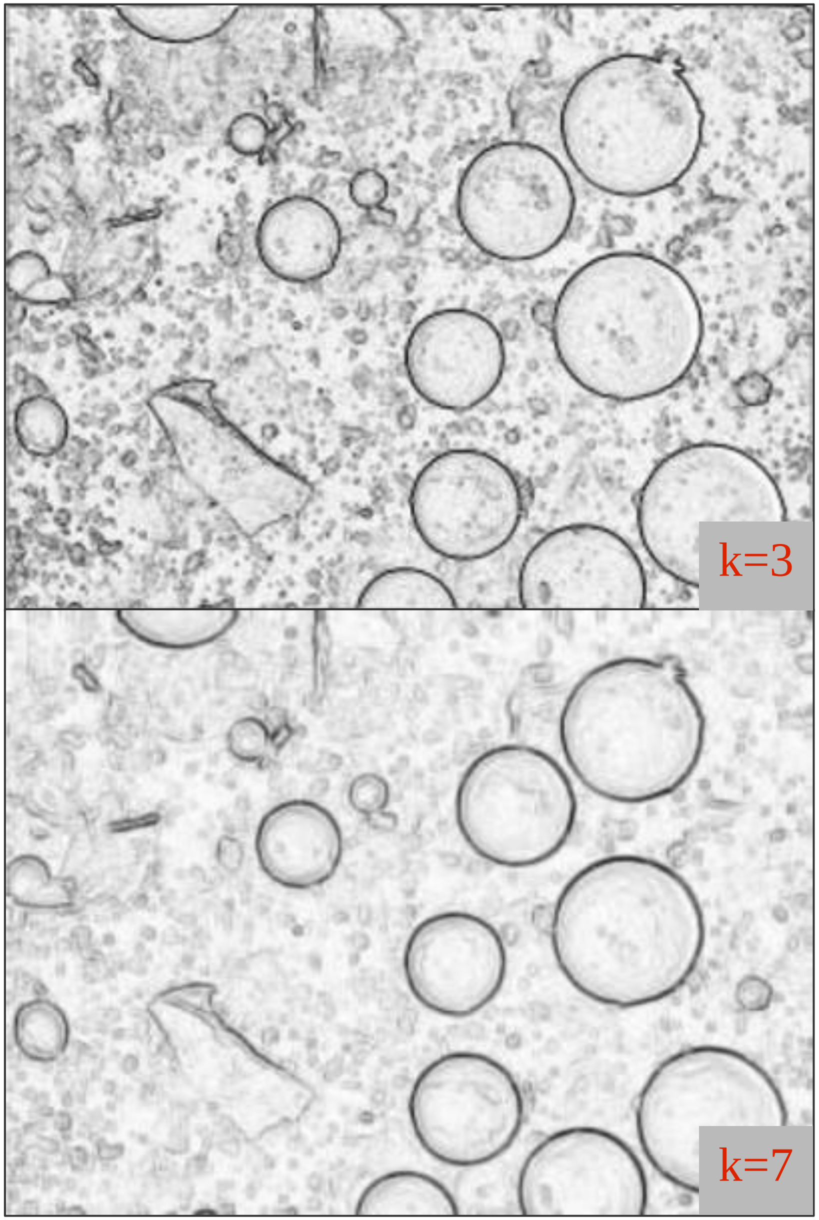
Laplace Filter

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Kernel size





Canny edge detector

Gaussian filtering

Derivative filtering in two/four directions

- (Roberts, Prewitt, Sobel...)
- Gradient magnitude and direction

Delete non-maxima (in gradient direction)

Hysteresis thresholding: strong edges, weak edges

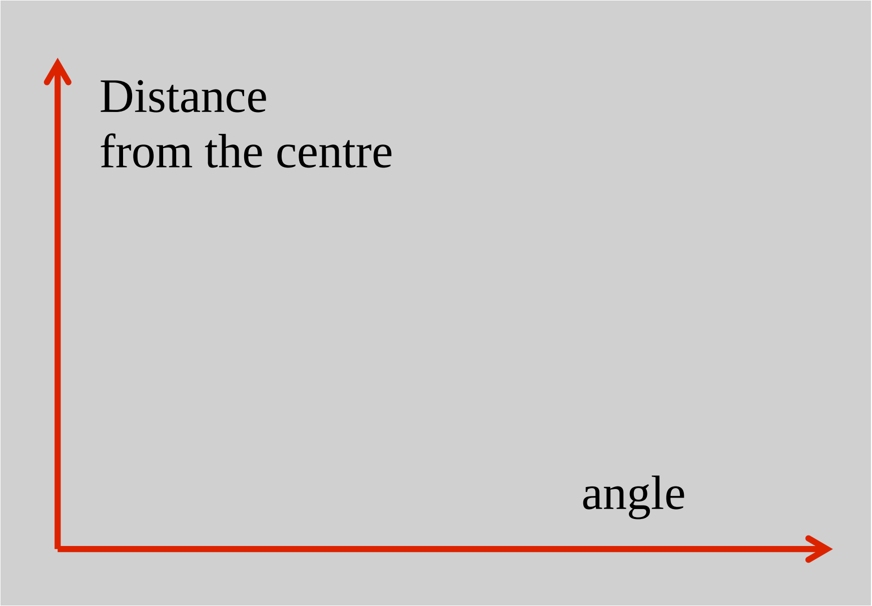
Canny edge detector

Directions, angles

Hough transform

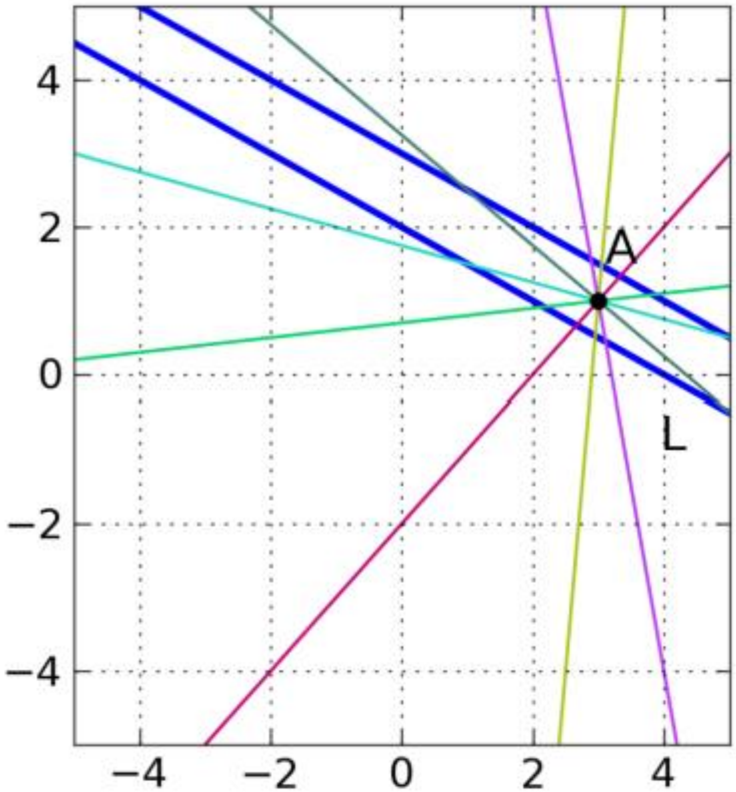
Object is a single point in Hough Space

- **Lines** (primarily!)
- Circles
- Ellipses
- Arbitrary objects



Hough transform

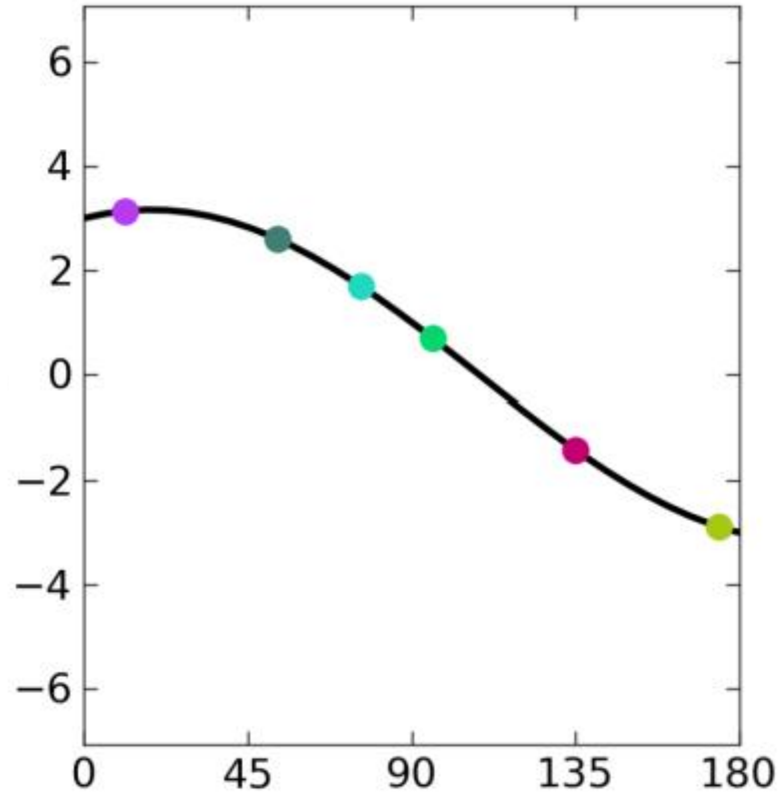
Original image



y

x

Hough Space



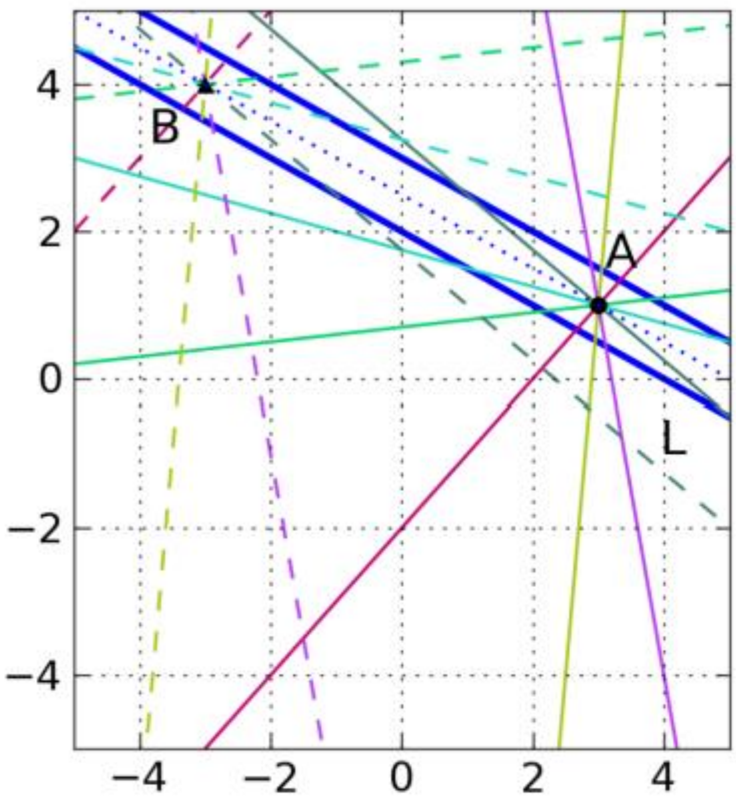
d

θ

Hough transform

Original image

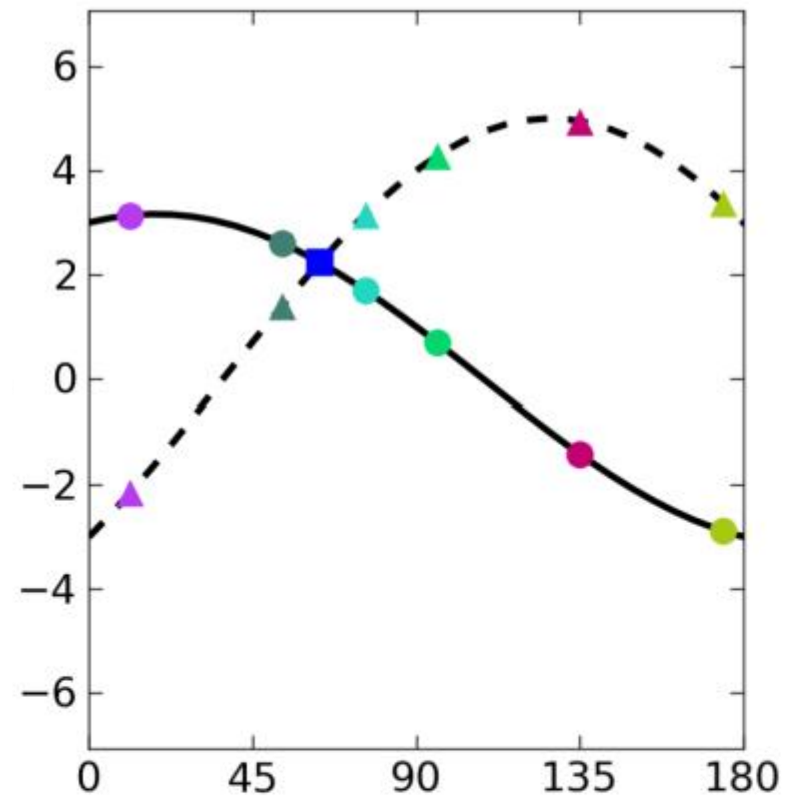
y



x

Hough Space

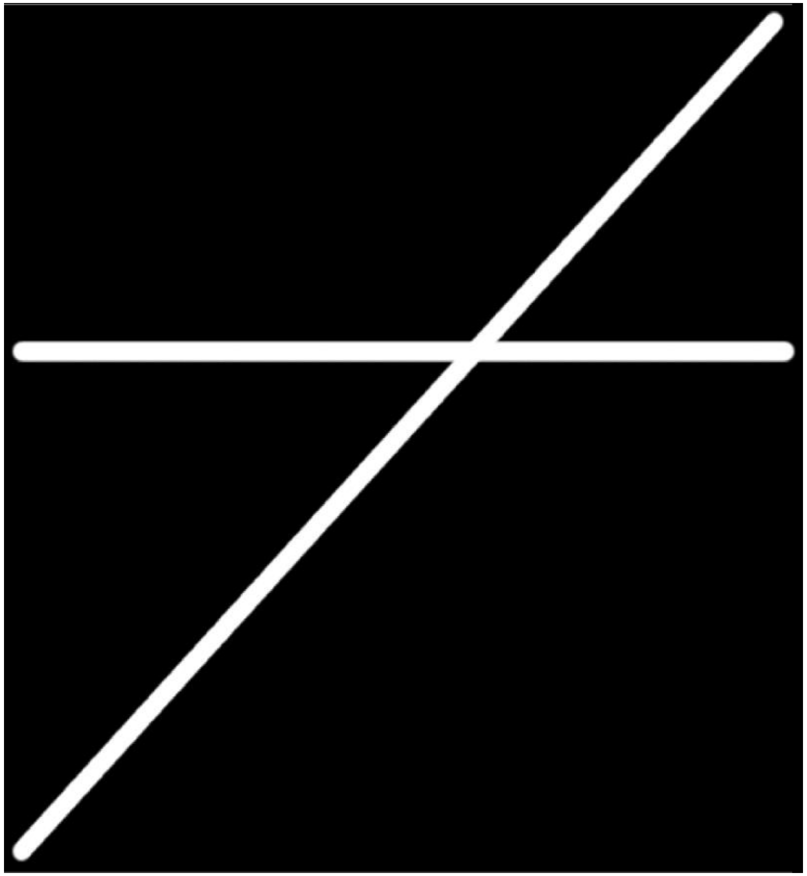
d



θ

Hough transform

Original image



Hough Space