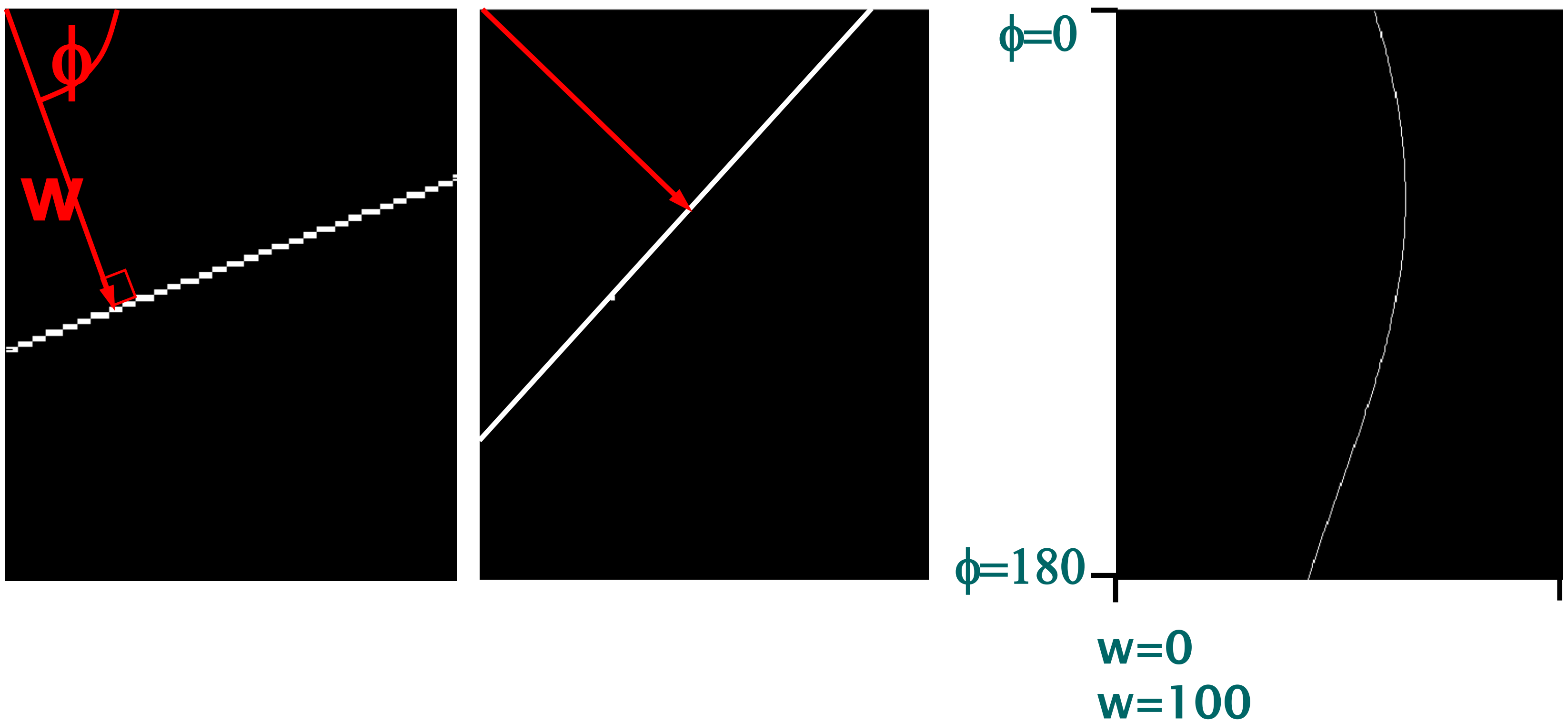
Finding lines in an image

$$w = x\cos(\phi) + y\sin(\phi)$$



Hough transform algorithm

Basic Hough transform algorithm

1. Initialize H[d, θ]=0
2. for each edge point I[x,y] in the image

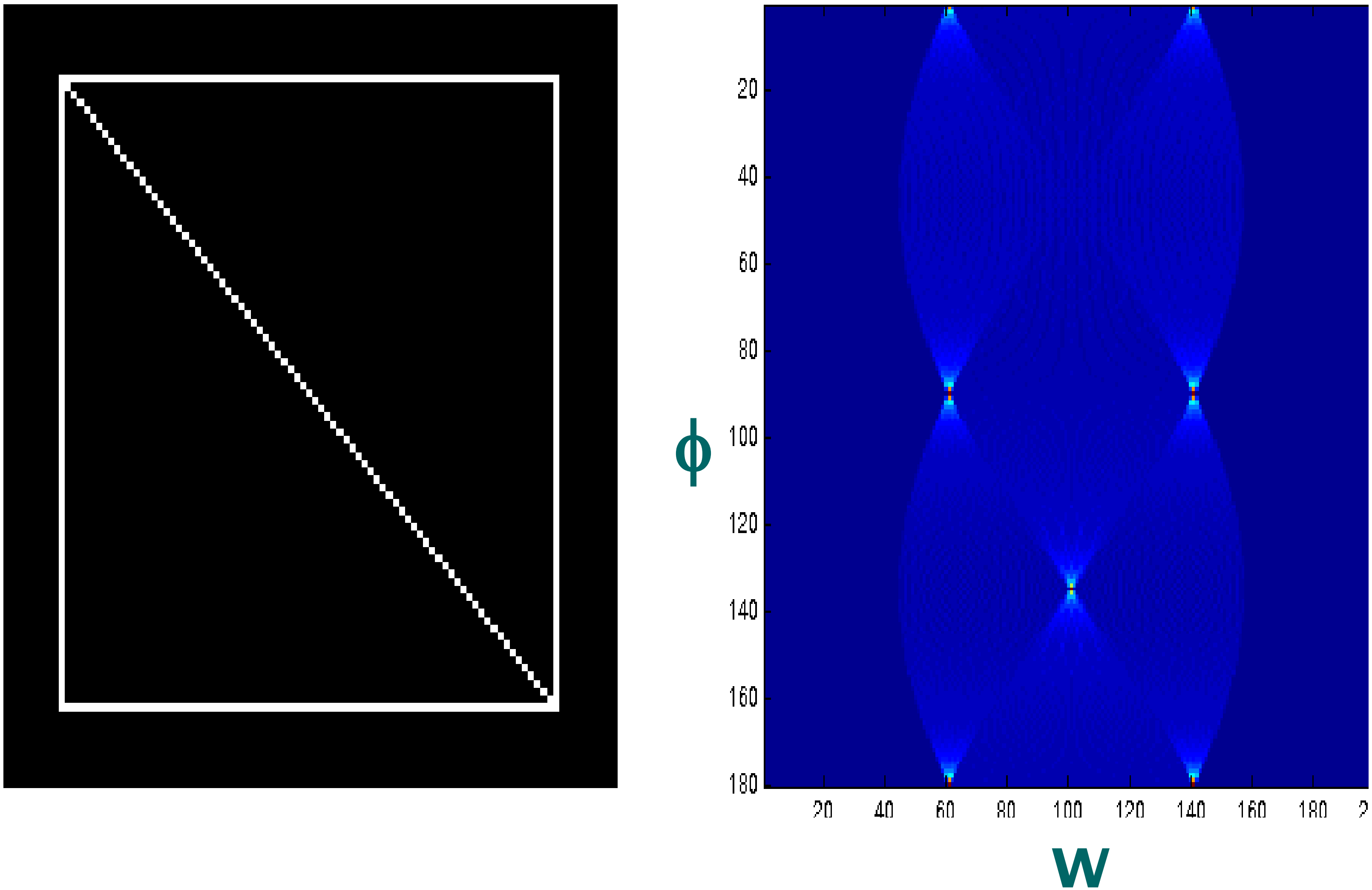
    for θ = 0 to 180

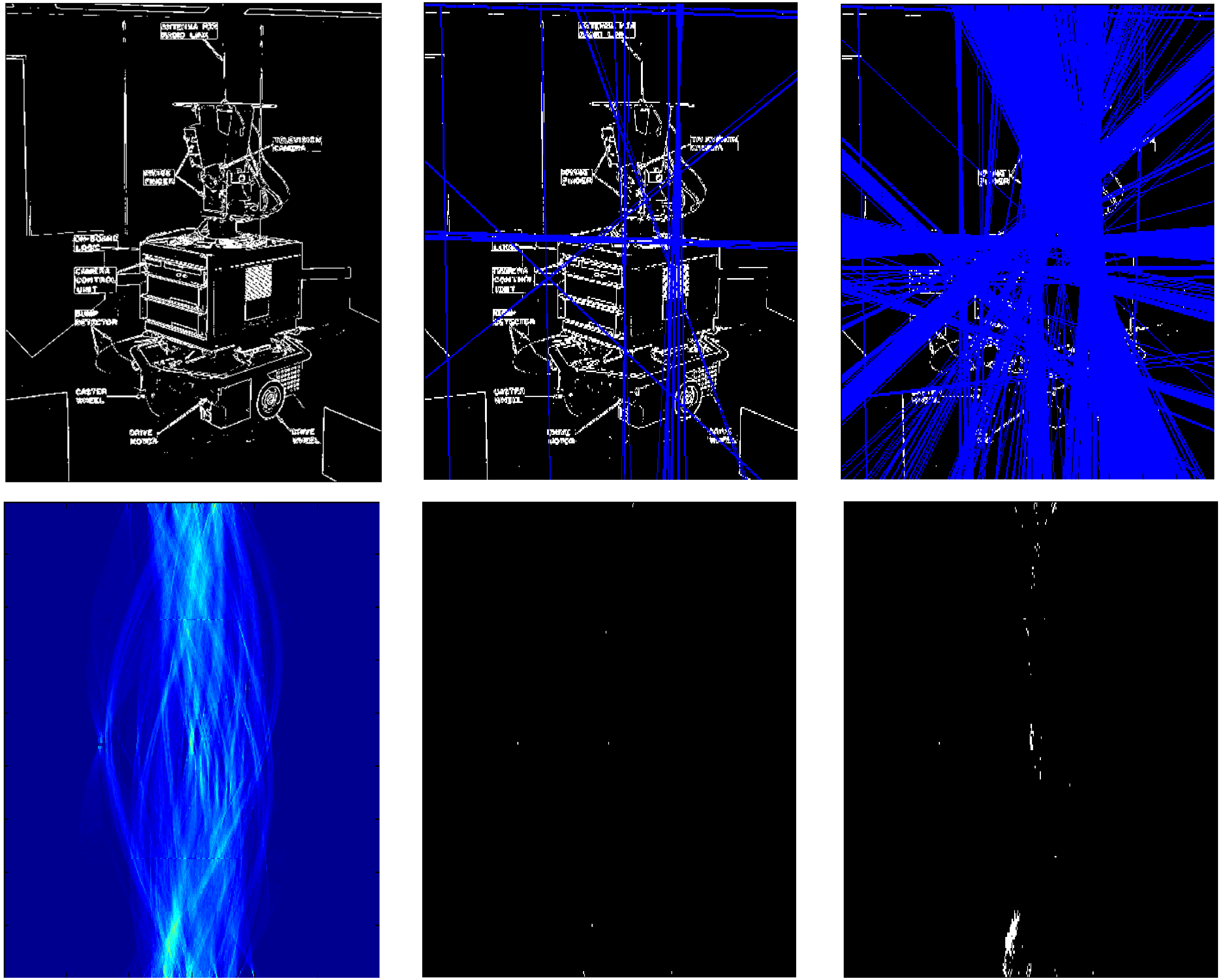
        H[d, θ] += 1

3. Find the value(s) of (d, θ) where H[d, θ] is maximum
4. The detected line in the image is given by

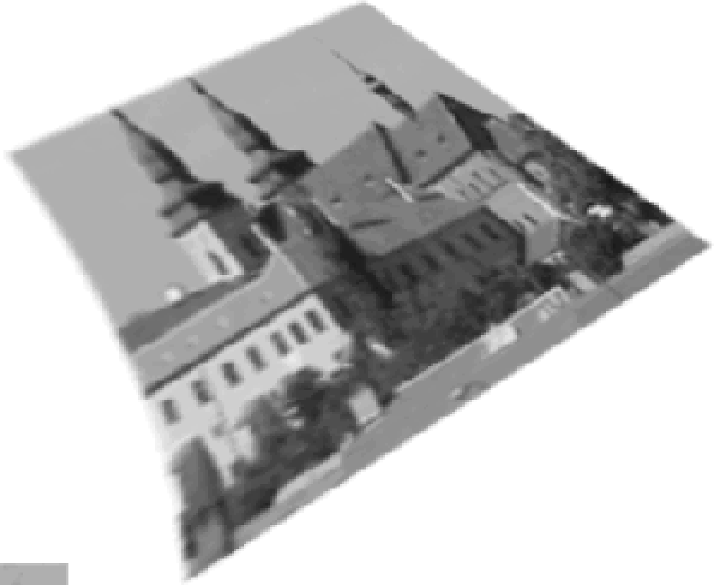
$$d = x cos\theta + y sin\theta$$

A simple example




TELEVISION CAMERA
CAMERA CONTROL UNIT
CASTER WHEEL
DRIVE MOTOR
DRIVE WHEEL

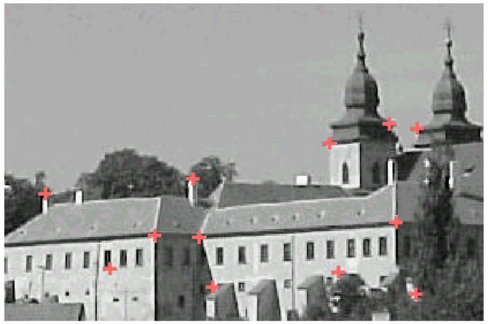
Registrace dat

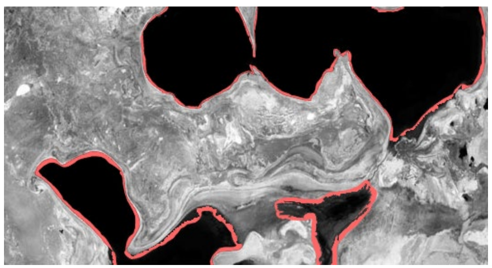


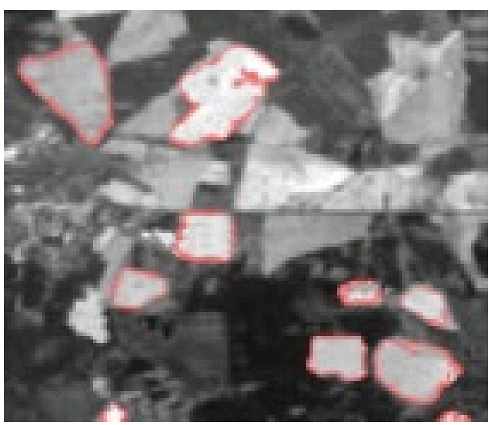
4 základní kroky registrace

1. Volba řídících bodů
2. Korespondence
3. Design mapovací funkce
4. Resampling a transformace

FEATURE DETECTION





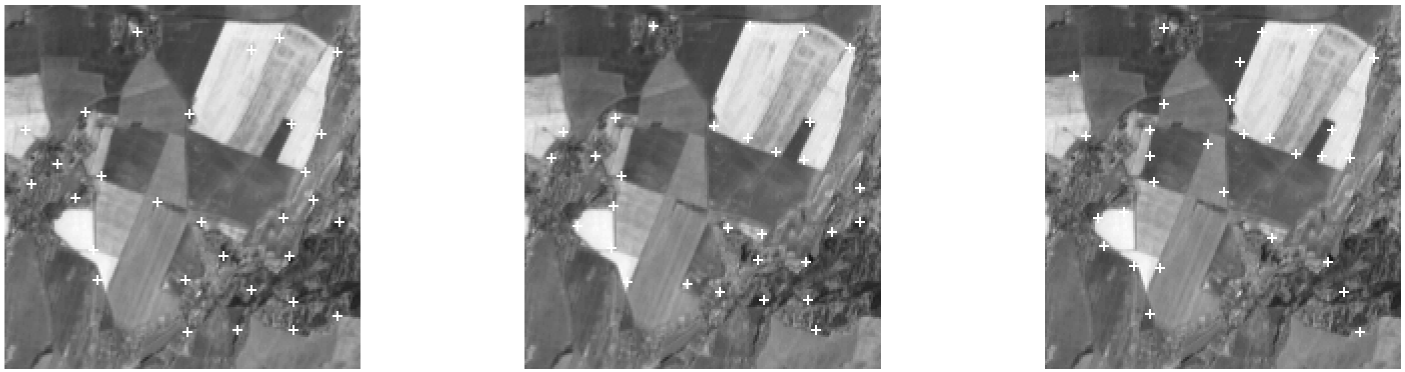


Area-based methods - windows

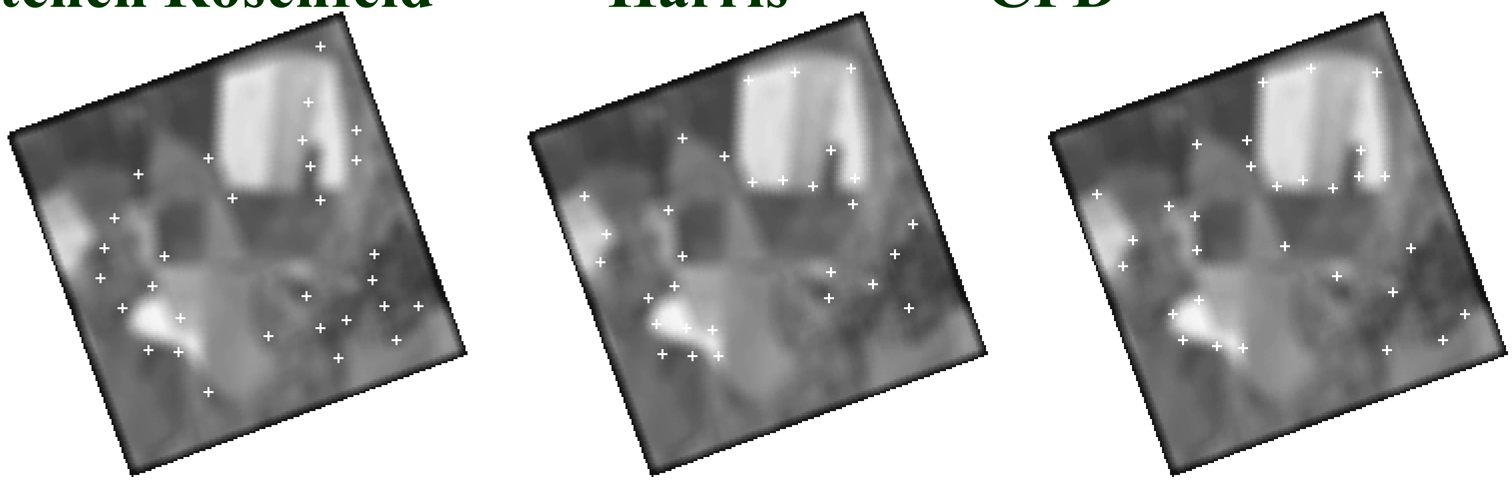
Feature-based methods (higher level info)

- distinctive points
- corners
- lines
- closed-boundary regions
- invariant regions

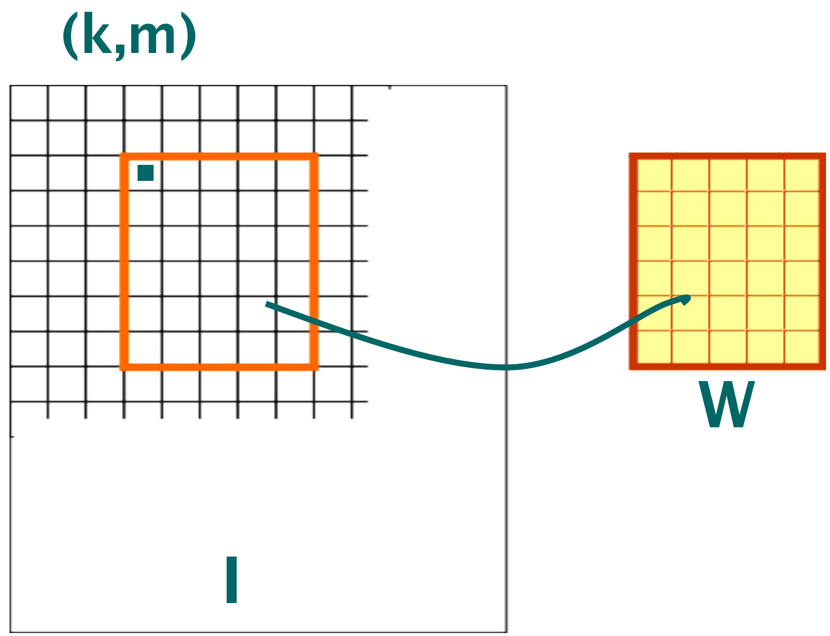
FEATURE DETECTION POINTS AND CORNERS



Kitchen Rosenfeld **Harris** **CPD**



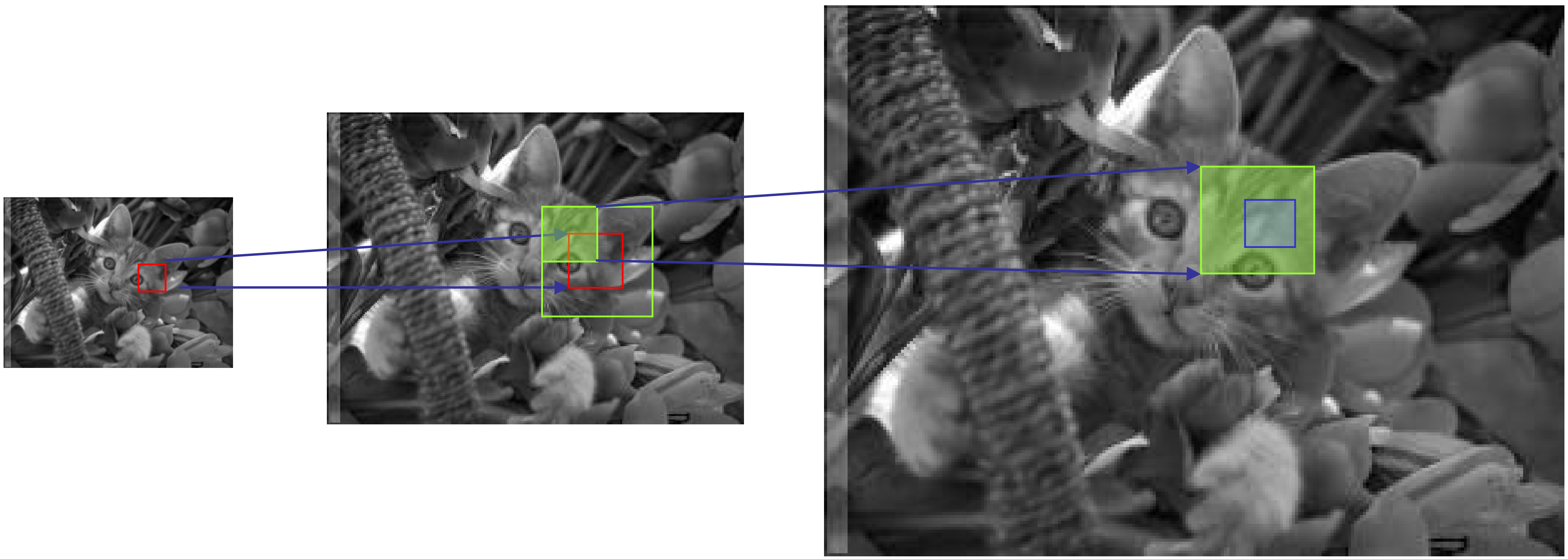
Kros-korelace a podobné metody



$$C(k,m) = \frac{\sum (I_{k,m} - \text{mean}(I_{k,m})) \cdot (W - \text{mean}(W))}{\sqrt{\sum (I_{k,m} - \text{mean}(I_{k,m}))^2} \cdot \sqrt{\sum (W - \text{mean}(W))^2}}$$

FEATURE MATCHING PYRAMIDAL REPRESENTATION

processing from coarse to fine level



wavelet transform

FEATURE MATCHING PHASE CORRELATION

Fourier shift theorem

if *f(x)* is shifted by *a* to *f(x-a)*

- FT magnitude stays constant
- phase is shifted by $-2\pi a\omega$

shift parameter – spectral comparison of images

FEATURE MATCHING PHASE CORRELATION

SPOMF symmetric phase - only matched filter

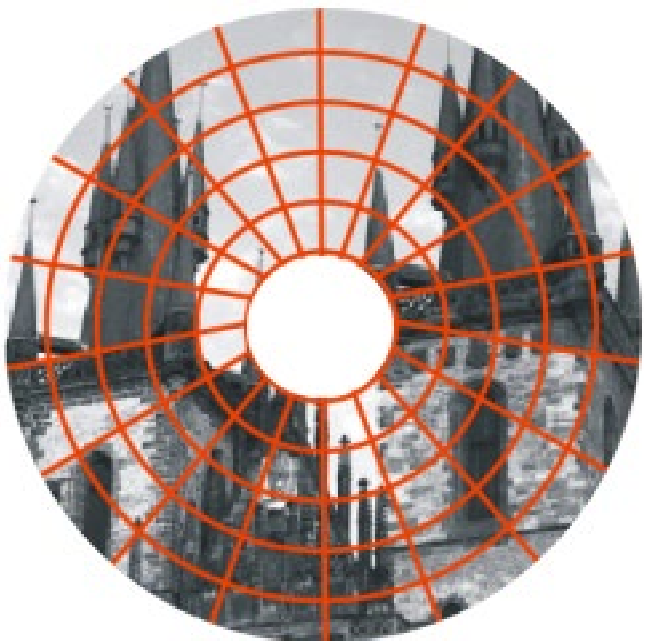
image f window w

$$\frac{W \,.\, F^{*}}{|W \,.\, F|} = e^{-2\pi i\,(\omega a \;+\; \xi b)}$$

$$\mathrm{IFT}\left(e^{-2\pi i\,(\omega a \;+\; \xi b)}\right) = \delta(x-a, y-b)$$

Log-polární transformace

log-polar transform



polar

$$r = [\,(x-x_c)^2 + (y-y_c)^2]^{1/2}$$

$$\theta = \tan^{-1}((y-y_c) / (x-x_c))$$

log

$$R = \frac{(n_r-1)\log(r/r_{min})}{\log(r_{max}/r_{min})}$$

$$W = n_w\theta / (2\pi)$$

RTS registration

$$\mathcal{F}_x[f(x-x_0)](k) = e^{-2\pi i k x_0} F(k).$$

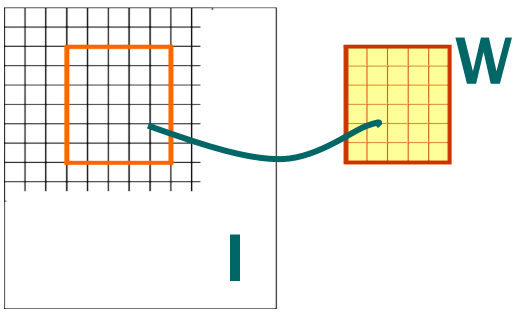
F(R(f)) = R(F(f))

$$\mathcal{F}_x[f(ax)](k) = |a|^{-1} F\left(\frac{k}{a}\right).$$

FT $\longrightarrow$ | | $\longrightarrow$ log-polar $\longrightarrow$ FT $\longrightarrow$ phase correlation

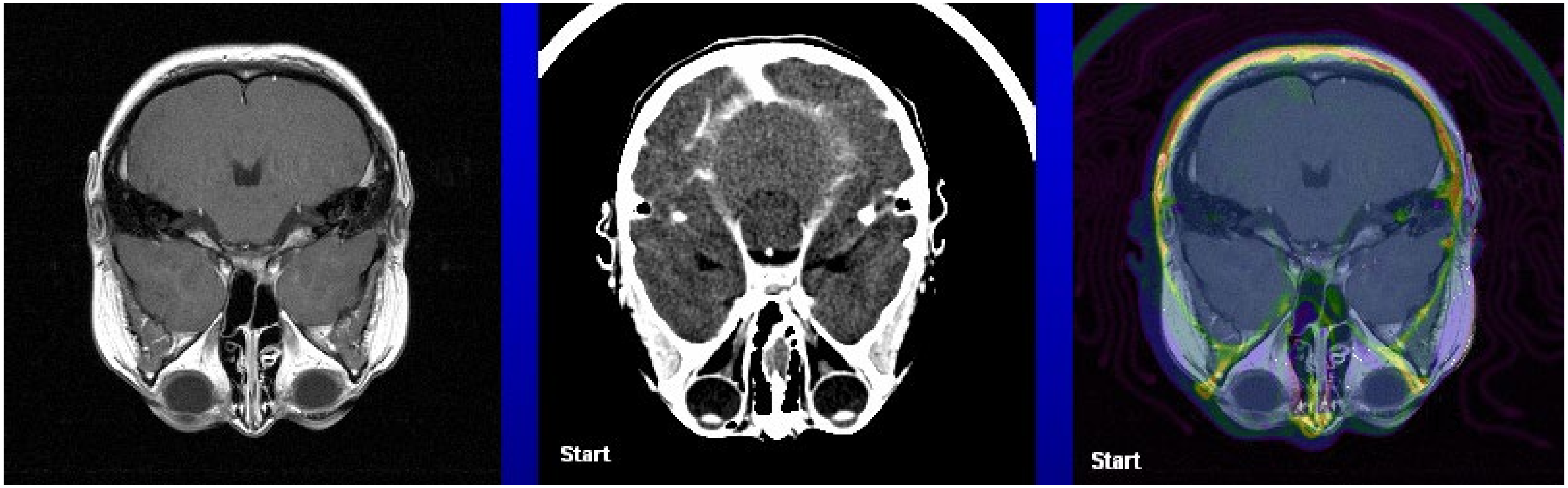
- π - periodicity of amplitude - > 2 angles
- log(abs(FT)+1)
- discrete problems

Mutual information method



Statistical measure of the dependence between two images

$$MI(f,g) = H(f) + H(g) - H(f,g)$$



MUTUAL INFORMATION

Entropy

$$H(X) = -\sum_{x} p(x)\log p(x)$$

Joint entropy

$$H(X,Y) = -\sum_{x}\sum_{y} p(x,y)\log p(x,y)$$

Mutual infomation

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

FEATURE MATCHING FEATURE-BASED METHODS

Combinatorial matching

no feature description, global information

graph matching
parameter clustering
ICP (3D)

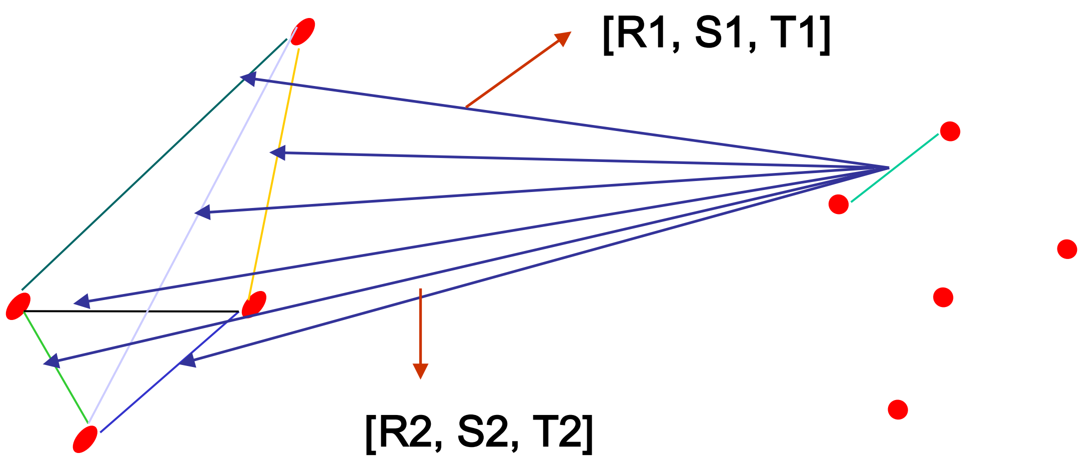
Matching in the feature space

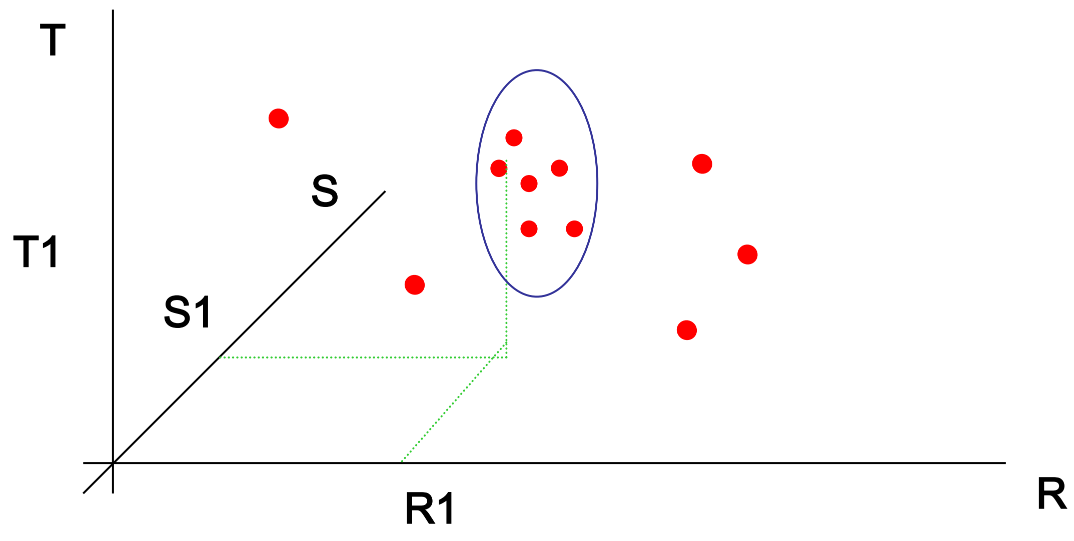
pattern classification, local information

invariance
feature descriptors

Hybrid matching combination, higher robustness

FEATURE MATCHING COMBINATORIAL - CLUSTER





FEATURE MATCHING FEATURE SPACE MATCHING

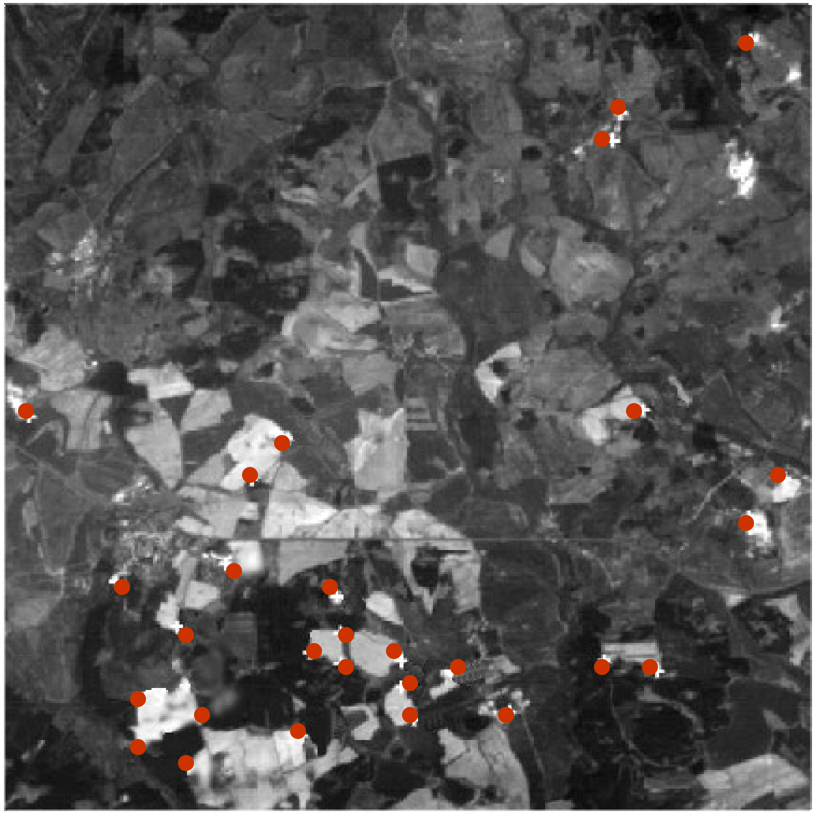
Detected features - points, lines, regions

Invariants description

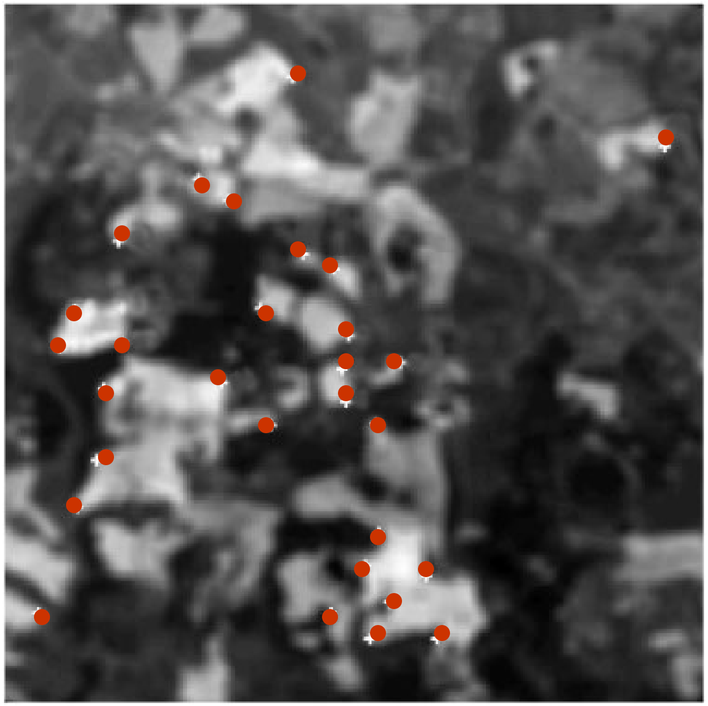
- intensity of close neighborhood
- geometrical descriptors (MBR, etc.)
- spatial distribution of other features
- angles of intersecting lines
- shape vectors
- moment invariants
- …

Combination of descriptors

FEATURE MATCHING



FEATURE SPACE MATCHING



$$\min_{k,m} \ \mathrm{distance}\big((v1_k, v2_k, v3_k, \ldots) , (\overline{v1}_m, \overline{v2}_m, \overline{v3}_m, \ldots) \big)$$

maximum likelihood coefficients

	W1	W2	W3	W4
V1	Dist			
V2				
V3				
V4				

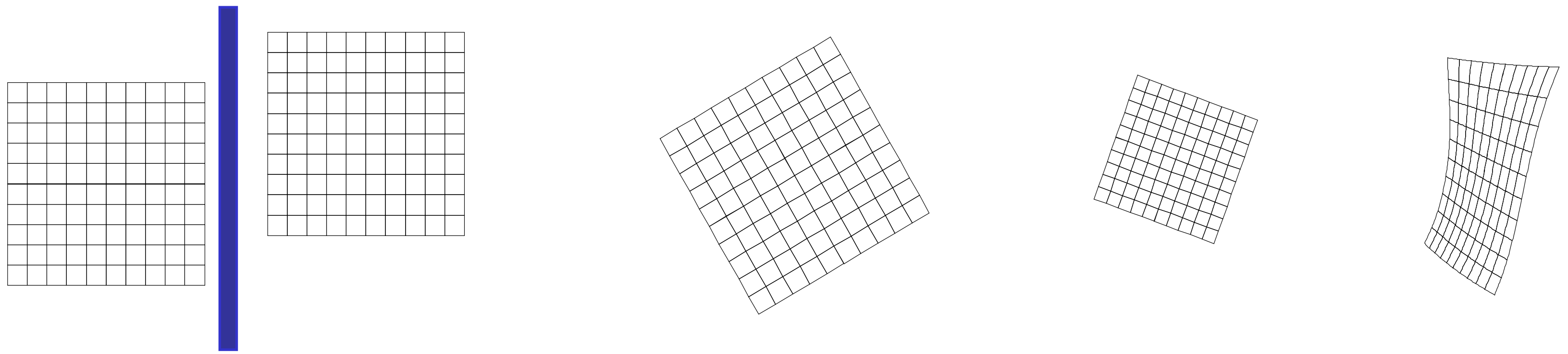
…

⋮

min (best / 2nd best)

TRANSFORM MODEL ESTIMATION

$$x' = f(x,y)$$
$$y' = g(x,y)$$



incorporation of *a priory* known information

removal of differences

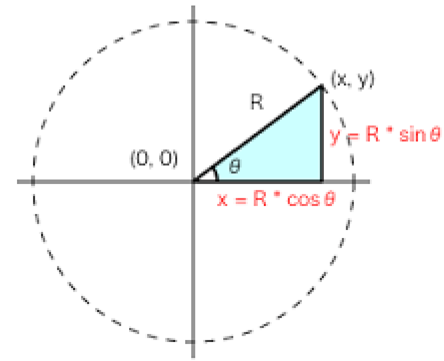
transformations represented with a 2x2 matrix

2D Scaling

$$x' = s_x * x$$

$$y' = s_y * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

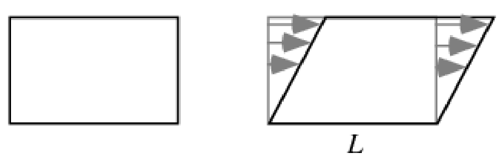


2D Rotate around (0,0)

$$x' = \cos\Theta * x - \sin\Theta * y$$
$$y' = \sin\Theta * x + \cos\Theta * y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta \\ \sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

2D Shear



$$x' = x + sh_x * y$$

$$y' = sh_y * x + y$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Source: Alyosha Efros

transformations represented with a 2x2 matrix

2D Mirror about Y axis

$$x' = -x$$
$$y' = y$$

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

2D Mirror over (0,0)

$$x' = -x$$
$$y' = -y$$

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

2D Translation?

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}_x$$
$$\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{t}_y$$

NO!

Source: Alyosha Efros

All 2D Linear Transformations

Properties of linear transformations:

Origin maps to origin

Lines map to lines

Parallel lines remain parallel

Ratios are preserved

Closed under composition

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} i & j \\ k & l \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Podobnostní (similarity) transformace

$$X = s\,x\cos(\theta) - s\,y\sin(\theta) + h$$

$$Y = s\,x\sin(\theta) + s\,y\cos(\theta) + k$$

Rigid body transform

$s=1$

Affine Transformations

Affine transformations are combinations of …

- Linear transformations, and
- Translations

Properties of affine transformations:

- Origin does not necessarily map to origin
- Lines map to lines
- Parallel lines remain parallel
- Ratios are preserved
- Closed under composition
- Models change of basis

$$x' = a_0 + a_1 x + a_2 y$$

$$y' = b_0 + b_1 x + b_2 y$$

Projective Transformations

Projective transformations …

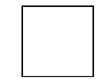
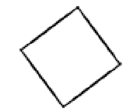
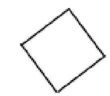
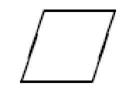
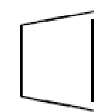
- Affine transformations, and
- Projective warps

Properties of projective transformations:

- Origin does not necessarily map to origin
- Lines map to lines
- Parallel lines do not necessarily remain parallel
- Ratios are not preserved
- Closed under composition
- Models change of basis

$$x' = (a_0 + a_1x + a_2y) / (1 + c_1x + c_2y)$$

$$y' = (b_0 + b_1x + b_2y) / (1 + c_1x + c_2y)$$

Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array} \right]_{2\times 3}$	2	orientation $+ \cdots$	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array} \right]_{2\times 3}$	3	lengths $+ \cdots$	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array} \right]_{2\times 3}$	4	angles $+ \cdots$	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array} \right]_{2\times 3}$	6	parallelism $+ \cdots$	
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array} \right]_{3\times 3}$	8	straight lines	

TRANSFORM MODEL ESTIMATION - SIMILARITY TRANSFORM

translation $[\Delta x, \Delta y]$ rotation φ uniform scaling s

$$x' = s\,(x * \cos\varphi - y * \sin\varphi) + \Delta x$$

$$y' = s\,(x * \sin\varphi + y * \cos\varphi) + \Delta y$$

$$s\cos\varphi = a, \quad s\sin\varphi = b$$

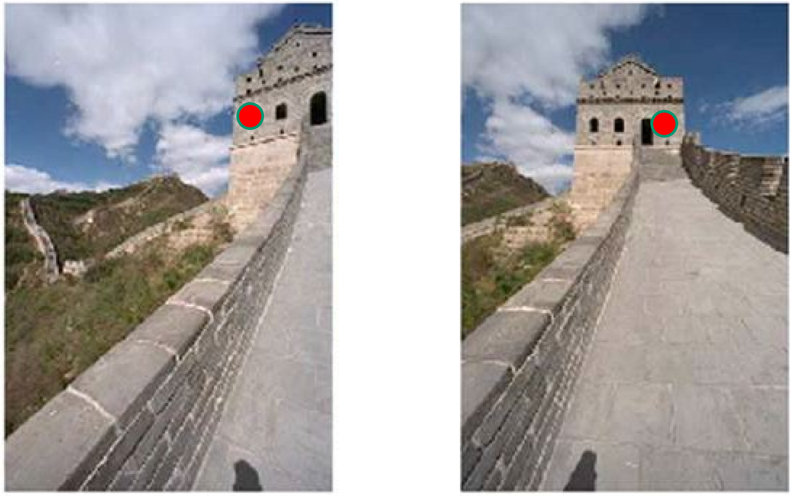
$$\min\left(\Sigma_{i=1}\{[x_i' - (ax_i - by_i) - \Delta x]^2 + [y_i' - (bx_i + ay_i) - \Delta y]^2\}\right)$$

$$\begin{vmatrix} \Sigma(x_i^2 + y_i^2) & 0 & \Sigma x_i & \Sigma y_i \\ 0 & \Sigma(x_i^2 + y_i^2) & \Sigma y_i & \Sigma x_i \\ \Sigma x_i & -\Sigma y_i & N & 0 \\ \Sigma y_i & \Sigma x_i & 0 & N \end{vmatrix} \bullet \begin{vmatrix} a \\ b \\ \Delta x \\ \Delta y \end{vmatrix} = \begin{vmatrix} \Sigma(x_i' x_i - y_i' y_i) \\ \Sigma(y_i' x_i - x_i' y_i) \\ \Sigma x_i' \\ \Sigma y_i' \end{vmatrix}$$

Outliers

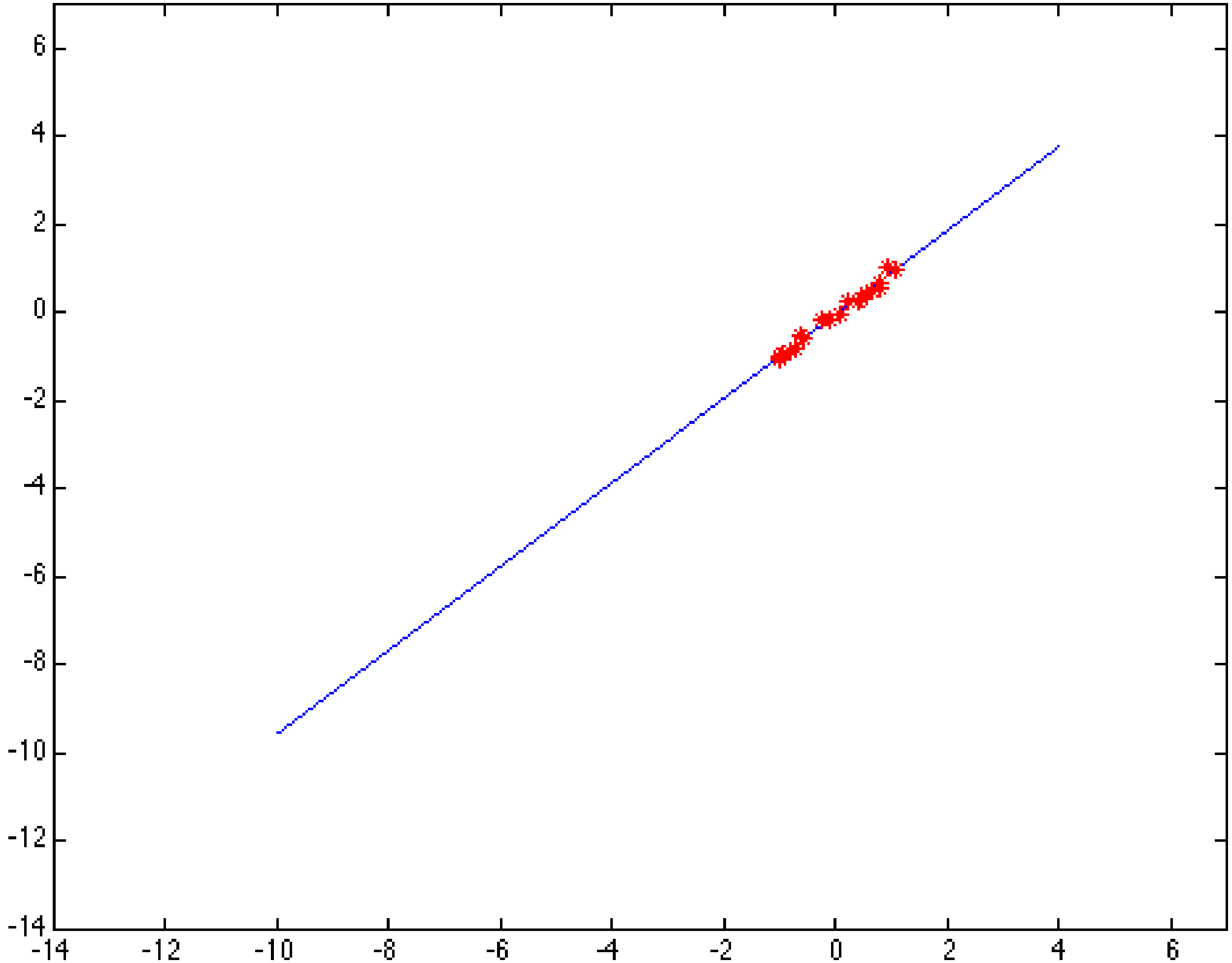
Outliers can hurt the quality of parameter estimates

an erroneous pair of matching points from two images

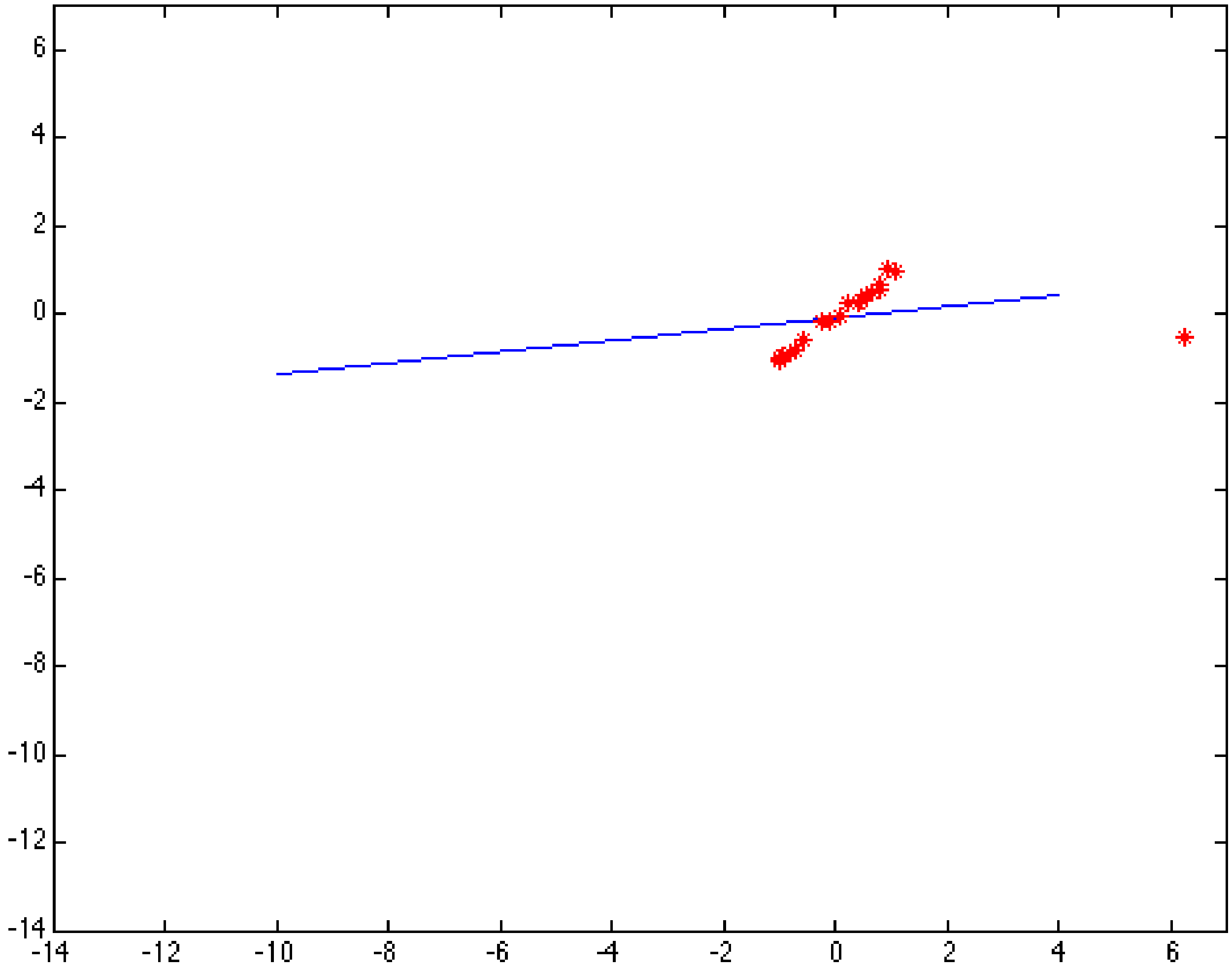


Kristen Grauman

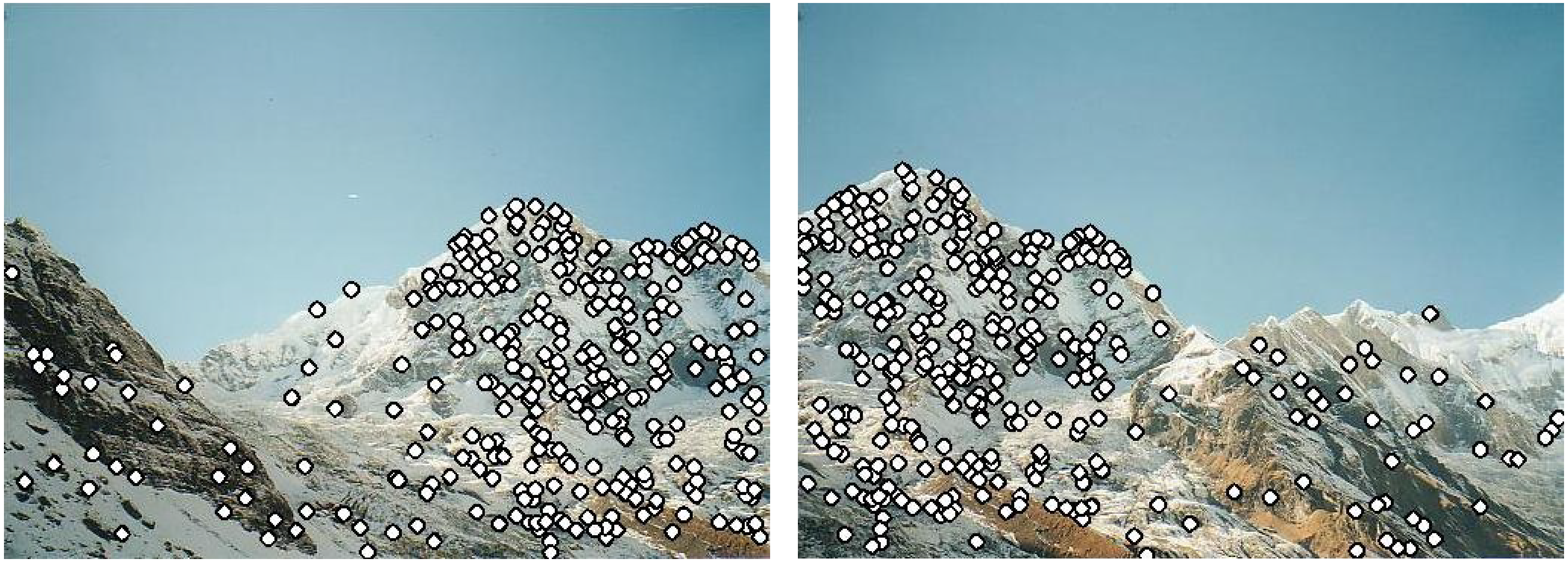
Outliers affect least squares fit



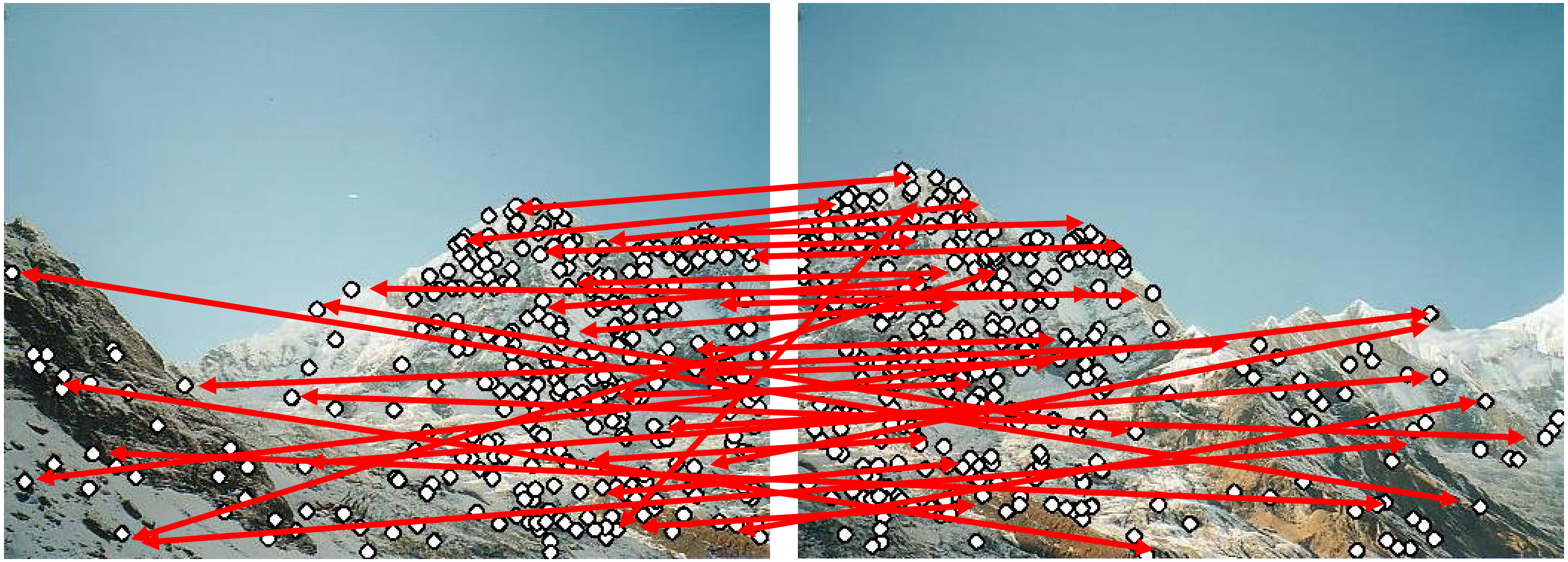
Outliers affect least squares fit



Feature-based alignment outline

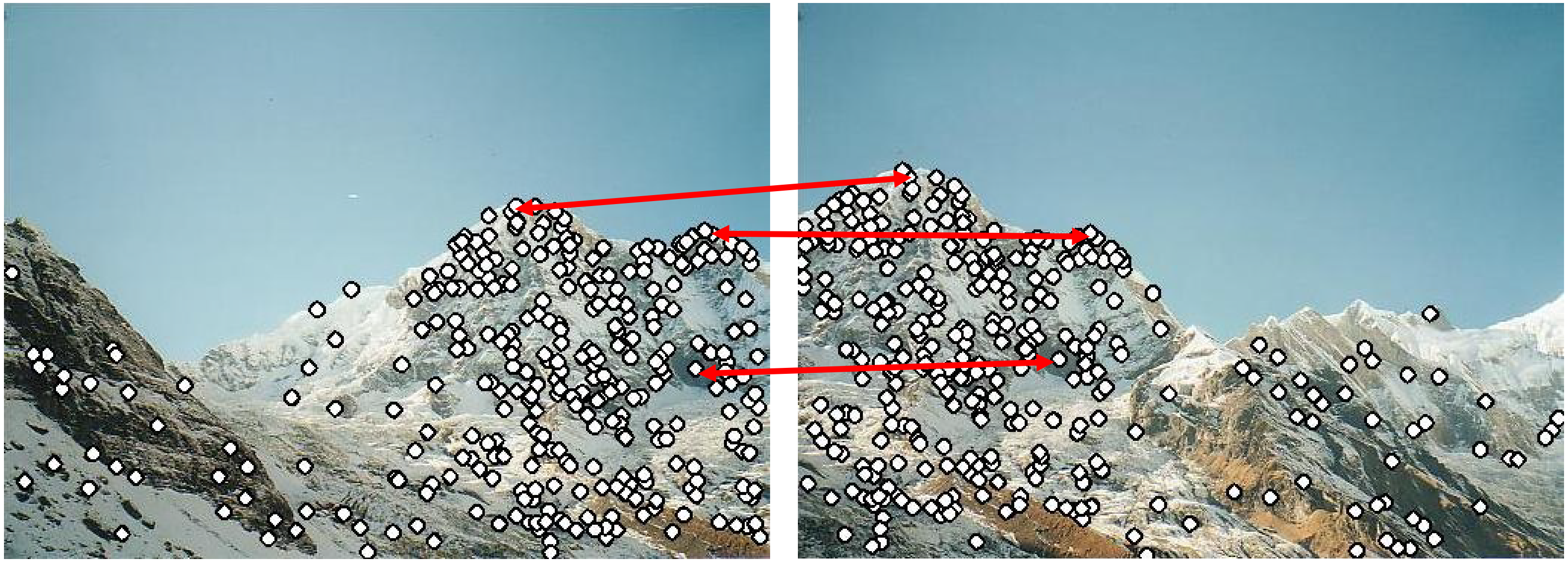


Extract features



Extract features

Compute *putative matches*

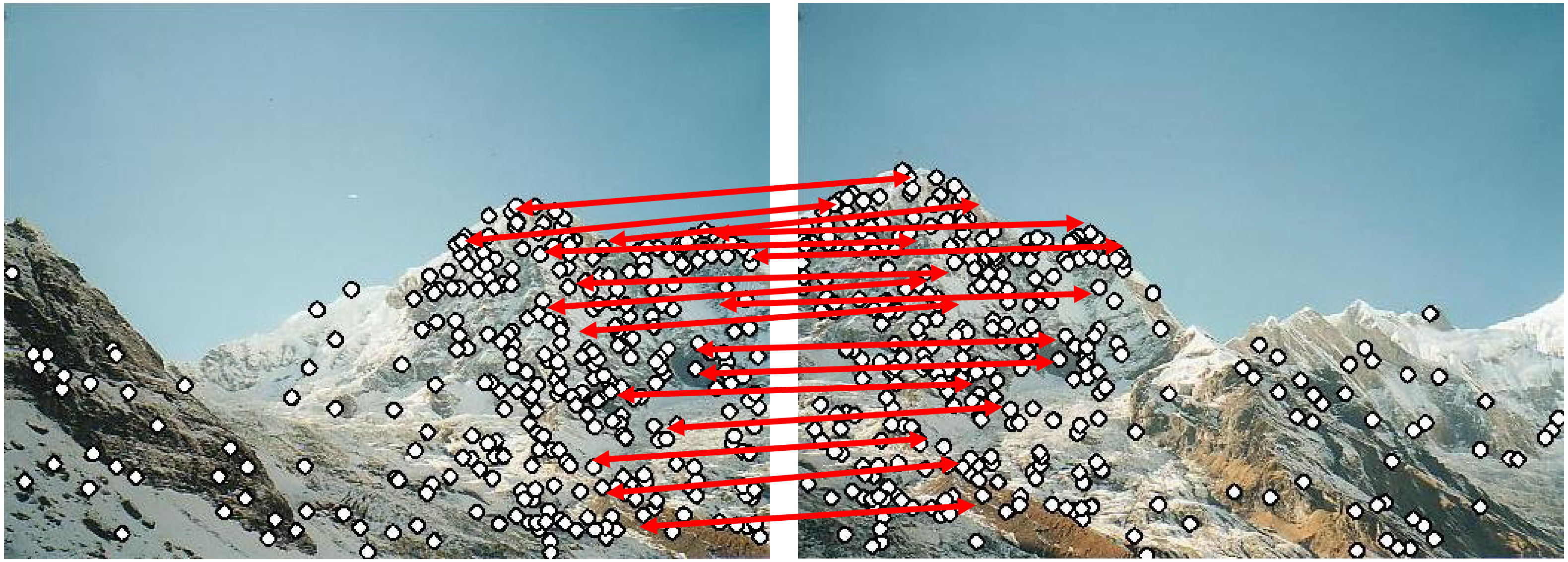


Extract features

Compute *putative matches*

Loop:

Hypothesize transformation T



Extract features

Compute *putative matches*

Loop:

- *Hypothesize* transformation T
- *Verify* transformation (search for other matches consistent with T)

Extract features

Compute *putative matches*

Loop:

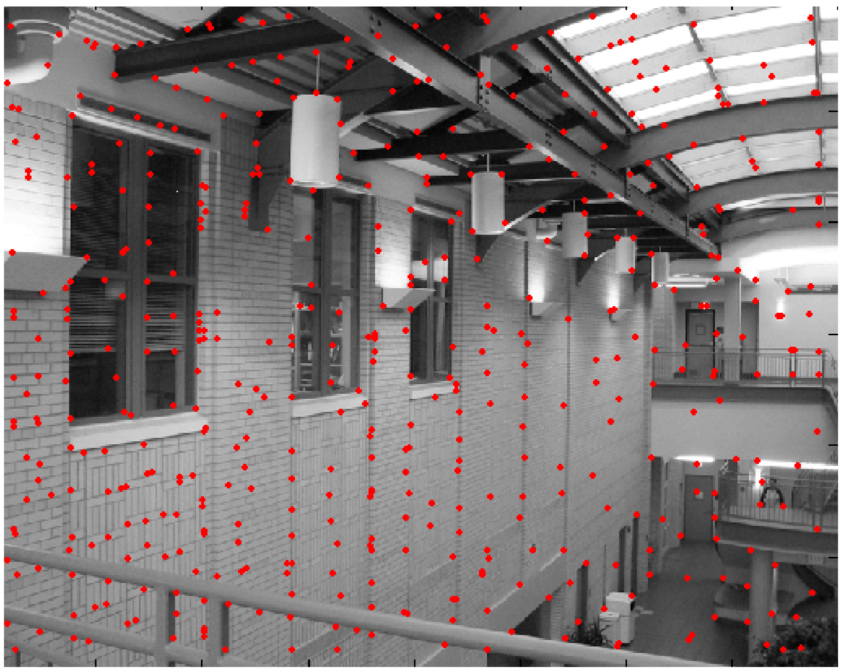
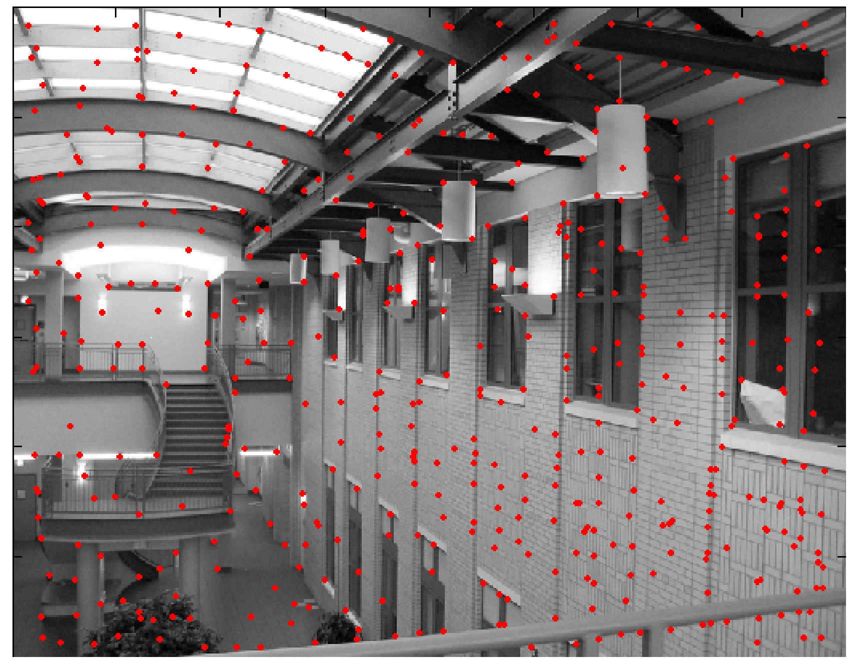
- *Hypothesize* transformation T
- *Verify* transformation (search for other matches consistent with T)

Iterative Closest Points (ICP) Algorithm

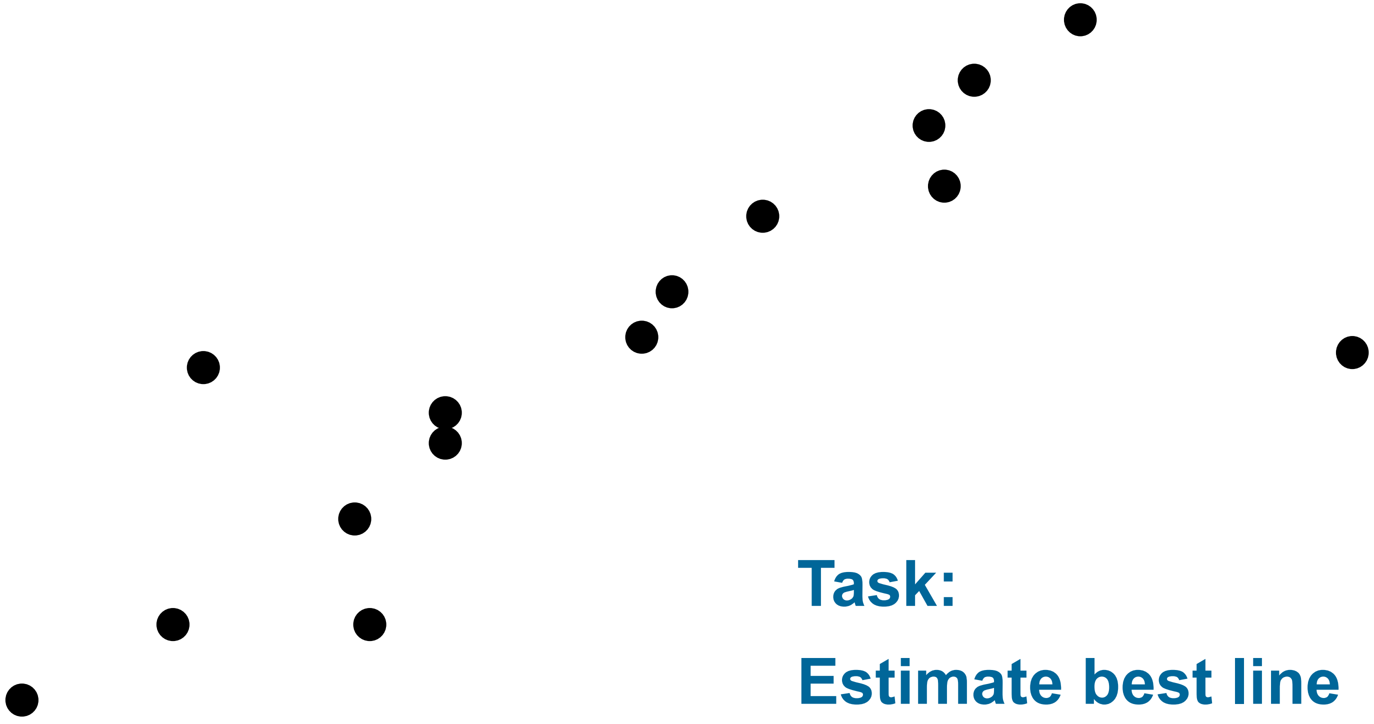
Goal: estimate transform between two dense sets of points

1. Assign each point in {Set 1} to its nearest neighbor in {Set 2}
2. Estimate transformation parameters
   e.g., least squares or robust least squares
3. Transform the points in {Set 1} using estimated parameters
4. Repeat steps 1-3 until change is very small

Slide from Derek Hoiem

RANSAC
random sample consensus



Task:

Estimate best line

RANSAC

RANdom Sample Consensus

Approach: we want to avoid the impact of outliers, so let's look for "inliers", and use those only.

Intuition: if an outlier is chosen to compute the current fit, then the resulting line won't have much support from rest of the points.

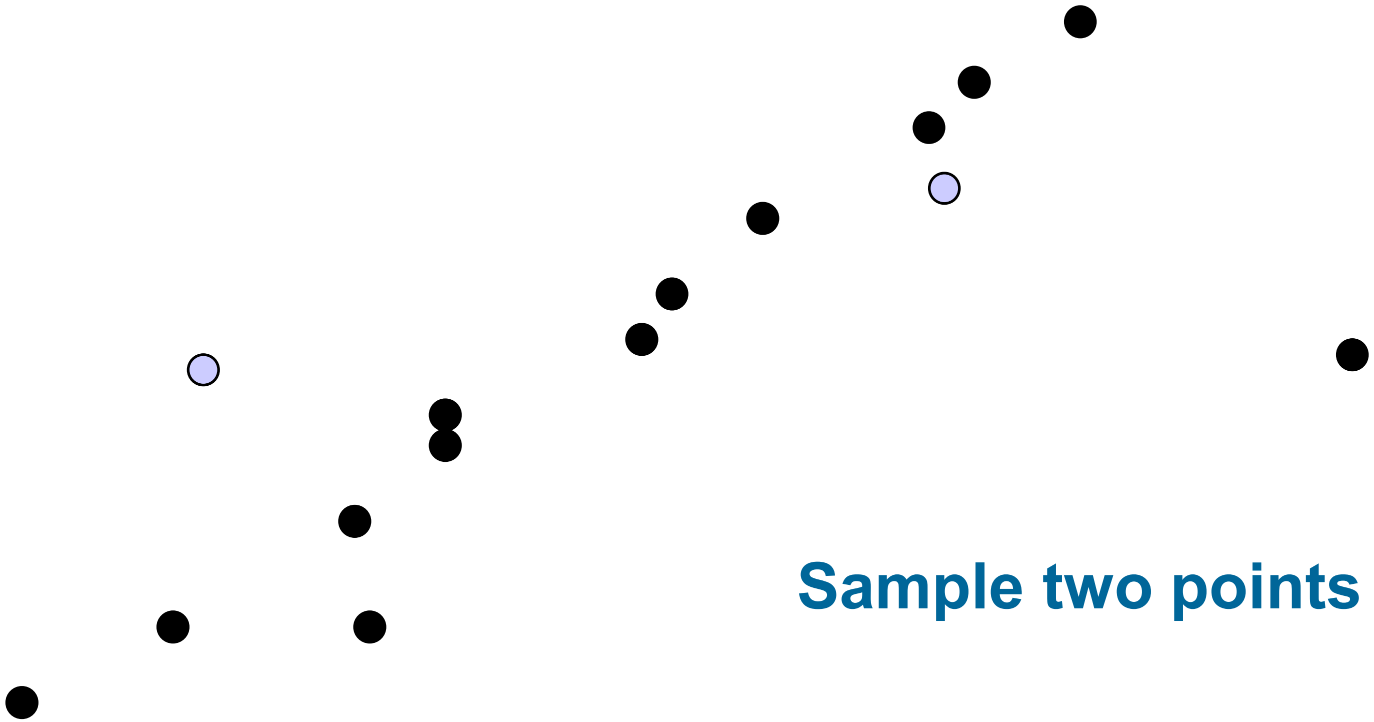
RANSAC: General form

RANSAC loop:

1. **Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)**
2. **Compute transformation from seed group**
3. **Find *inliers* to this transformation**
4. **If the number of inliers is sufficiently large, re-compute estimate of transformation on all of the inliers**

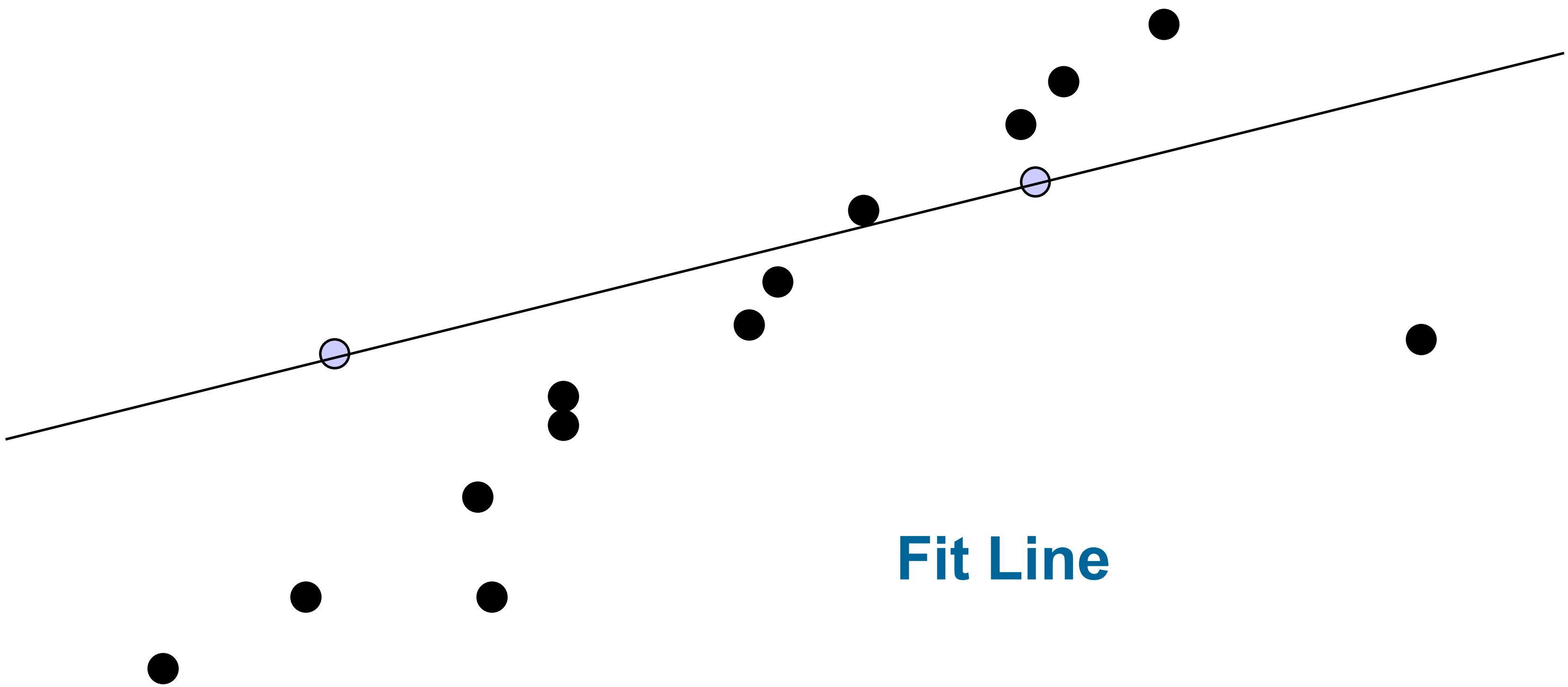
Keep the transformation with the largest number of inliers

RANSAC line fitting example

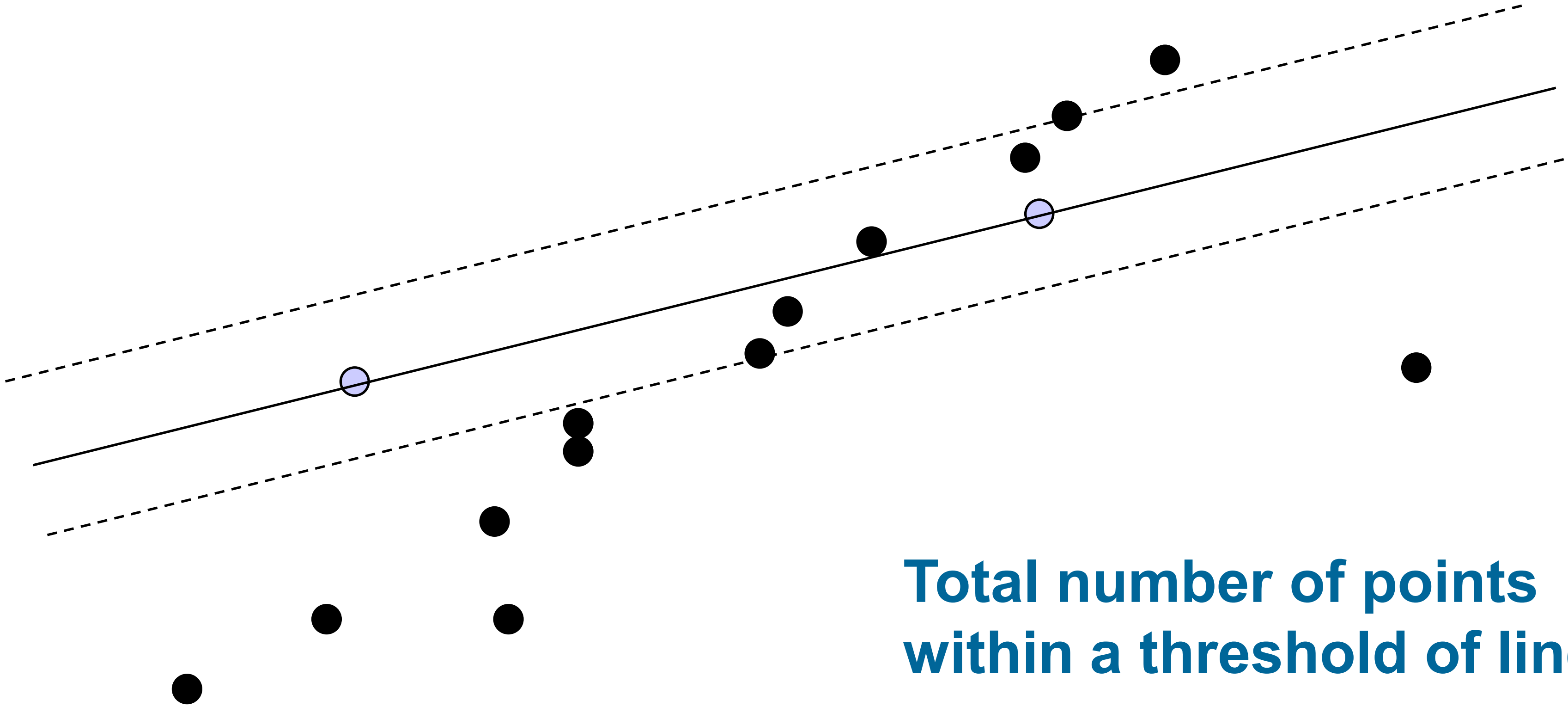


Sample two points

RANSAC line fitting example

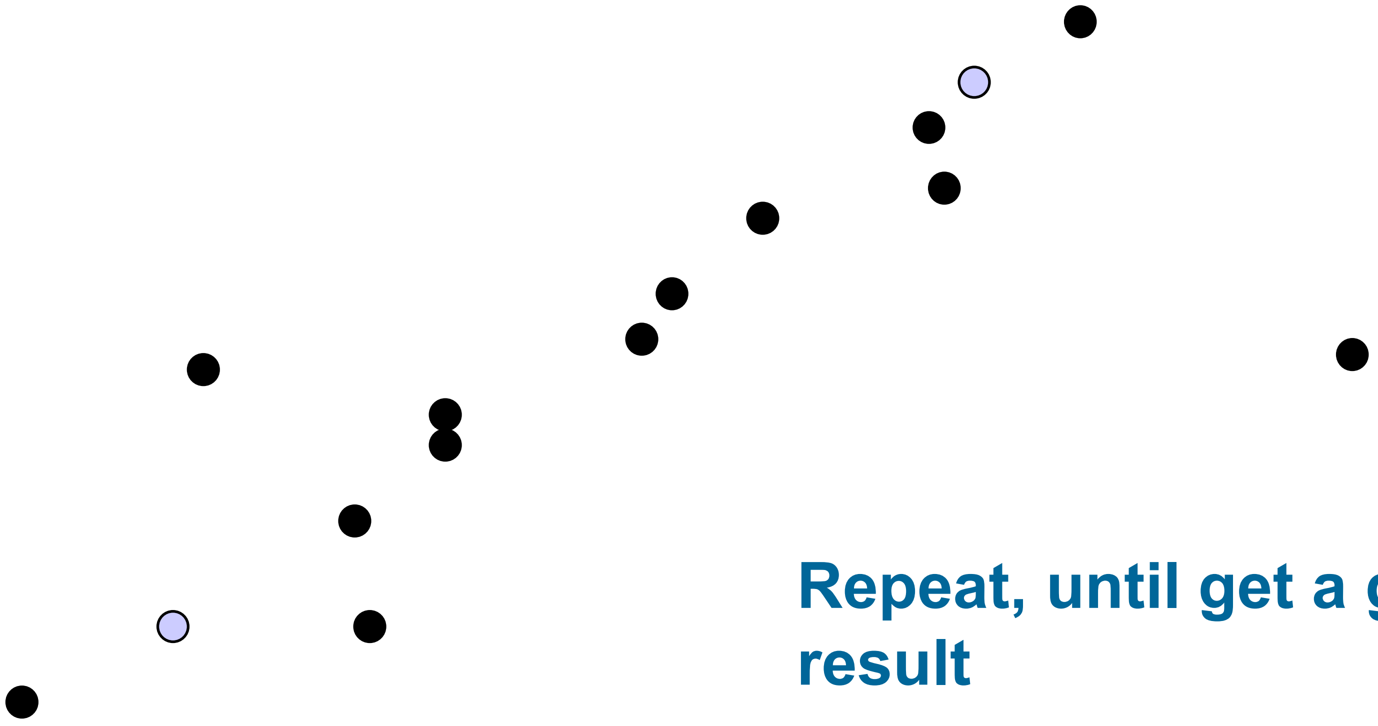


RANSAC line fitting example



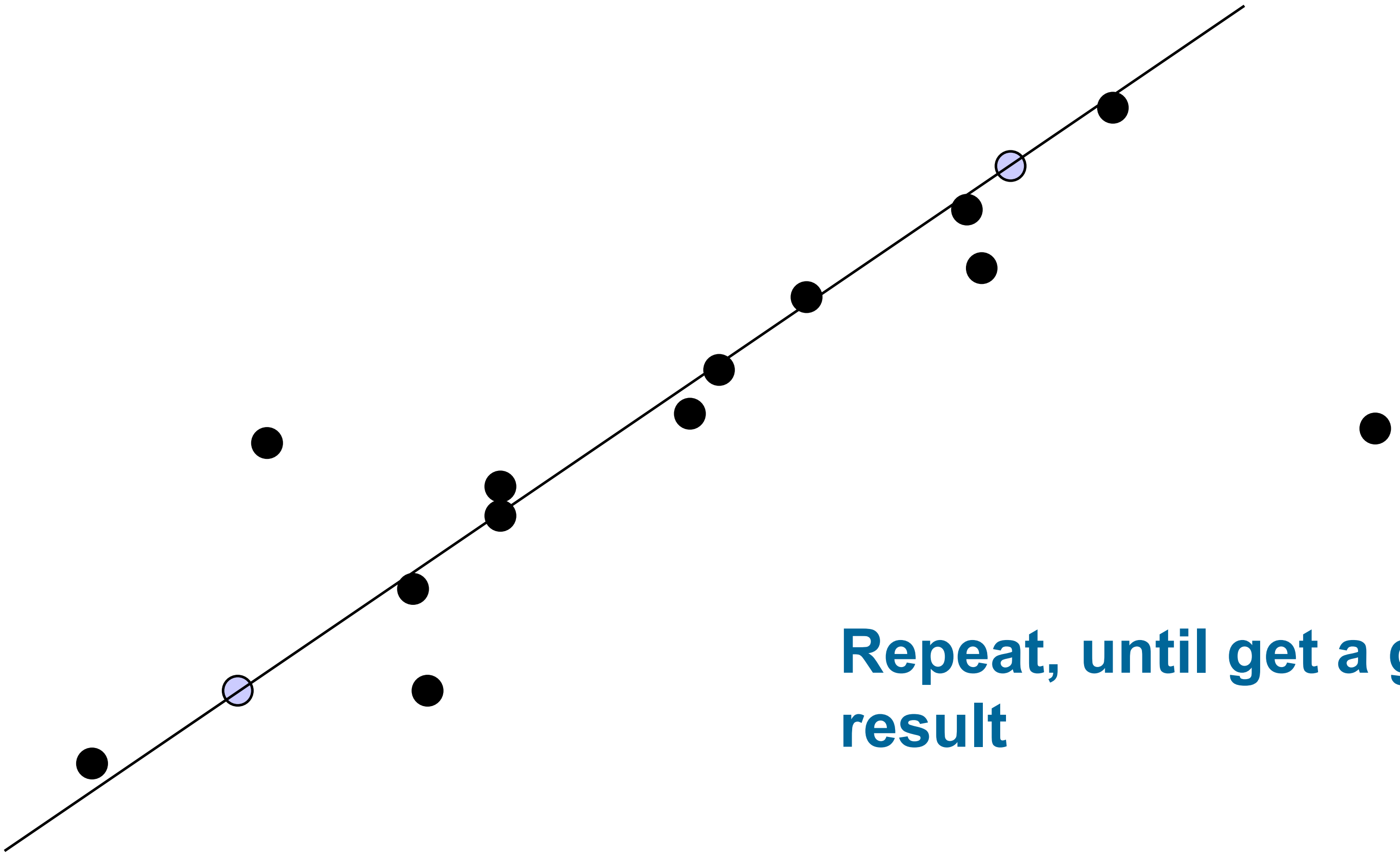
Total number of points within a threshold of line.

RANSAC line fitting example



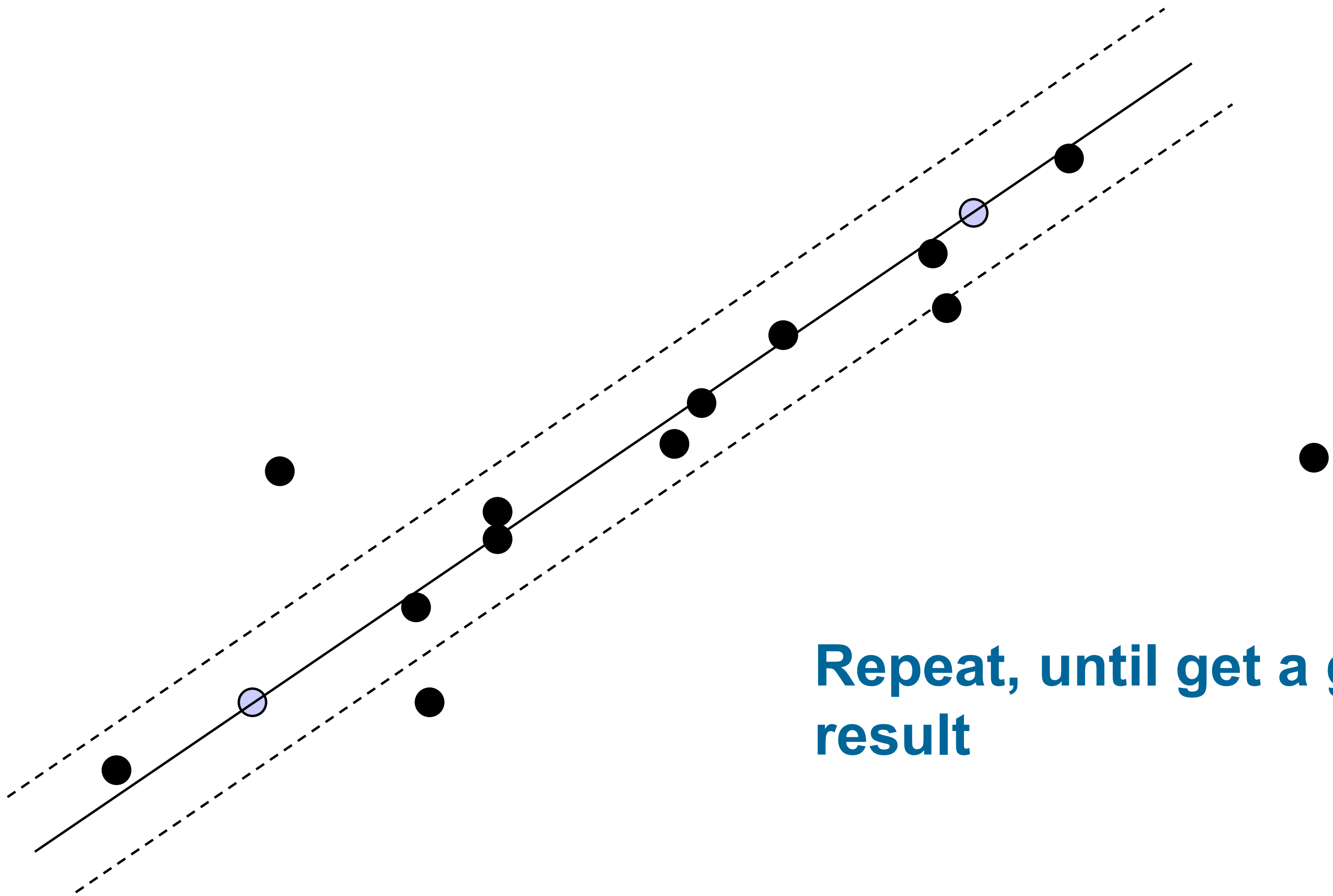
Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

RANSAC for line fitting

Repeat *N* times:

Draw *s* points uniformly at random

Fit line to these *s* points

Find inliers to this line among the remaining points (i.e., points whose distance from the line is less than *t*)

If there are *d* or more inliers, accept the line and refit using all inliers

Lana Lazebnik

RANSAC algorithm

Run *k* times:

(1) draw *n* samples randomly

(2) fit parameters Θ with these *n* samples

(3) for each of other *N*-*n* points, calculate their distance to the fitted model, count the number of inlier points, *c*

Output Θ with the largest *c*

How many times?

How big?
Smaller is better

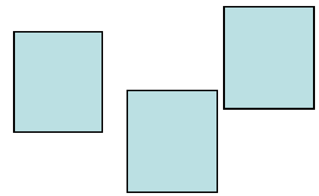
How to define?
Depends on the problem.

How to determine n ?

- <u>Minimum</u> n value depends on Model.

Translation
2dof

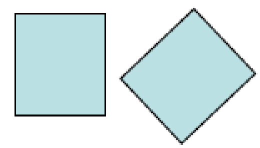
$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



n=1

Rigid
3dof

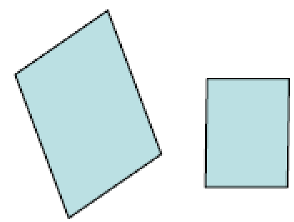
$$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



n=2

Affine
6dof

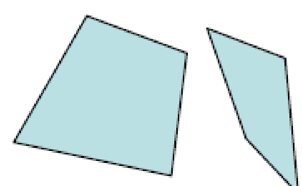
$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



n=3

Projective
8dof

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



n=4

How to determine k

- *p*: probability of real inliers
- *P*: probability of success after k trials

$$P = 1 - (1 - p^n)^k$$

p^n: n samples are all inliers

$1 - p^n$: a failure

$(1 - p^n)^k$: failure after k trials

$$k = \frac{\log(1-P)}{\log(1-p^n)}$$

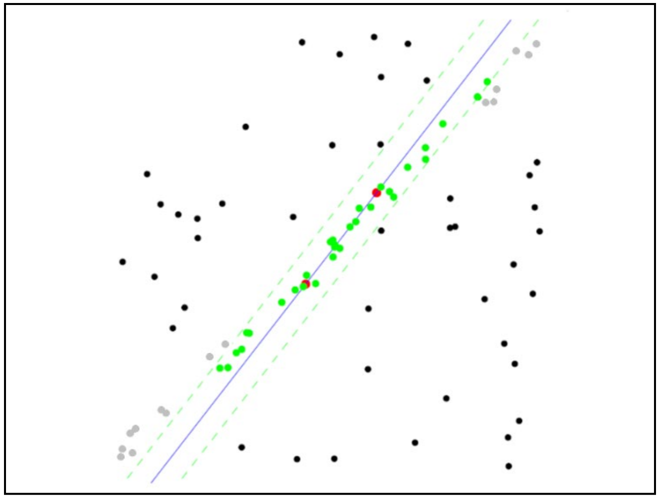
for $P=0.99$

n	*p*	*k*
3	0.5	35
6	0.6	97
6	0.5	293

RANSAC pros and cons

Pros

- Simple and general
- Applicable to many different problems
- Often works well in practice

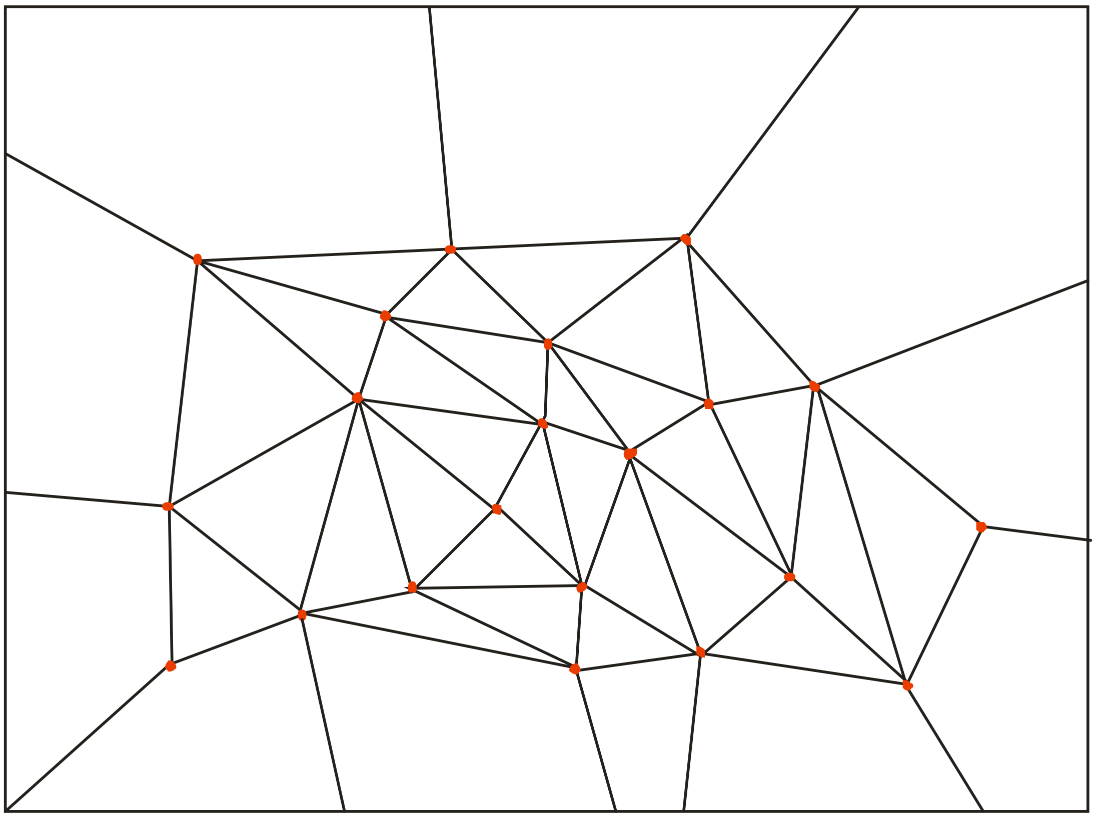


Cons

- Lots of parameters to tune
- Doesn't work well for low inlier ratios (too many iterations, or can fail completely)
- Can't always get a good initialization of the model based on the minimum number of samples

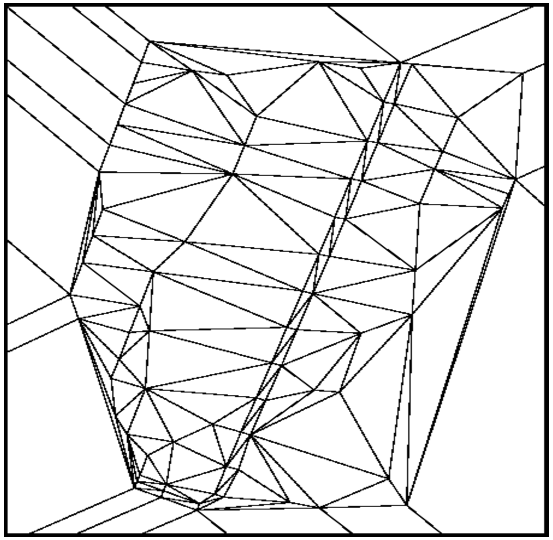
Lana Lazebnik

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM



Local mapping functions

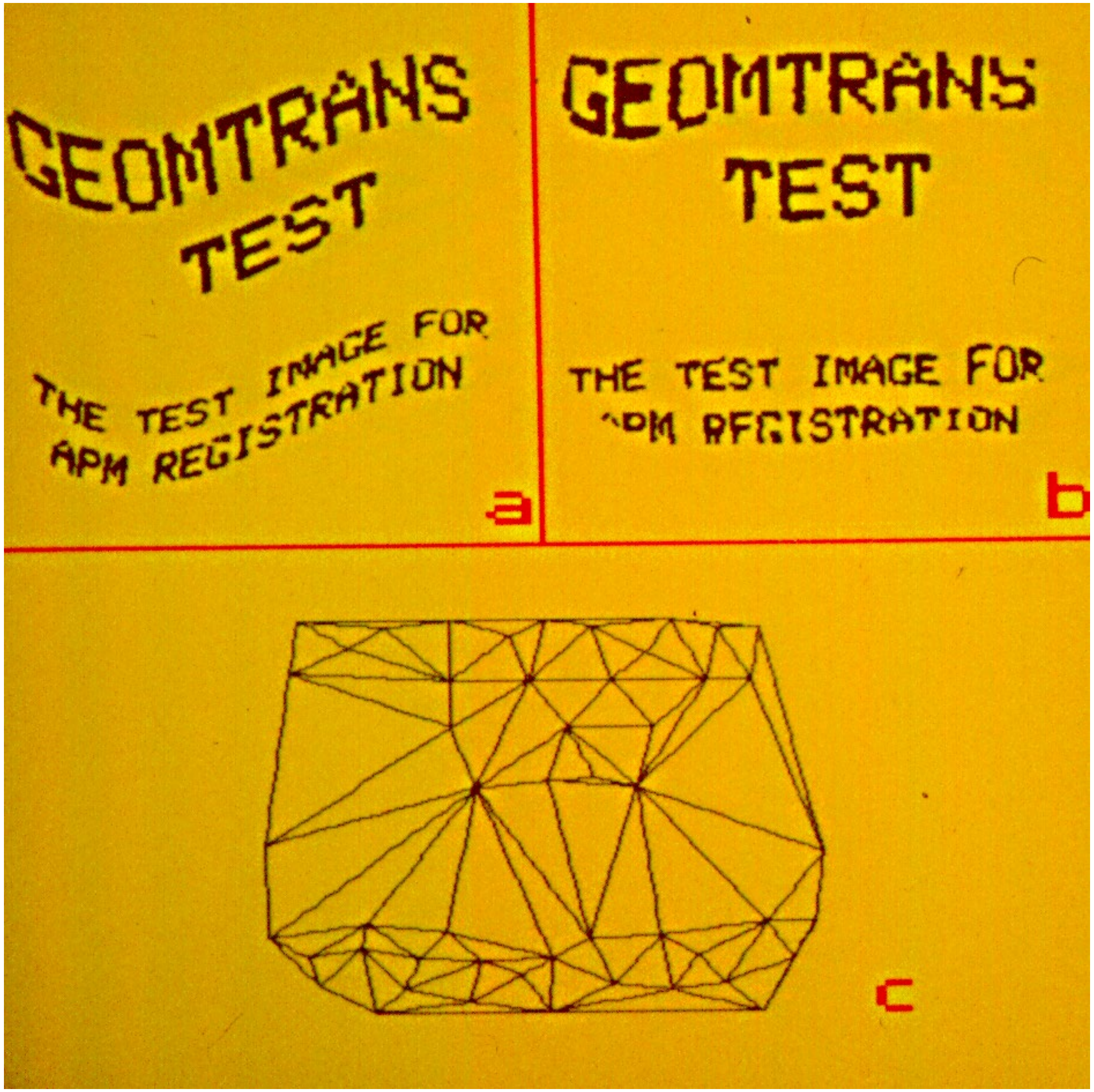
- **Piecewise affine or cubic**
- **Thin-Plate Splines (TPS)**



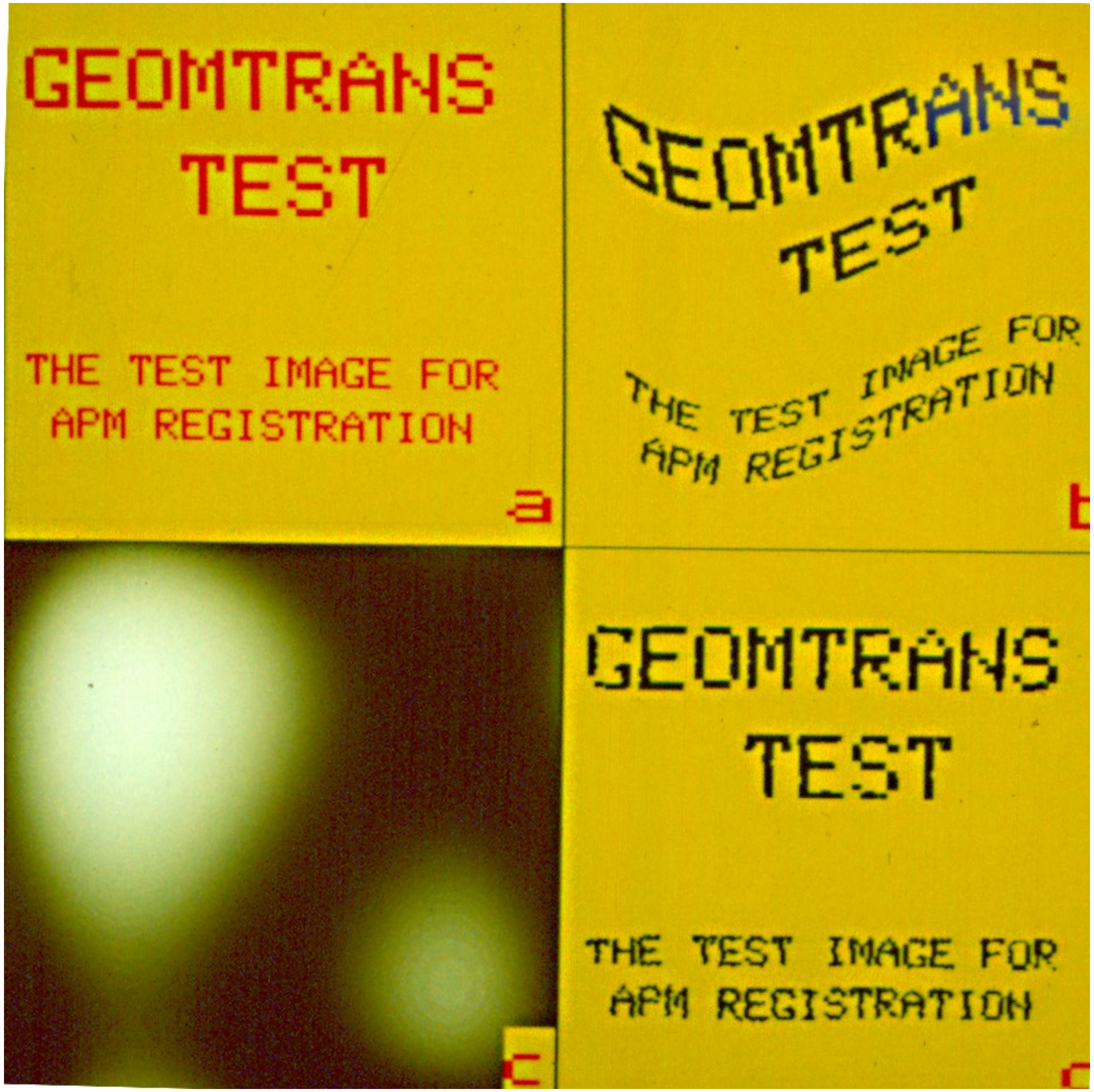
$$\alpha_1 + \alpha_2 x + \alpha_3 y + \sum_{i=1}^{N} a_i g_i(\|x - x_i, y - y_i\|),$$

$$g_i(t) = t^2 \log t.$$

Piecewise affine mapping

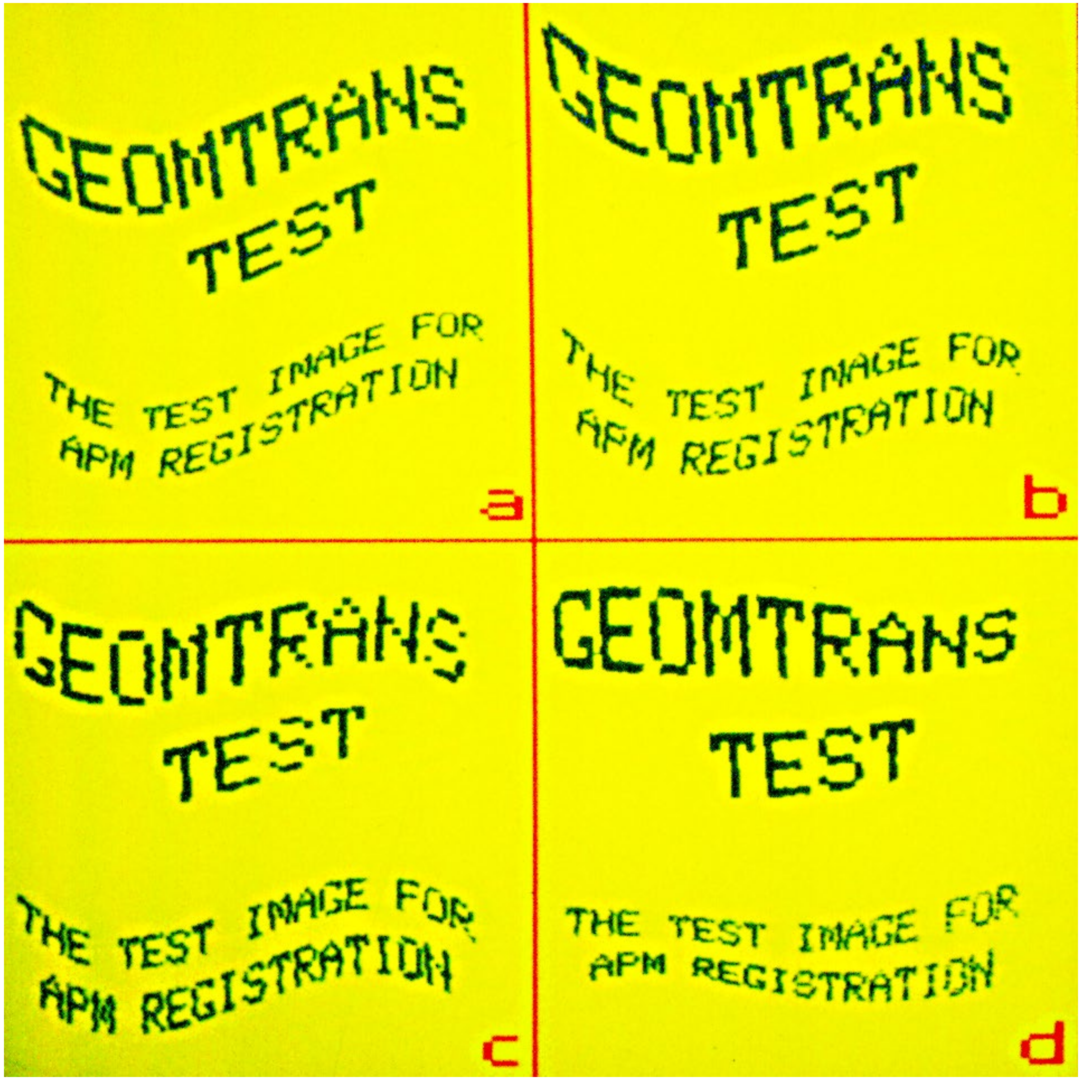


Mapping functions – a comparison



TPS

Mapping functions – a comparison

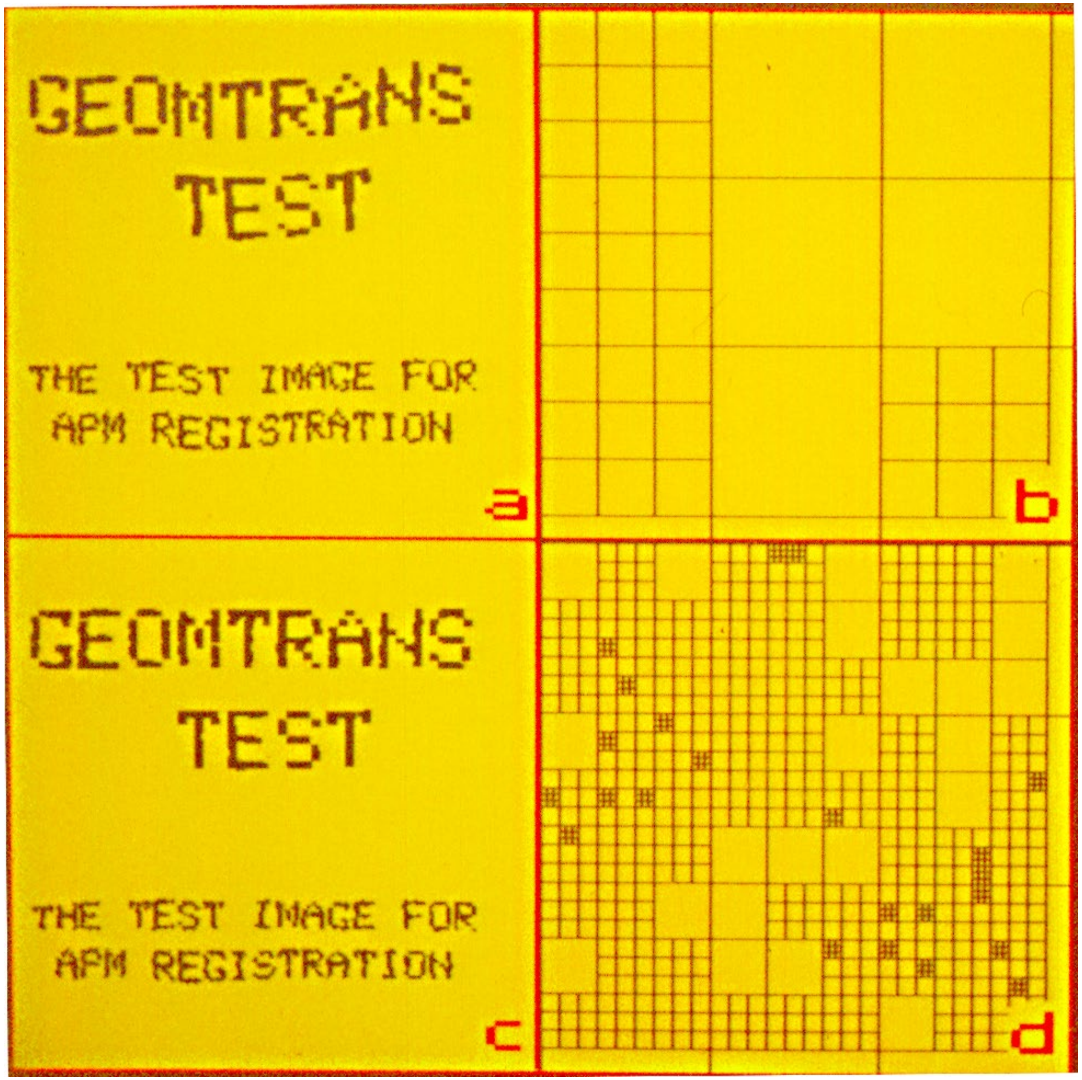


Affine

Quadratic

Cubic

Mapping functions – a comparison

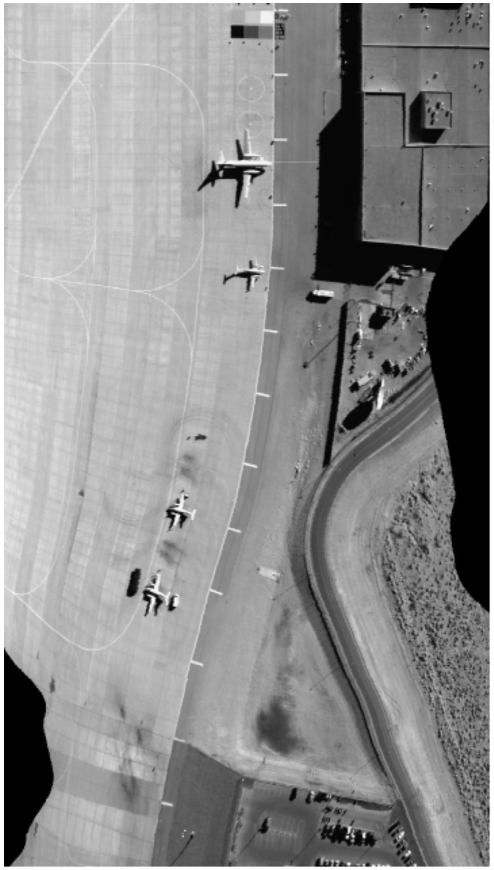


Piecewise projective

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM

reference

sensed (simulation)



From D. N. Fogel et al., UCSB

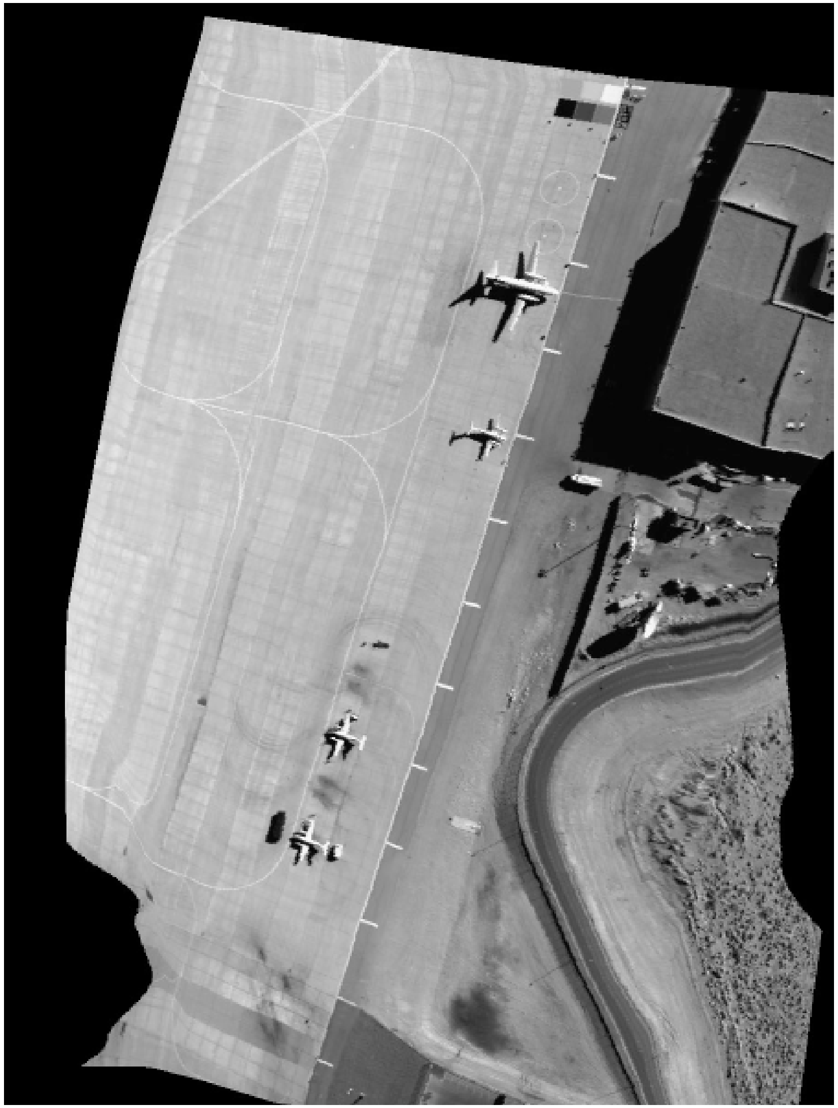
TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM

affine mapping

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM

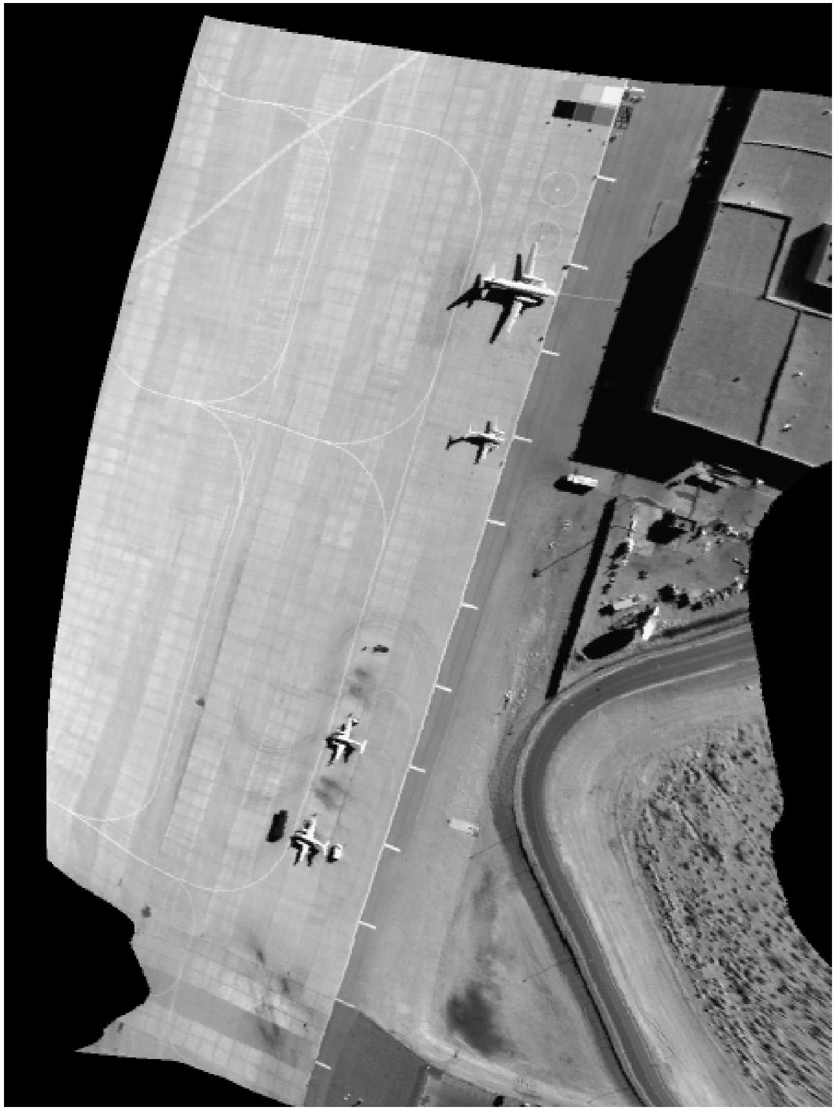
cubic mapping

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM



piecewise affine

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM



multiquadrics

TRANSFORM MODEL ESTIMATION - PIECEWISE TRANSFORM

TPS

TRANSFORM MODEL ESTIMATION UNIFIED APPROACH

Pure interpolation – ill posed

Regularized approximation – well posed

$$\min \; J(f) = a\,E(f) + b\,R(f)$$

$E(f)$	error term
$R(f)$	regularization term
a,b	weights

TRANSFORM MODEL ESTIMATION UNIFIED APPROACH

Choices for **min** $J(f) = a\,E(f) + b\,R(f)$

$E(f) = \Sigma\,(x_i' - f(x_i, y_i))^2$

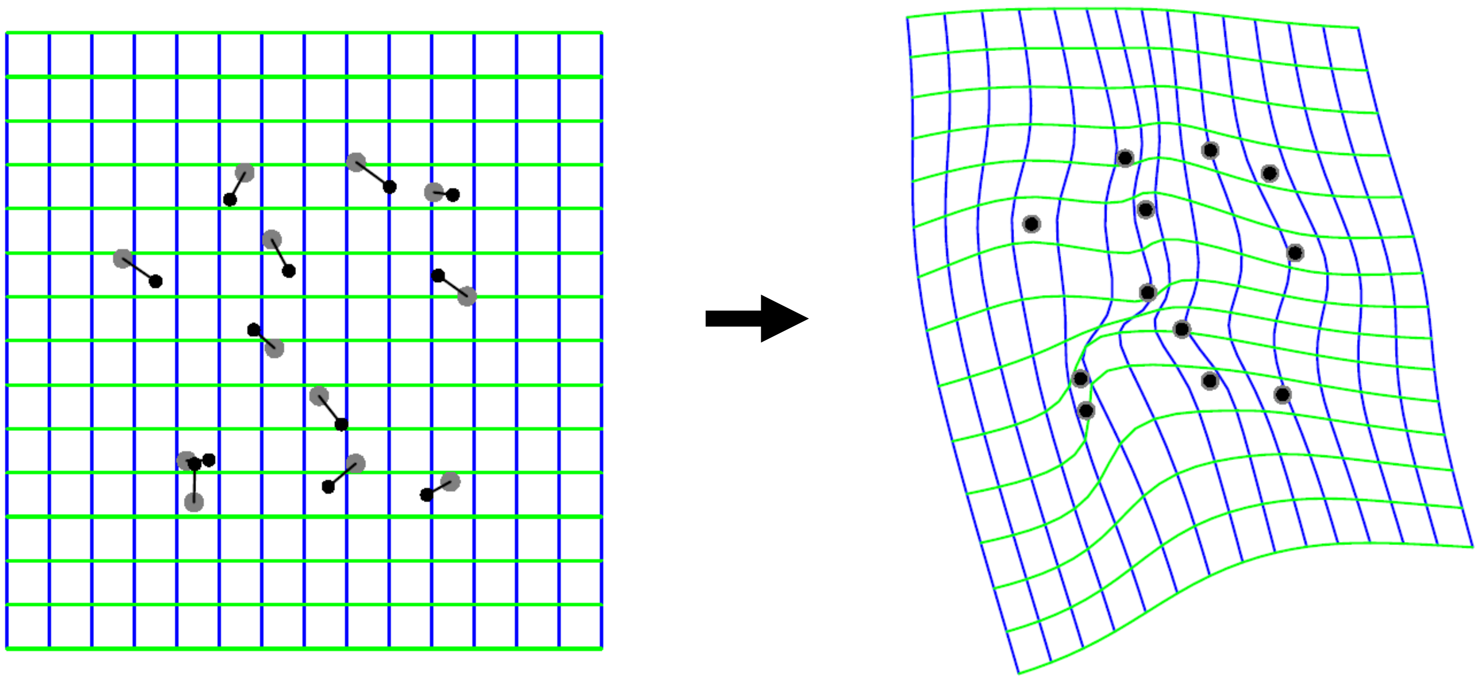
$R(f)$ $>=0$

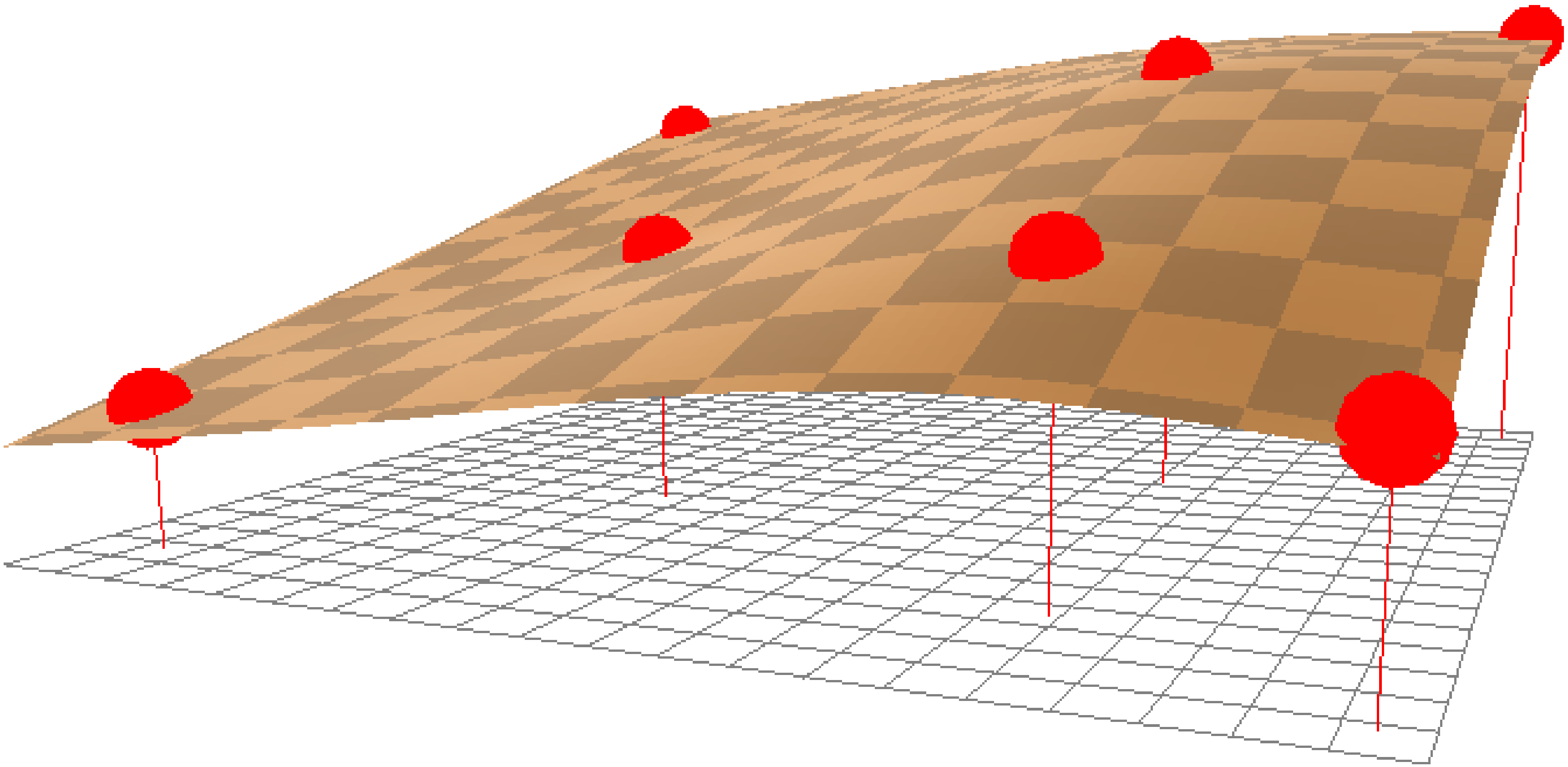
$||L(f)||$

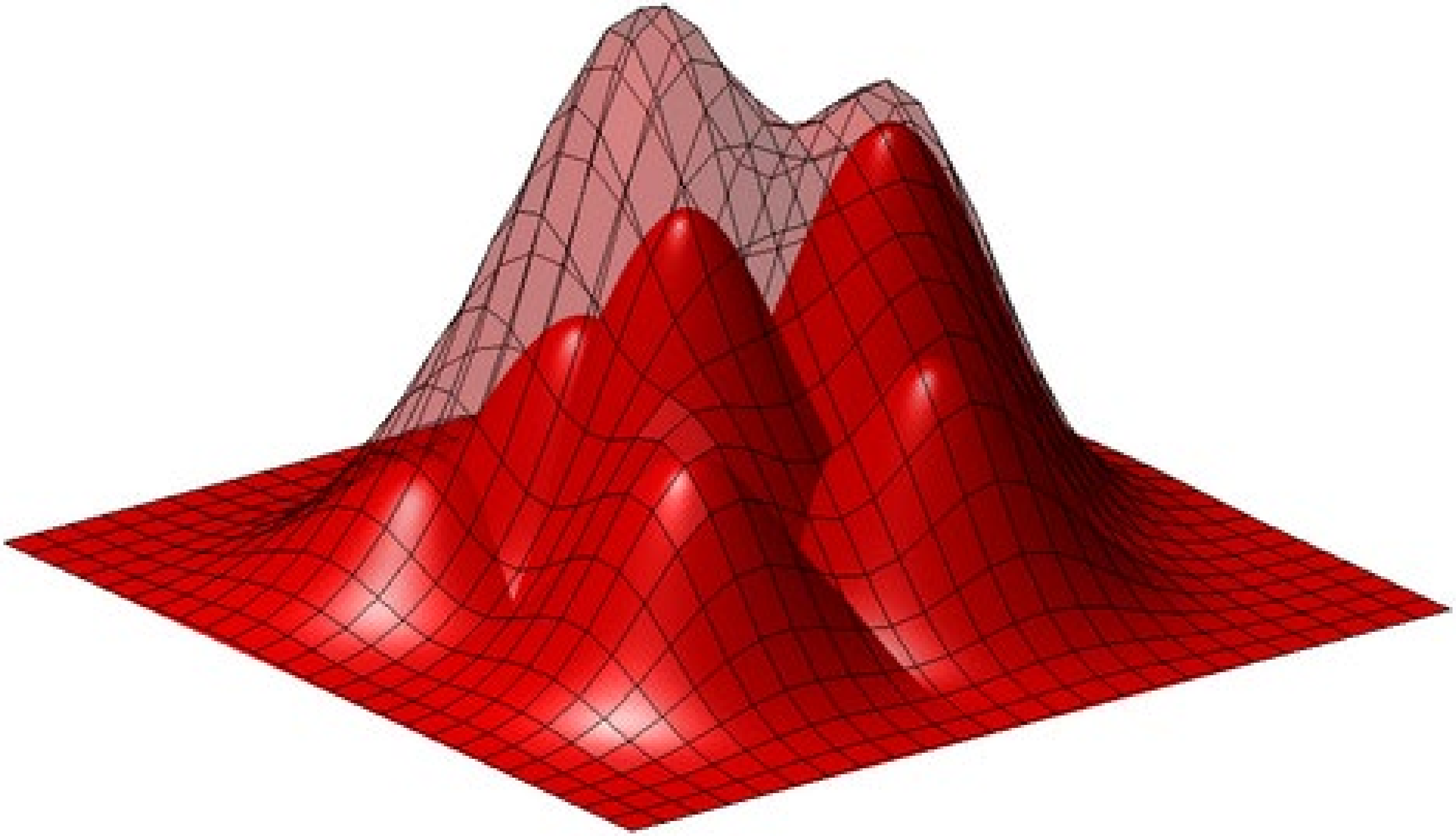
$a << b$ least-square fit,

f from the null-space of L

$a >> b$ “smooth” interpolation







TRANSFORM MODEL ESTIMATION UNIFIED APPROACH

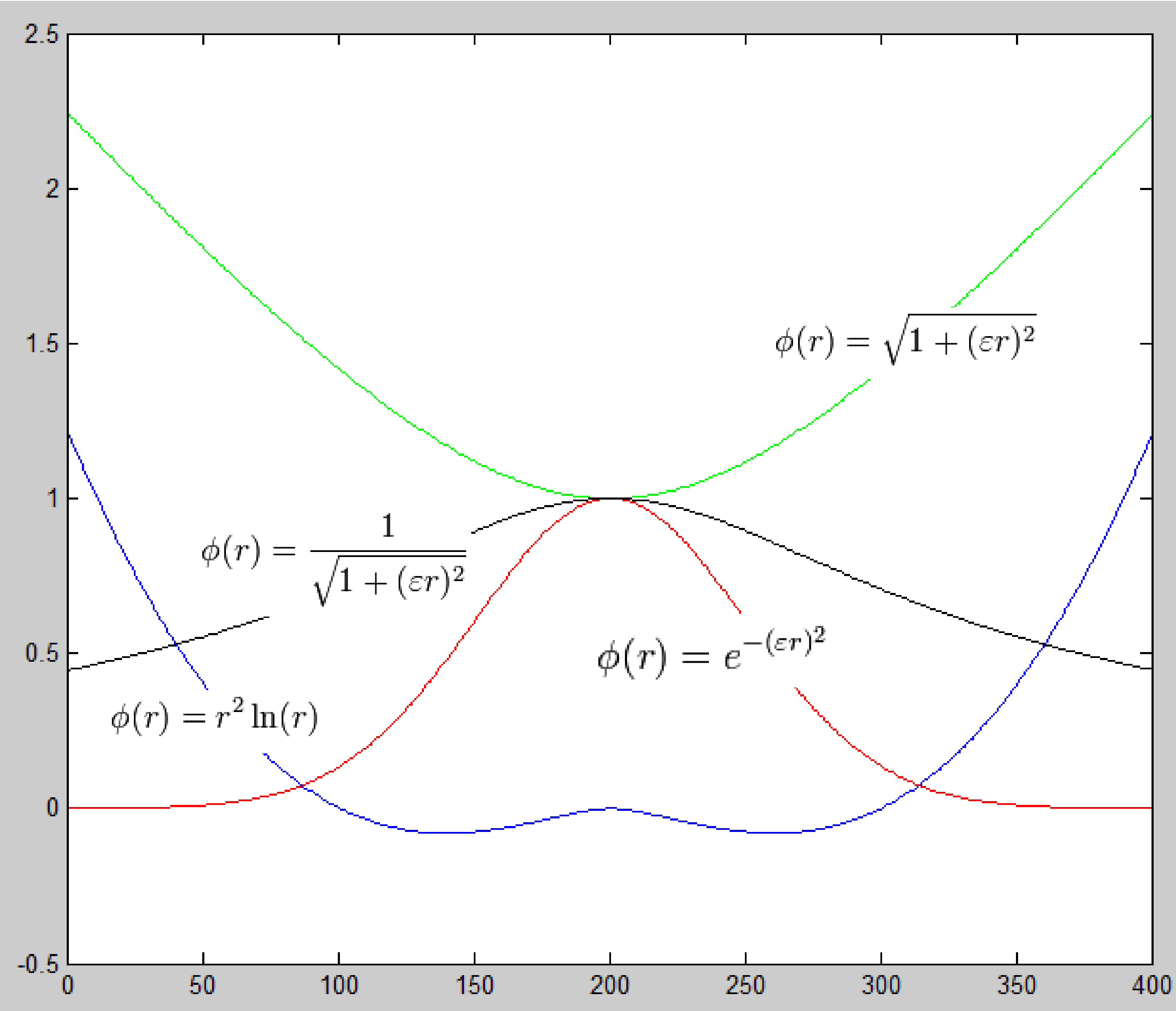
Choices for **min** $J(f) = a\,E(f) + b\,R(f)$

$$R(f) = \iint \left(\frac{\partial^2 f}{\partial x \partial x}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y \partial y}\right)^2 dx dy$$

$$f(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \sum_{i=1}^{N} a_i g_i(\|x - x_i, y - y_i\|),$$

TPS $g_i(t) = t^2 \log t$

another choice **G-RBF** $g_i(t) = \exp\left(\frac{-t^2}{\sigma^2}\right)$



TPS

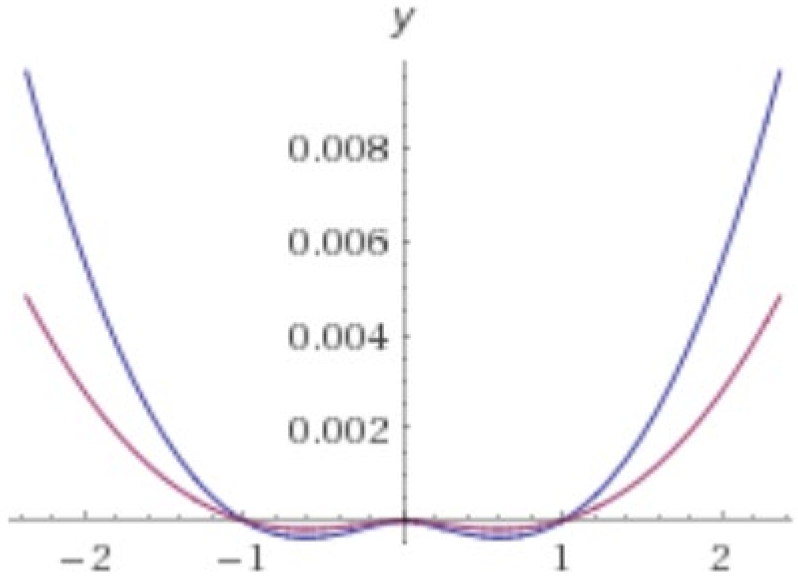
G-RBF

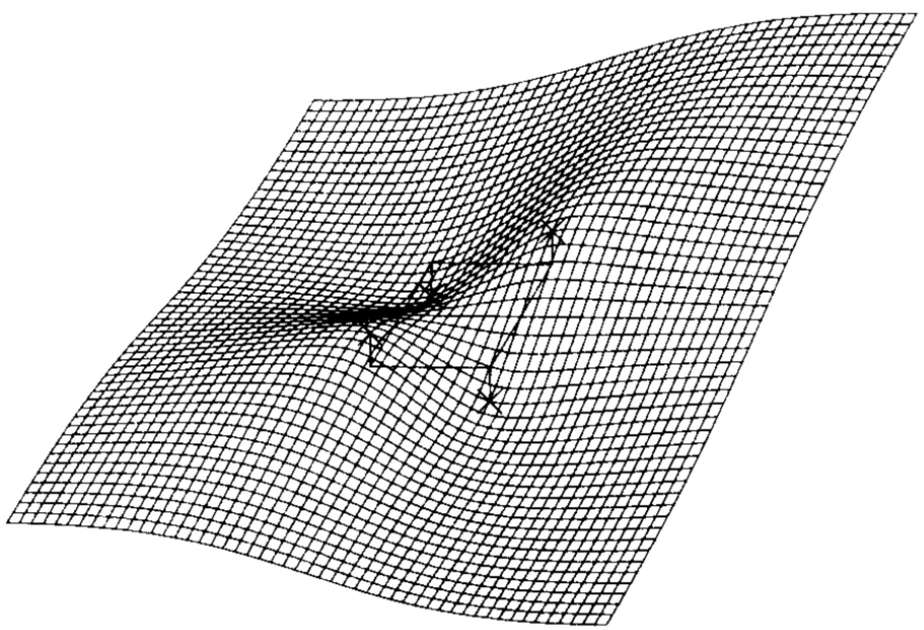
multiquadrics

inverse multiquadrics

$$\varphi(r) = r^2 \log r^2$$

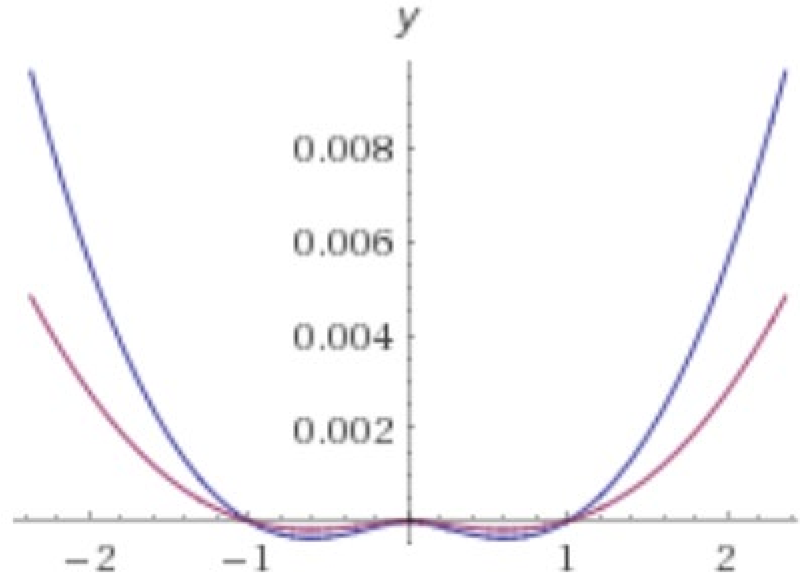
$$\varphi(r) = r^2 \log r$$

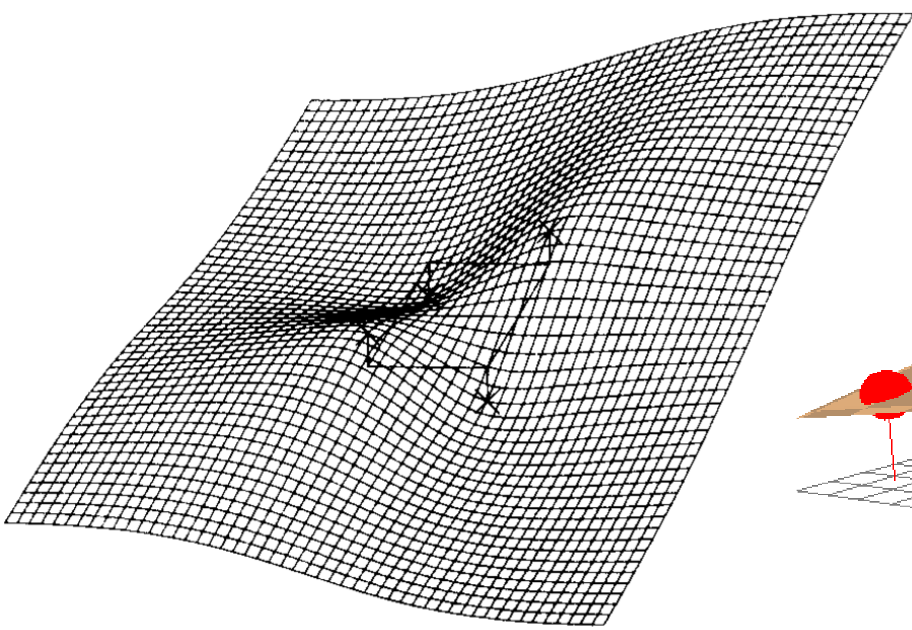


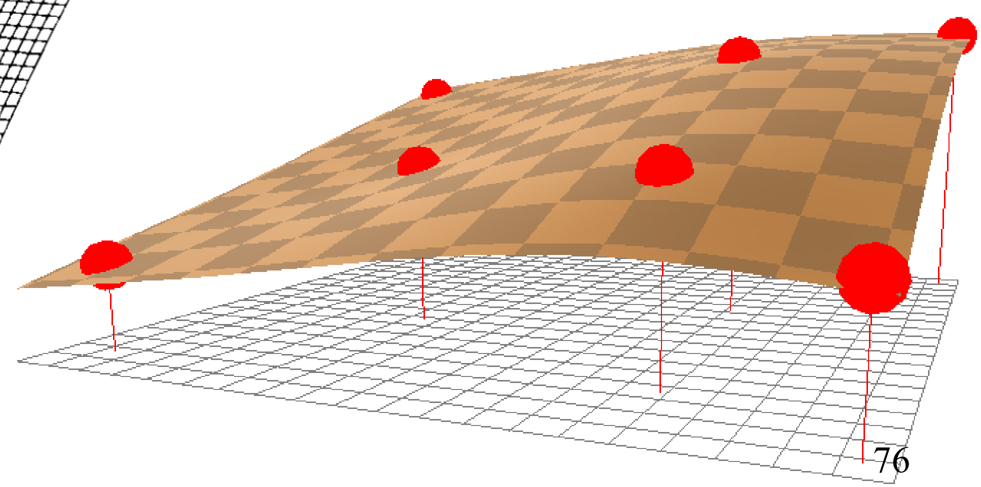


$$\varphi(r) = r^2 \log r^2$$

$$\varphi(r) = r^2 \log r$$







Registrace s TPS

1. N dvojic bodů (x,y) → (x',y')
2. Nalézt 6 + 2*N* koeficientů

$$a_0, a_1, a_2, b_0, b_1, b_2, F_i, G_i$$

$$x' = a_0 + a_1 x + a_2 y + \sum_{i=1}^{N} F_i r_i^2 \ln r_i^2$$

$$y' = b_0 + b_1 x + b_2 y + \sum_{i=1}^{N} G_i r_i^2 \ln r_i^2$$

...

Registrace s TPS

… ještě těchto 6 rovnic

$$\sum_{i=1}^{N} F_i = 0 \qquad \sum_{i=1}^{N} G_i = 0$$

$$\sum_{i=1}^{N} x_i F_i = 0 \qquad \sum_{i=1}^{N} x_i G_i = 0$$

$$\sum_{i=1}^{N} y_i F_i = 0 \qquad \sum_{i=1}^{N} y_i G_i = 0$$

Určete ostatní body obrázku

Jak to, že to funguje?

X→pryč od N bodů výraz $\sum_{i=1}^{N} F_i r_i^2 \ln r_i^2$

mizí, suma jde k 0, totéž pro y, tedy se obje-
vuje vztah

$$x' \rightarrow a_0 + a_1 x + a_2 y$$
$$y' \rightarrow b_0 + b_1 x + b_2 y$$

Jak to, že to funguje?

Vhodné pro situace s malou transformací

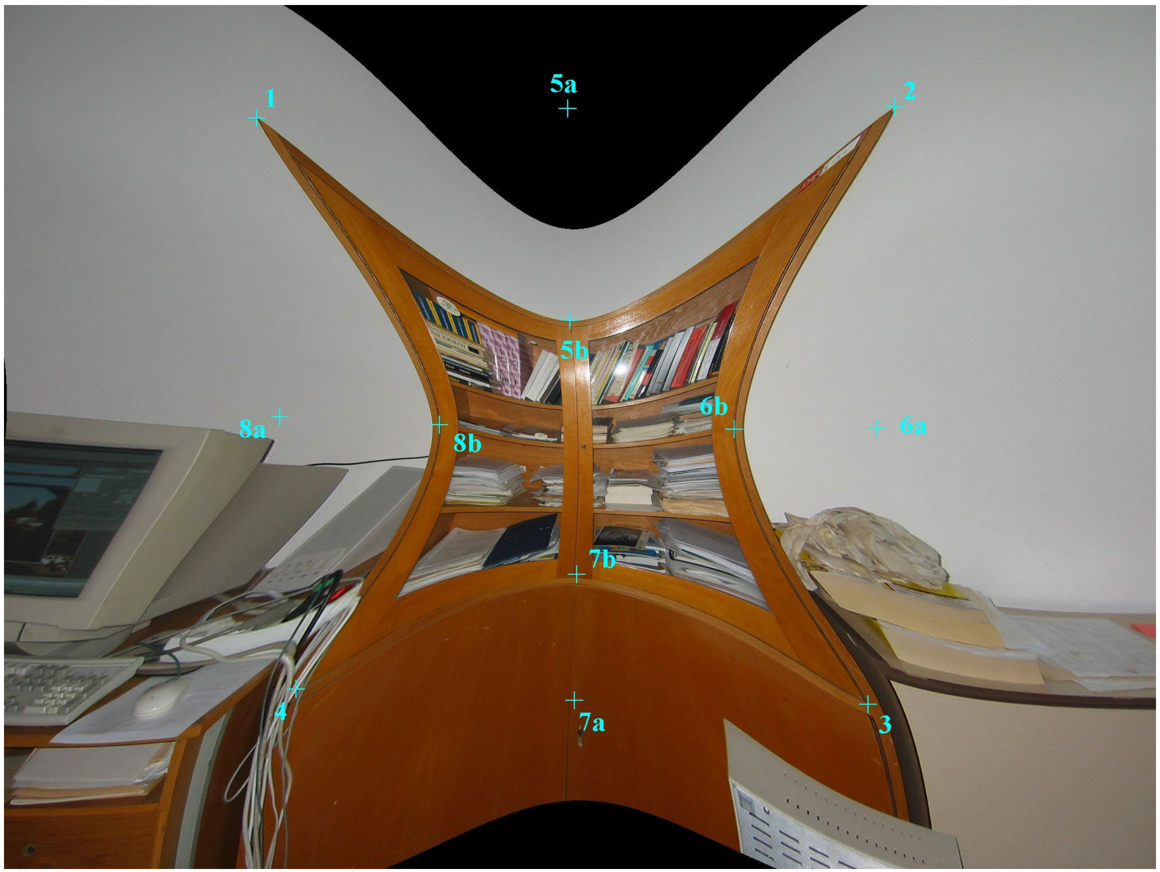
Body uniformně

Dostatek bodů a rovnoměrně

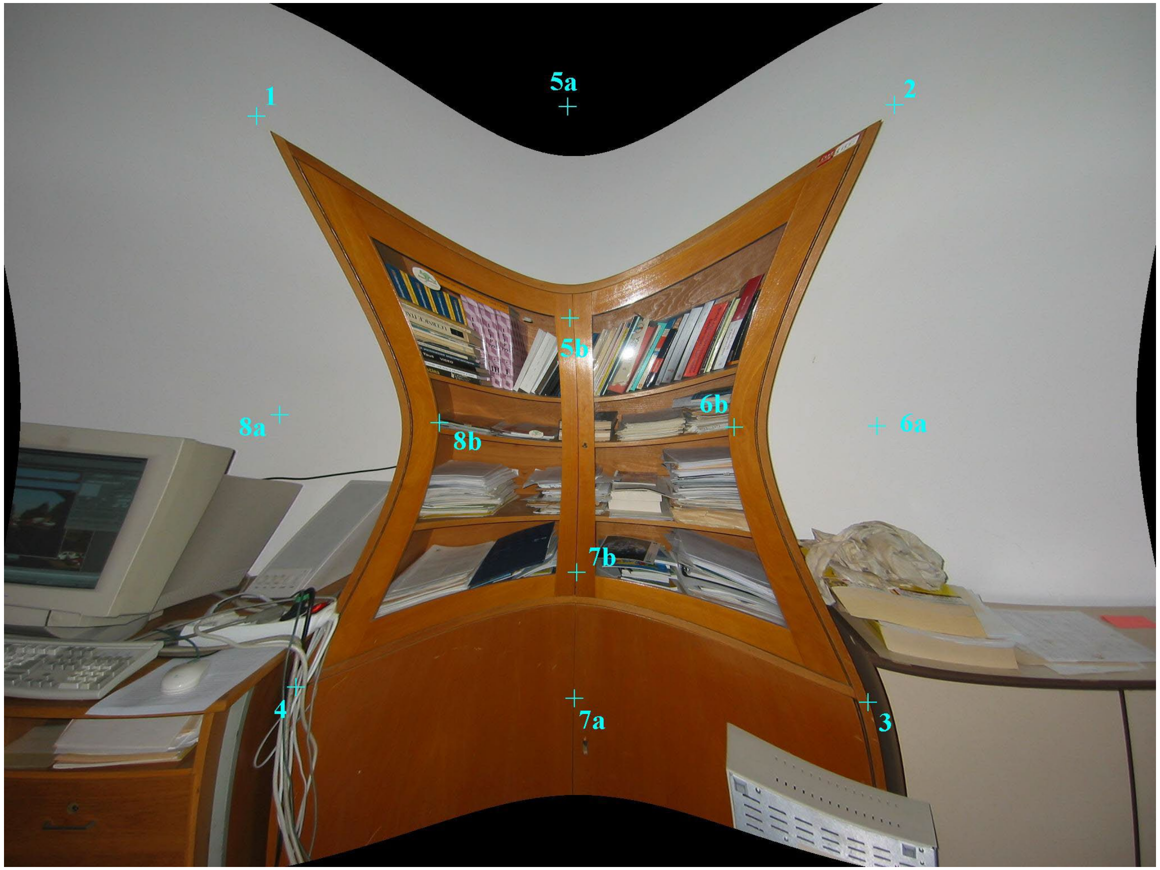
TRANSFORM MODEL ESTIMATION THIN-PLATE SPLINES

original

TPS, *a >> b,* “smooth” interpolation

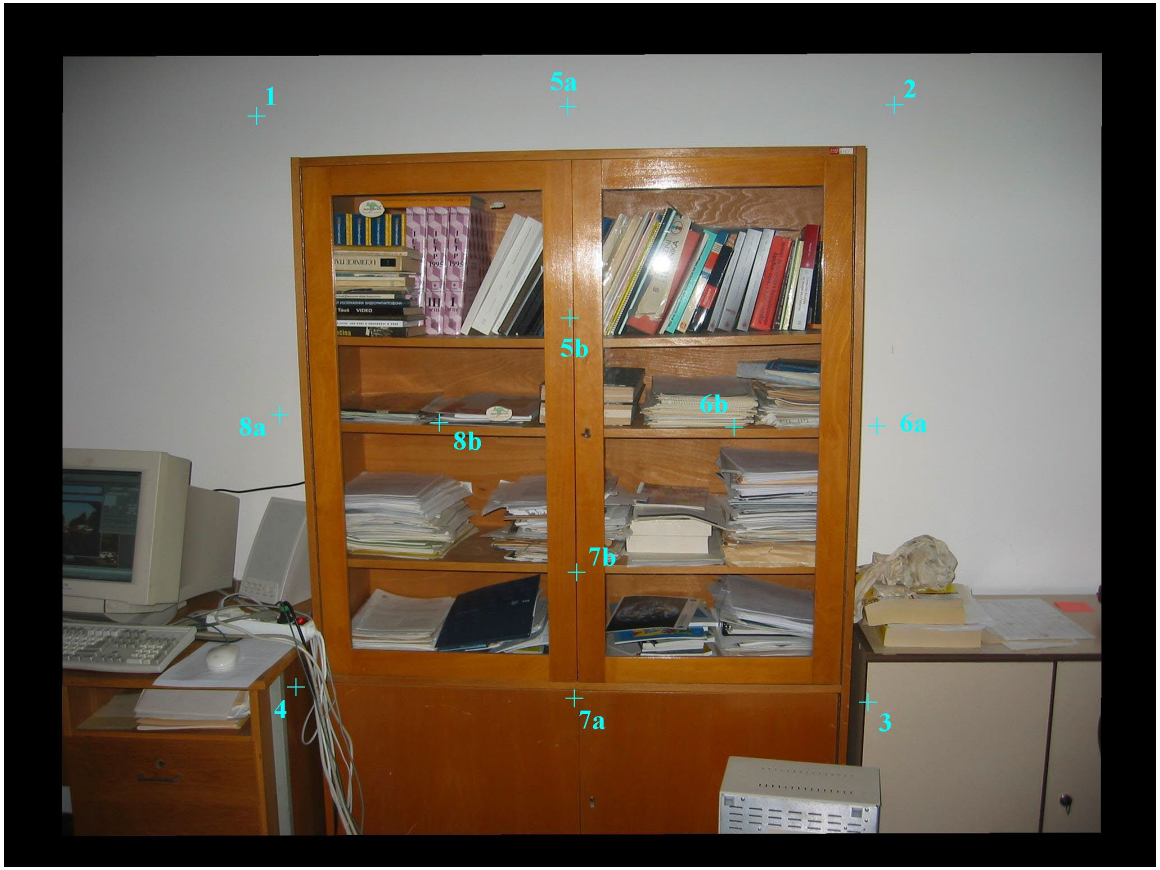


TPS, *a > b*



TPS, *a* < *b*

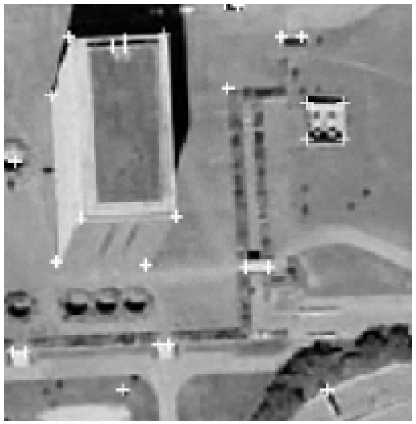
TPS, ***a << b,*** least-square fit



TRANSFORM MODEL ESTIMATION THIN-PLATE SPLINES

original

TRANSFORM MODEL ESTIMATION TPS x G-RBF



TRANSFORM MODEL ESTIMATION RIGIDITY

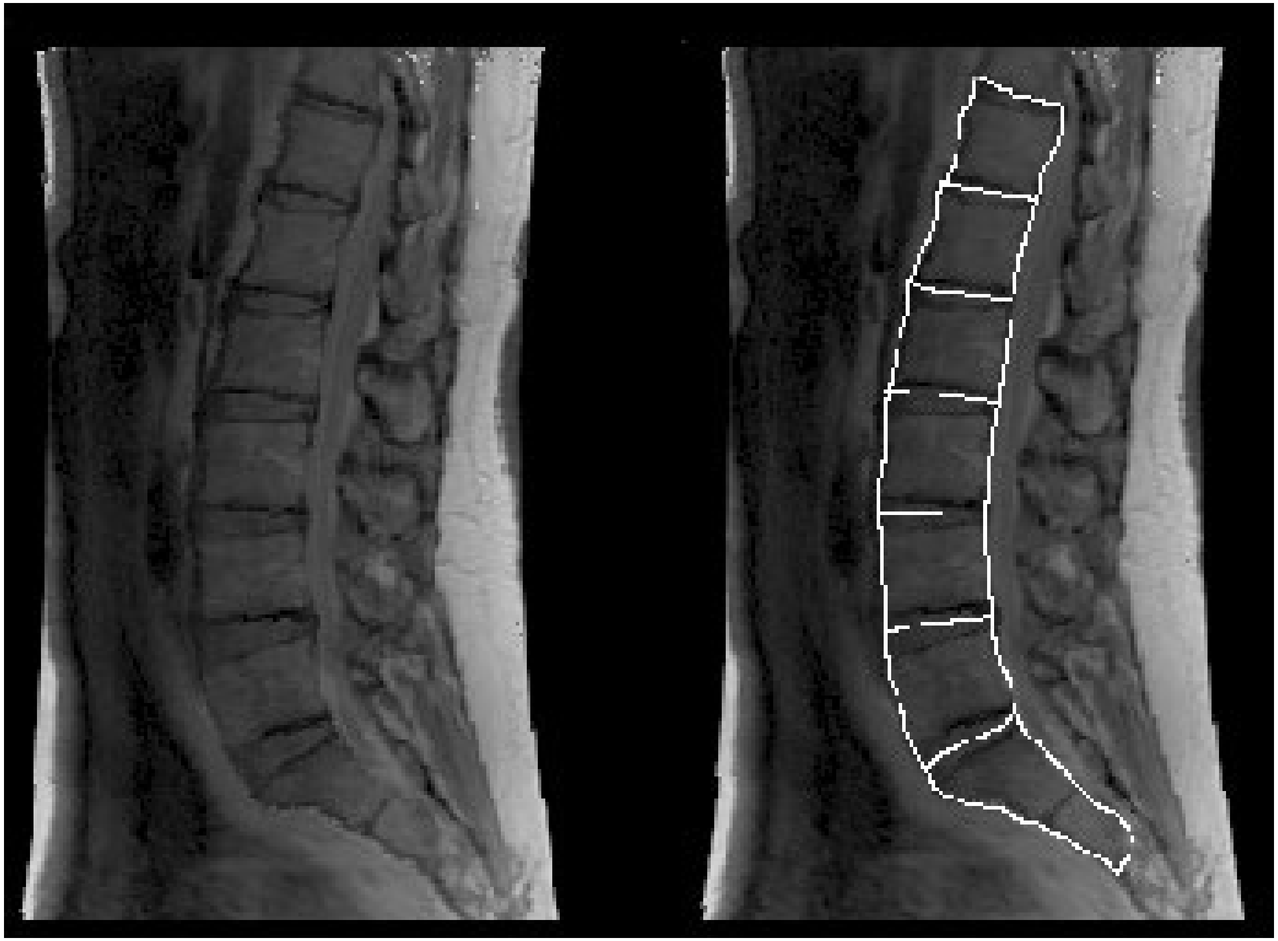


IMAGE RESAMPLING AND TRANSFORMATION

trade-off between accuracy and computational complexity

IMAGE RESAMPLING AND TRANSFORMATION

forward method

backward method

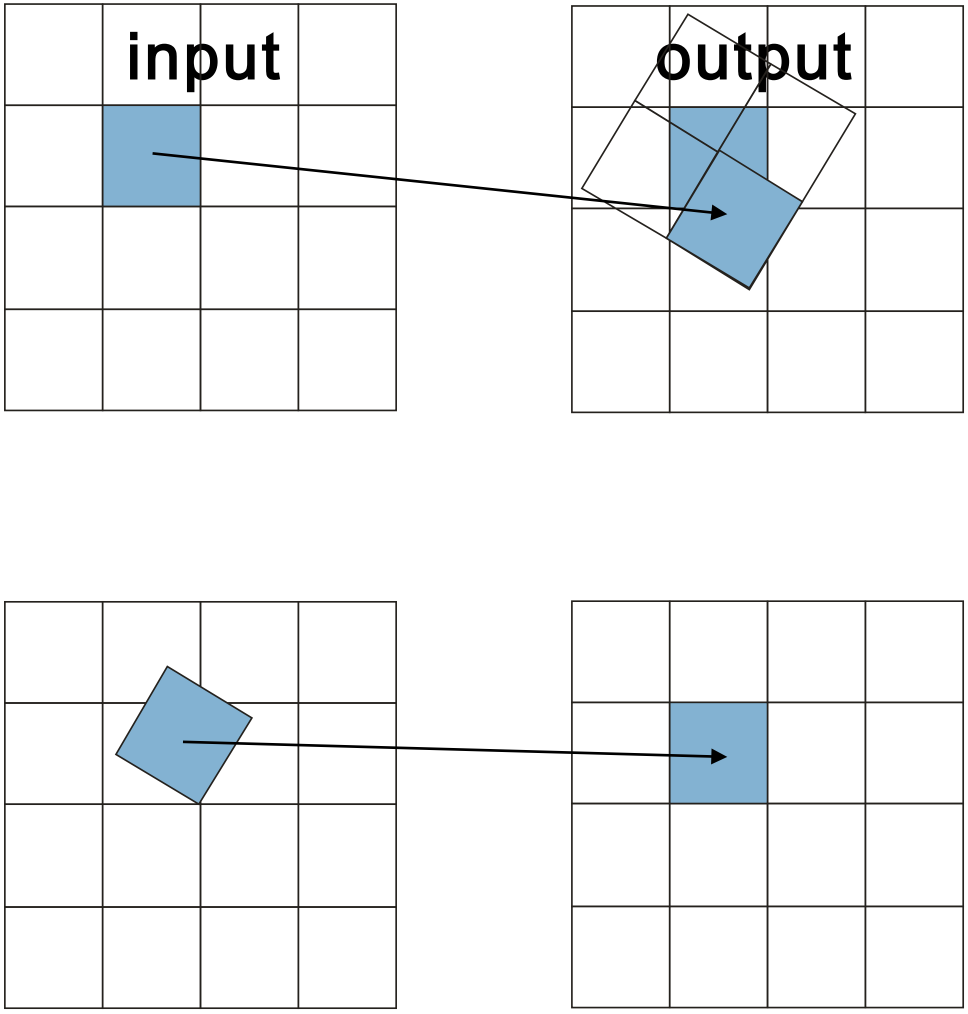


IMAGE RESAMPLING AND TRANSFORMATION

forward method

$B[u(x, y), v(x, y)] = A[x, y]$

Nemapuje se vždy na pozice pixelů ->

INTERPOLACE

Může produkovat díry

backward method

$B[u, v] = A[x(u, v), y(u, v)]$

Nemapuje se vždy z pozice pixelů

INTERPOLACE

Může nepostihnout všechny vstupní pixely

IMAGE RESAMPLING AND TRANSFORMATION

original

nearest neighbor

bilinear

bicubic

IMAGE RESAMPLING AND TRANSFORMATION

Interpolation   nearest neighbor

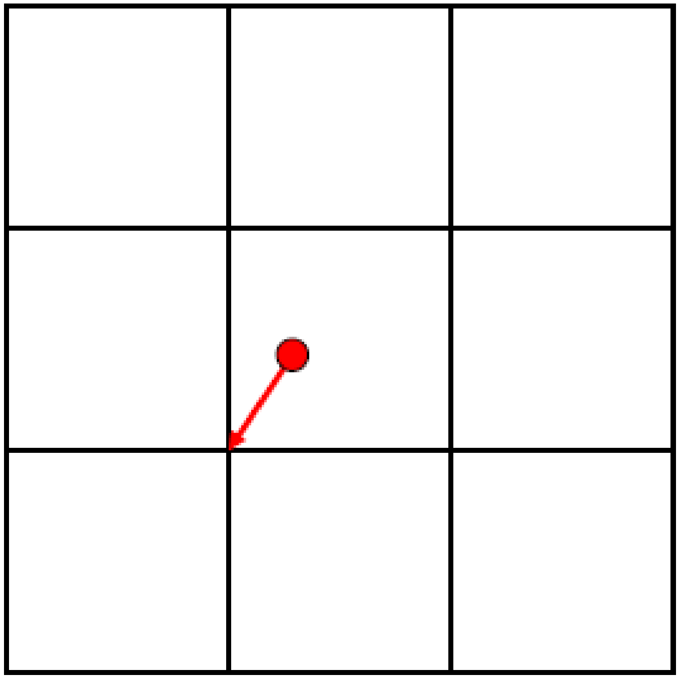
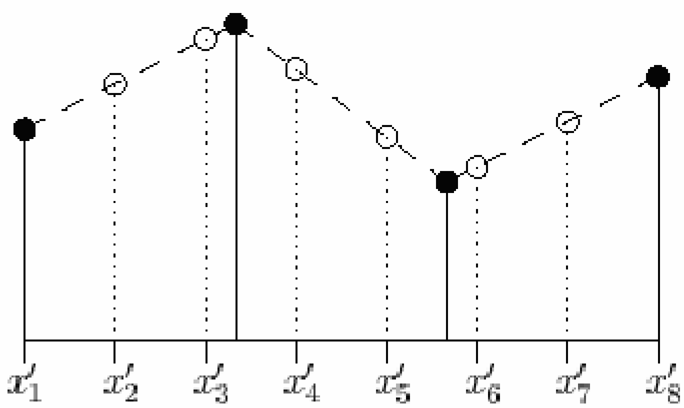


IMAGE RESAMPLING AND TRANSFORMATION

Interpolation   **nearest neighbor**

bilinear



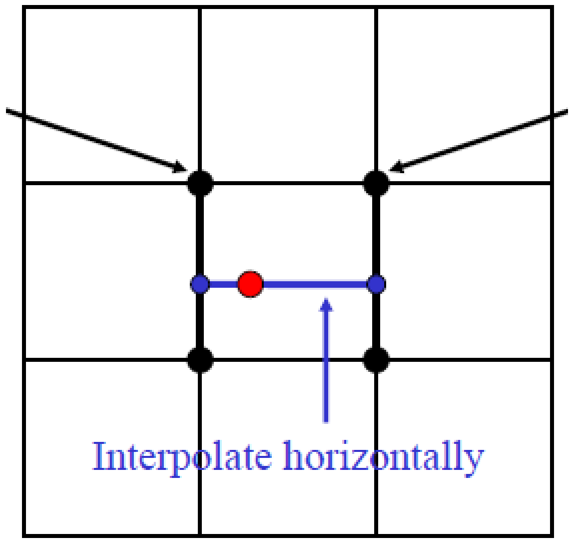


IMAGE RESAMPLING AND TRANSFORMATION

Interpolation nearest neighbor

bilinear

bicubic

Implementation 1-D convolution

$$f(x_o,k) = \Sigma\, d(I,k).c(i-x_0)$$

$$f(x_0, y_0) = \Sigma\, f(x_0, j).c(j-y_0)$$

ideal $c(x) = k.\mathrm{sinc}(kx)$

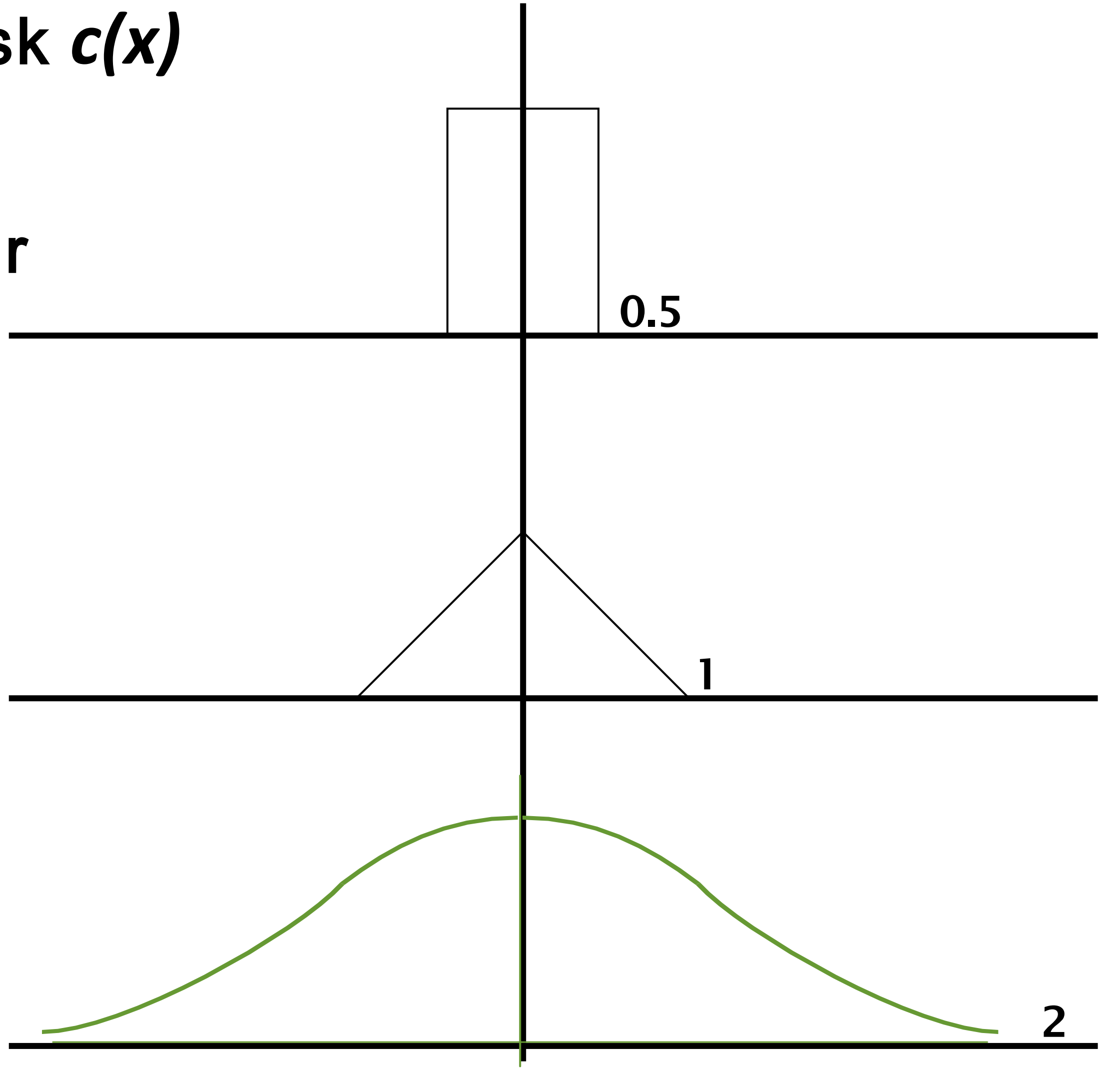
IMAGE RESAMPLING AND TRANSFORMATION

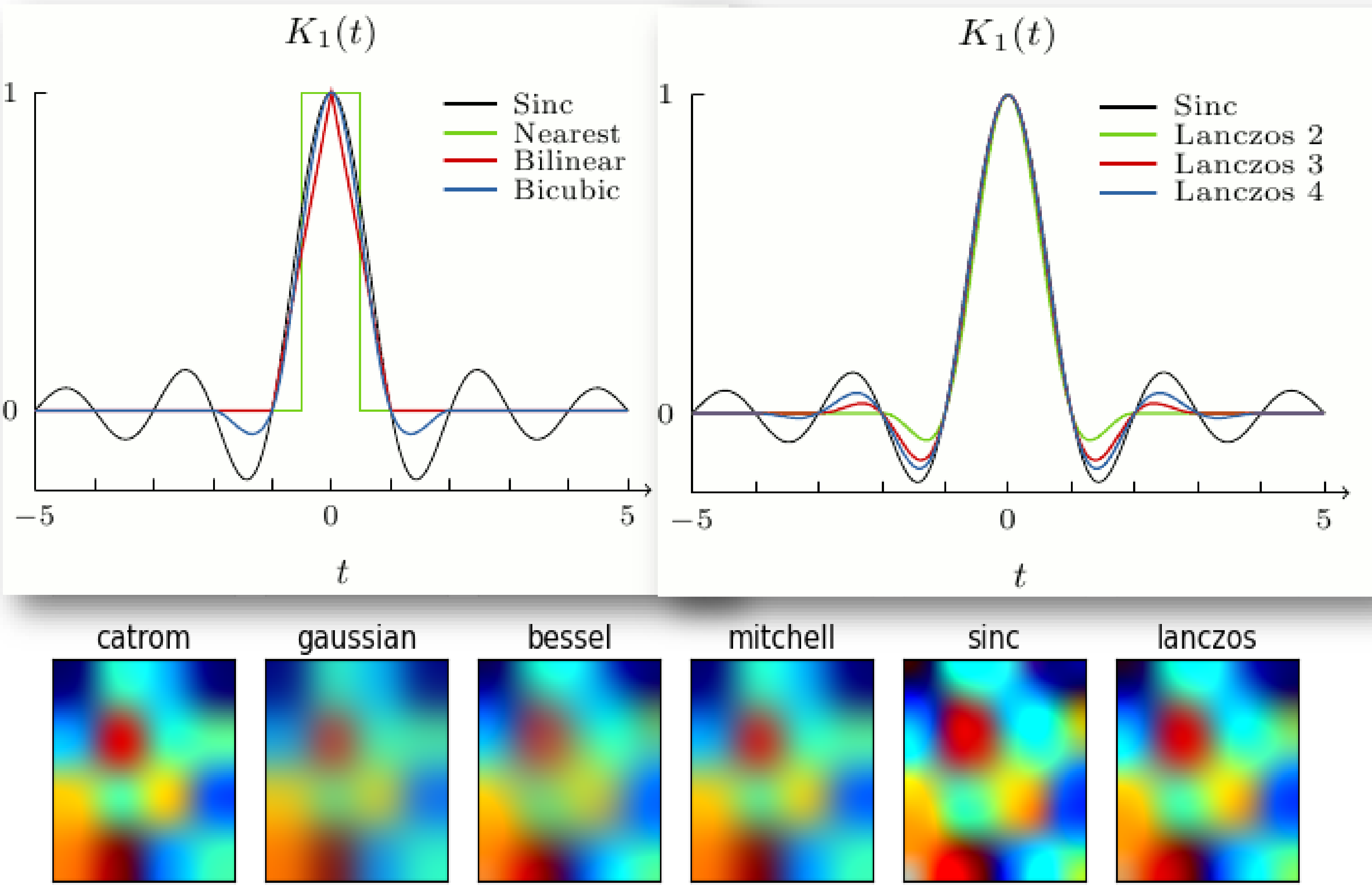
Interpolation mask *c(x)*

closest neighbour

linear

smooth cubic

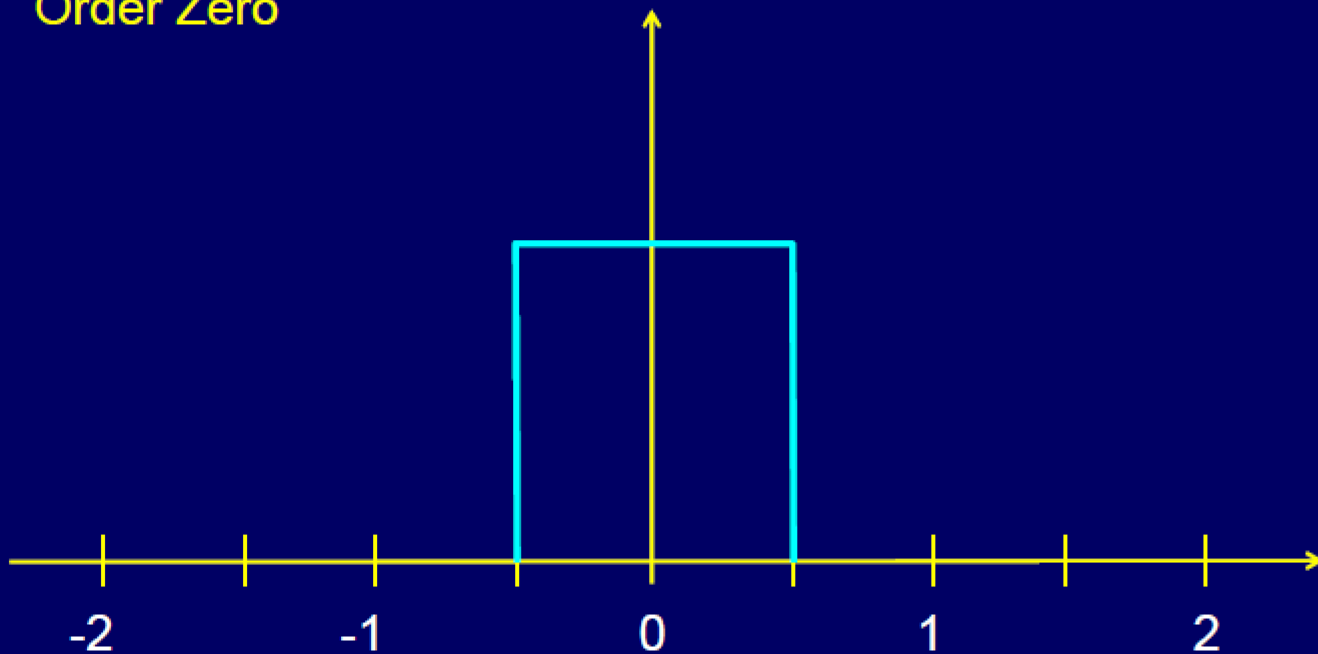




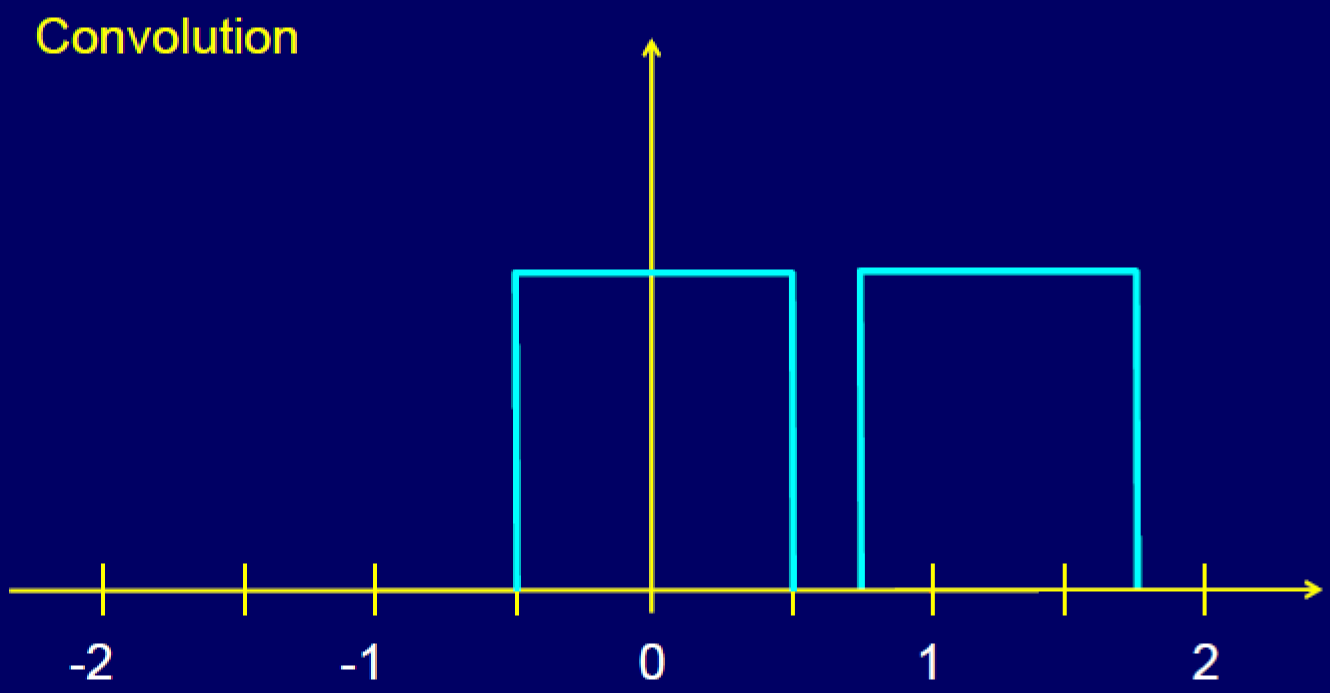
$K_1(t)$
Sinc
Nearest
Bilinear
Bicubic
1
0
−5
0
5
t
$K_1(t)$
Sinc
Lanczos 2
Lanczos 3
Lanczos 4
1
0
−5
0
5
t
catrom
gaussian
bessel
mitchell
sinc
lanczos

Interpolation Kernel

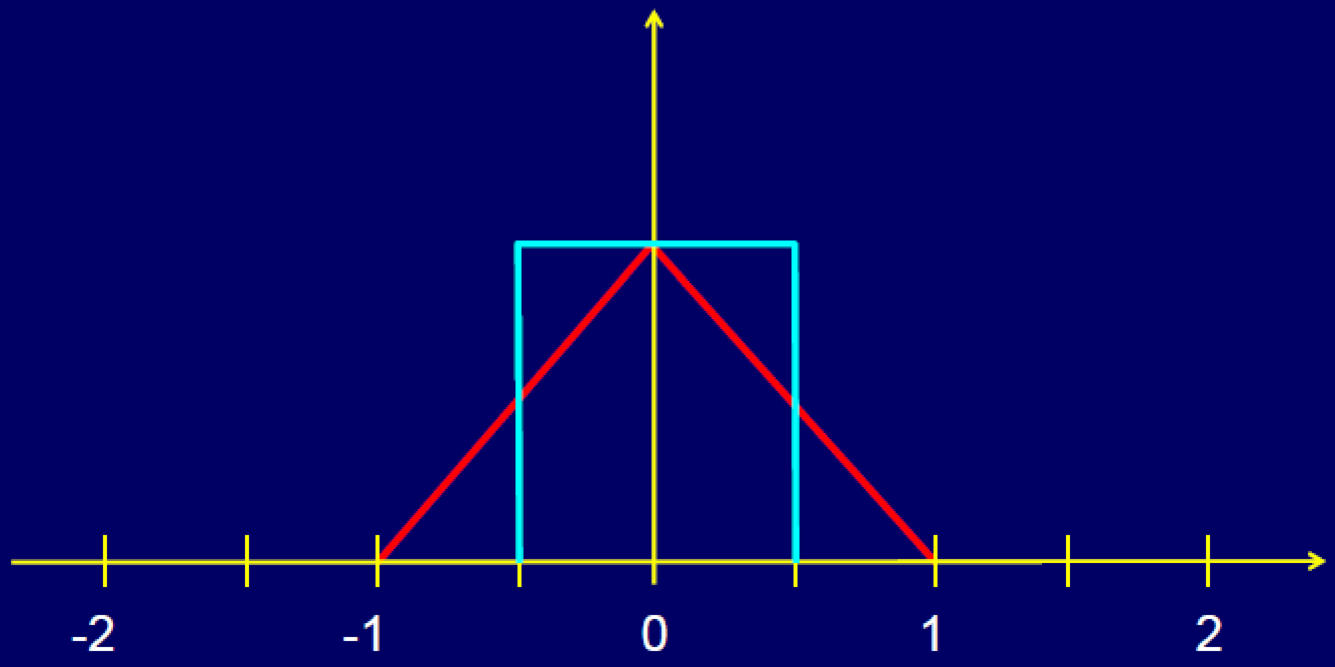
Order Zero



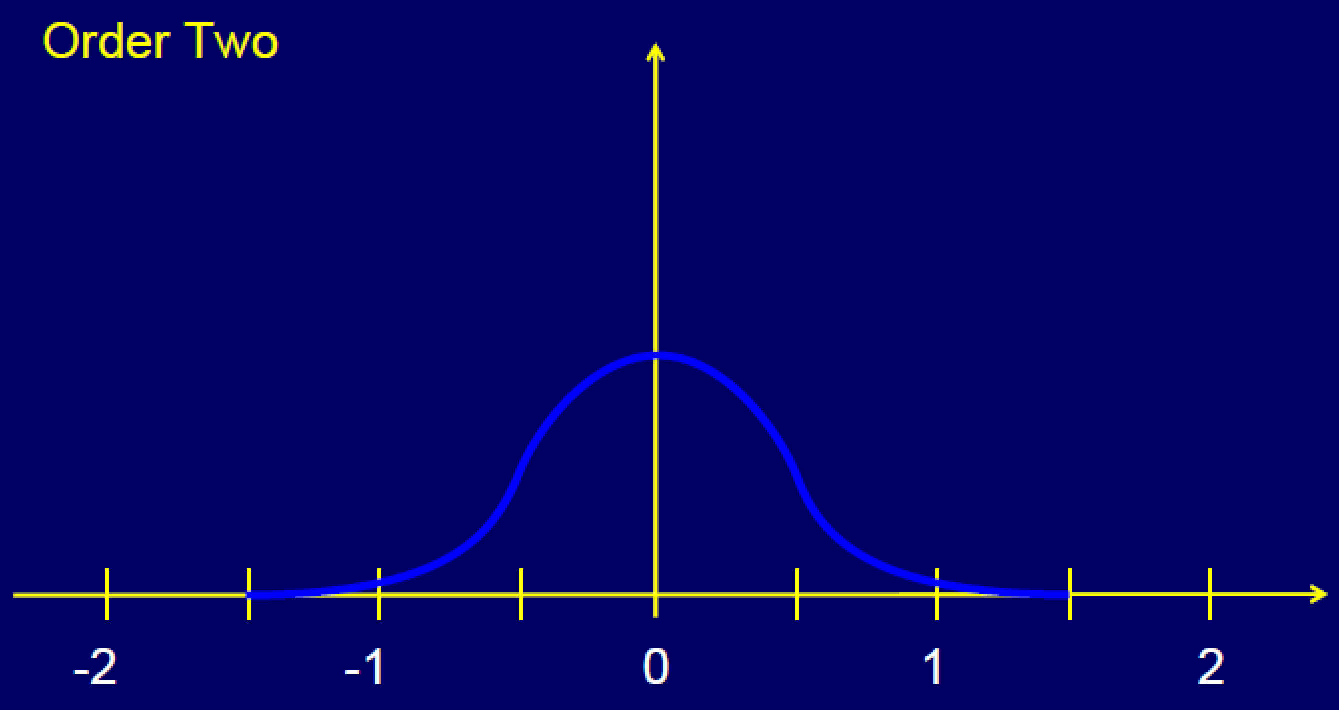
Interpolation Kernel



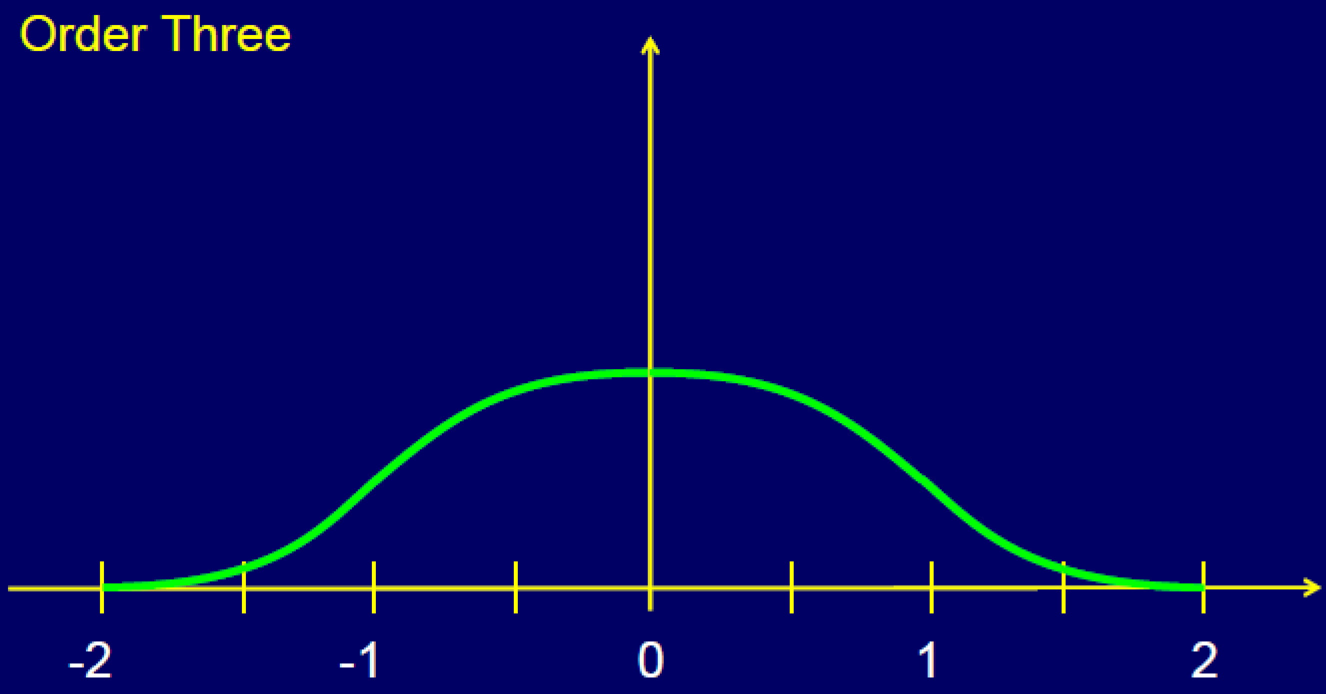
Interpolation Kernel



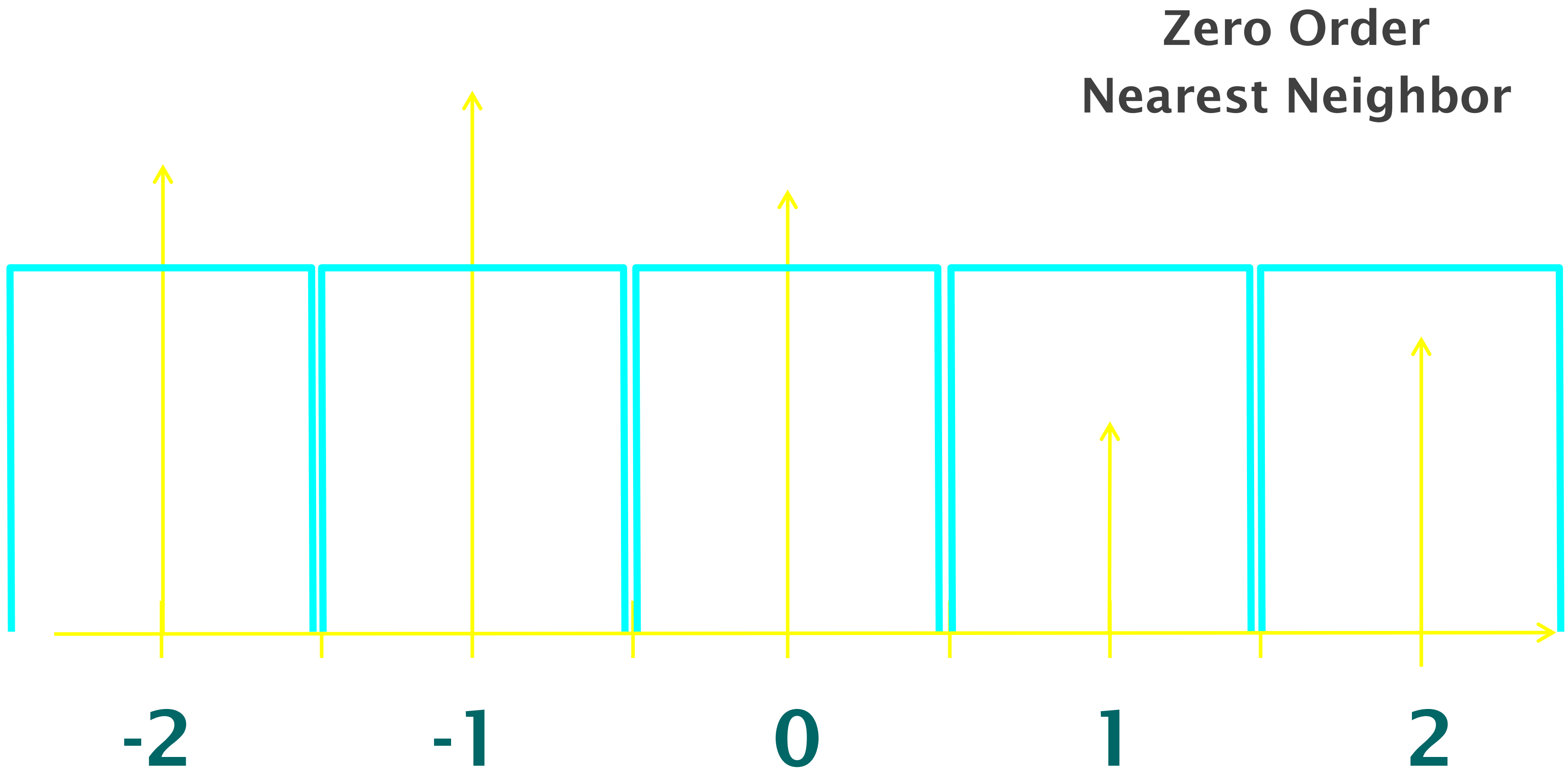
Interpolation Kernel



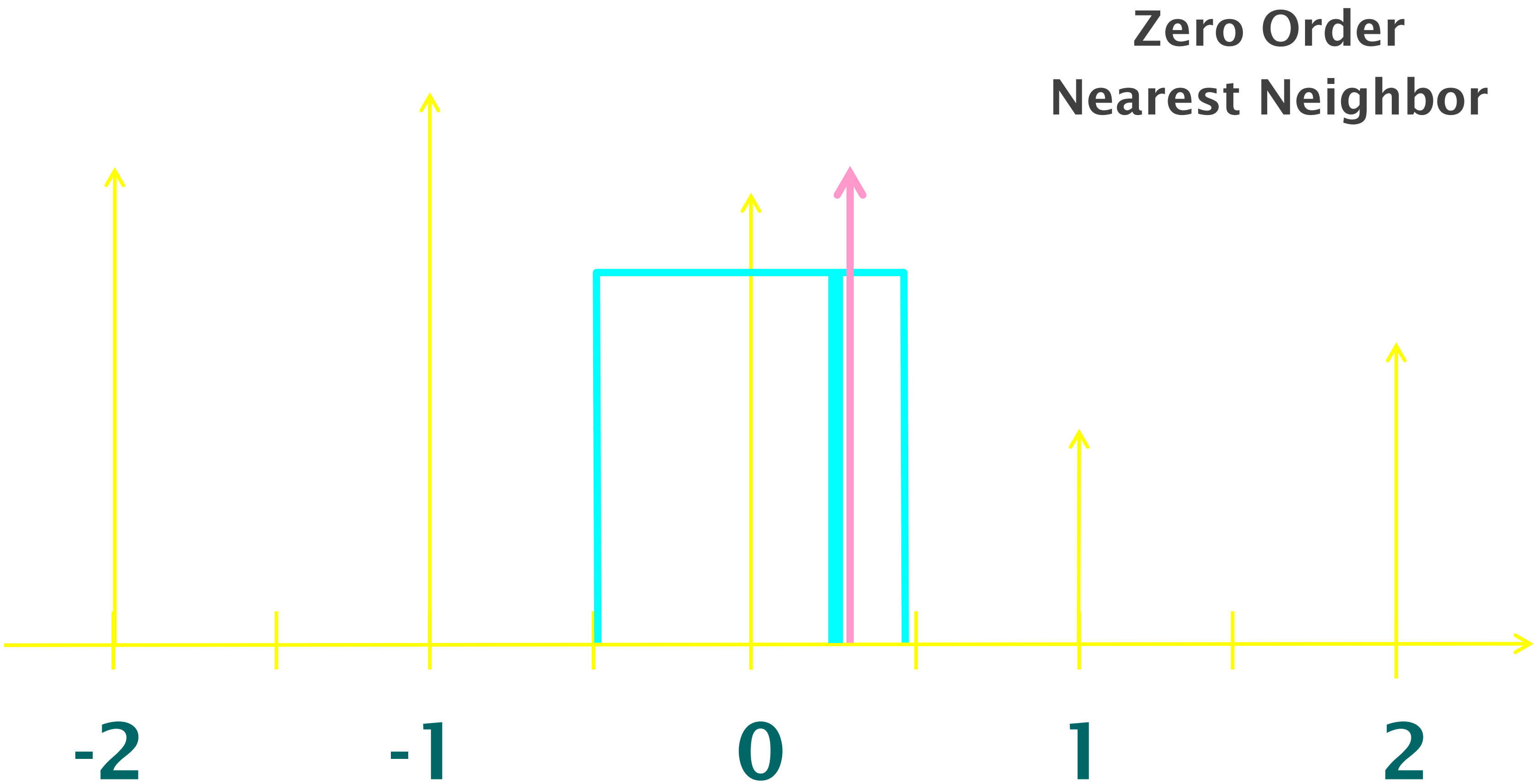
Interpolation Kernel



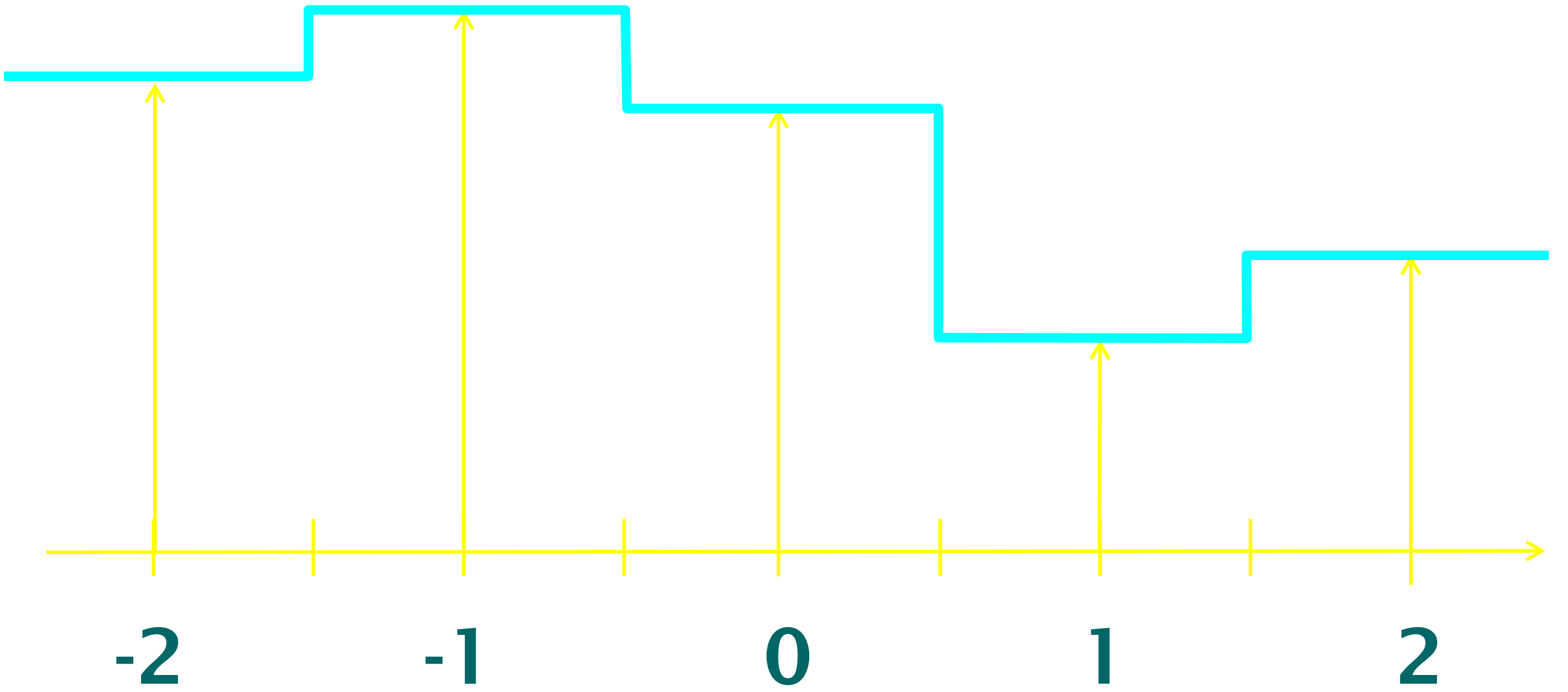
Interpolation



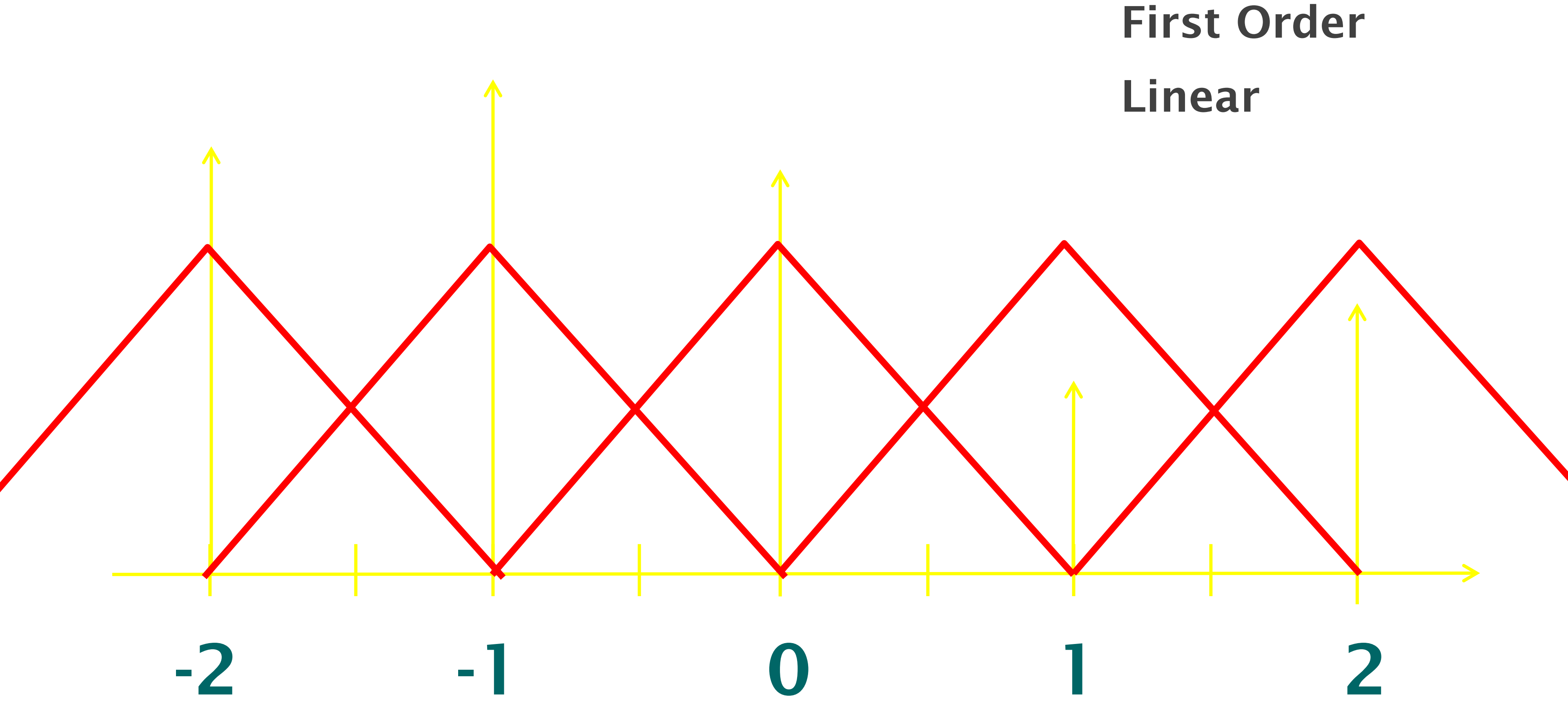
Interpolation



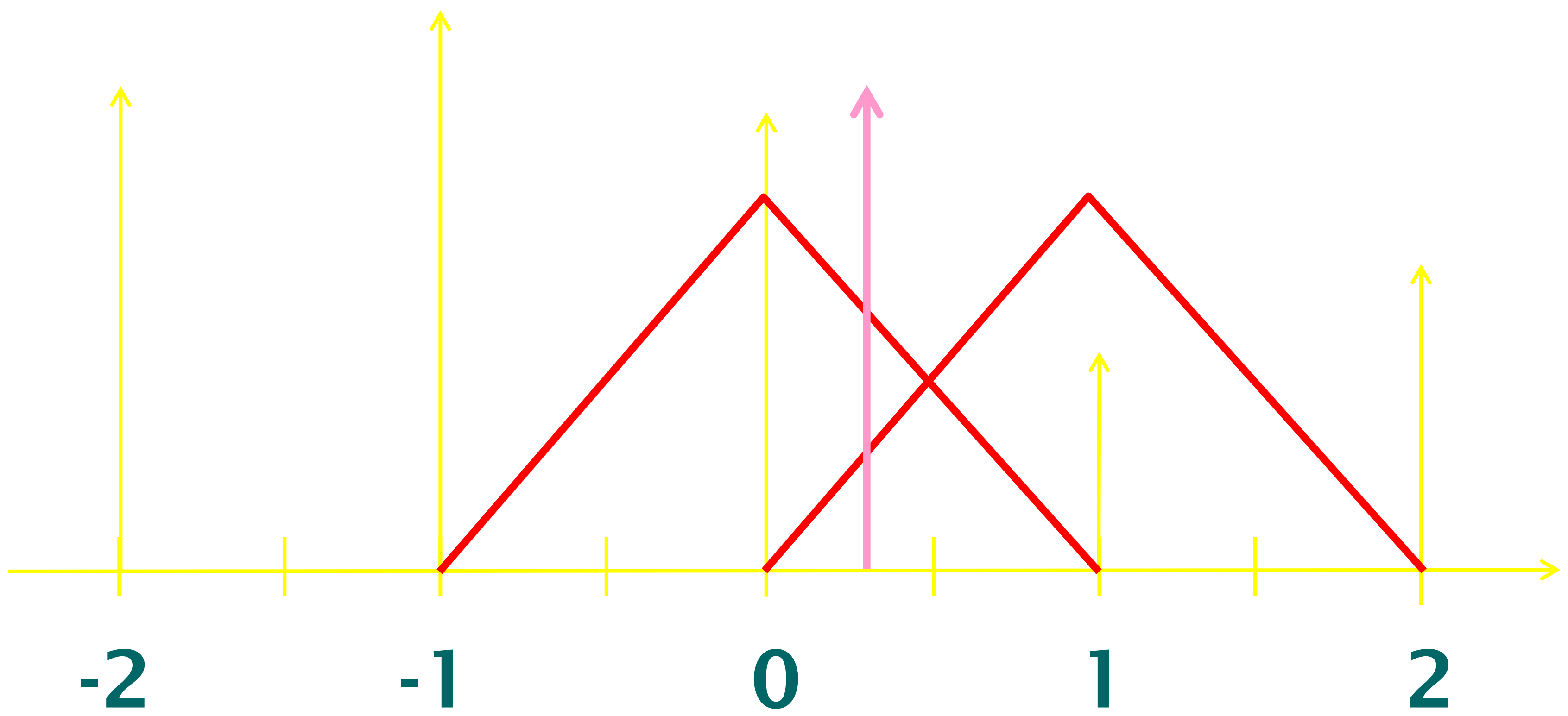
Interpolation



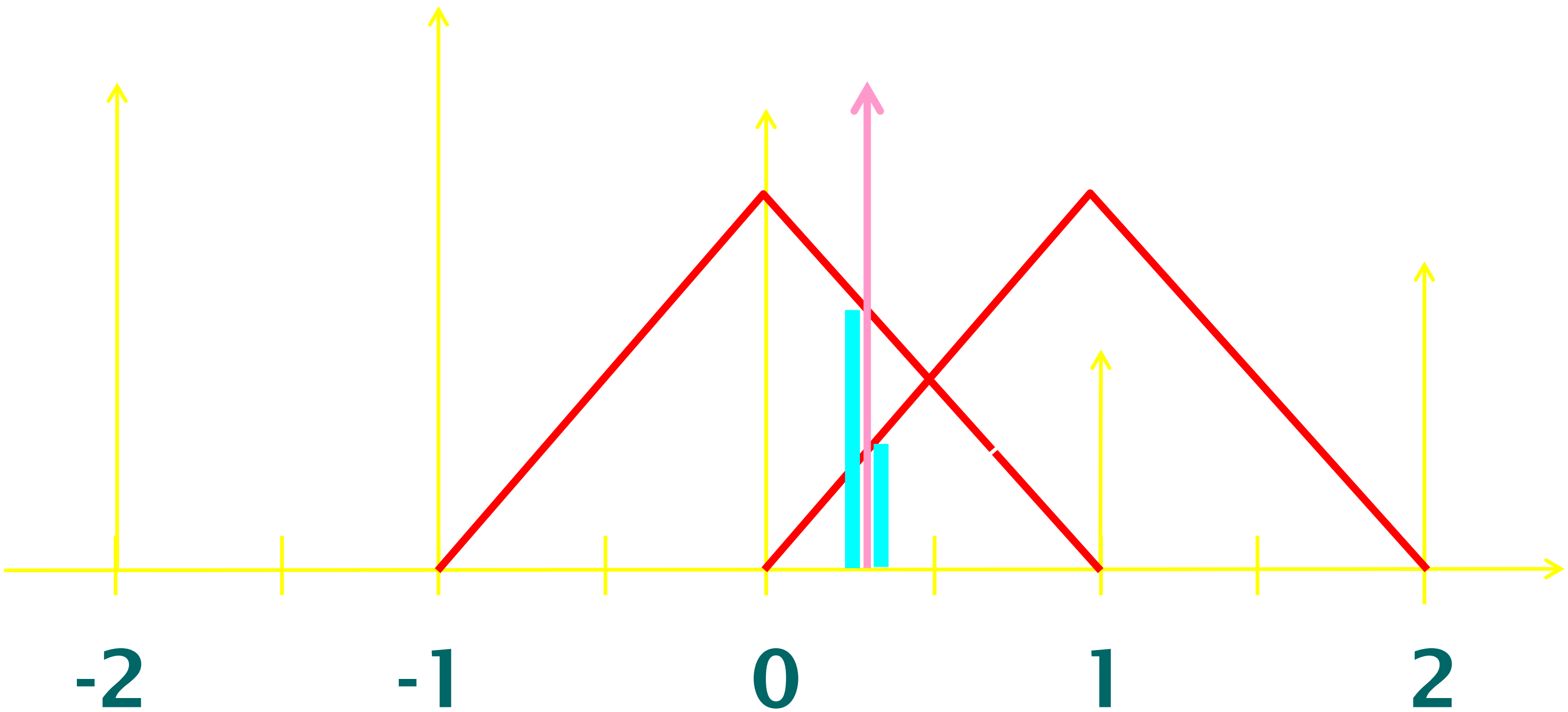
Interpolation



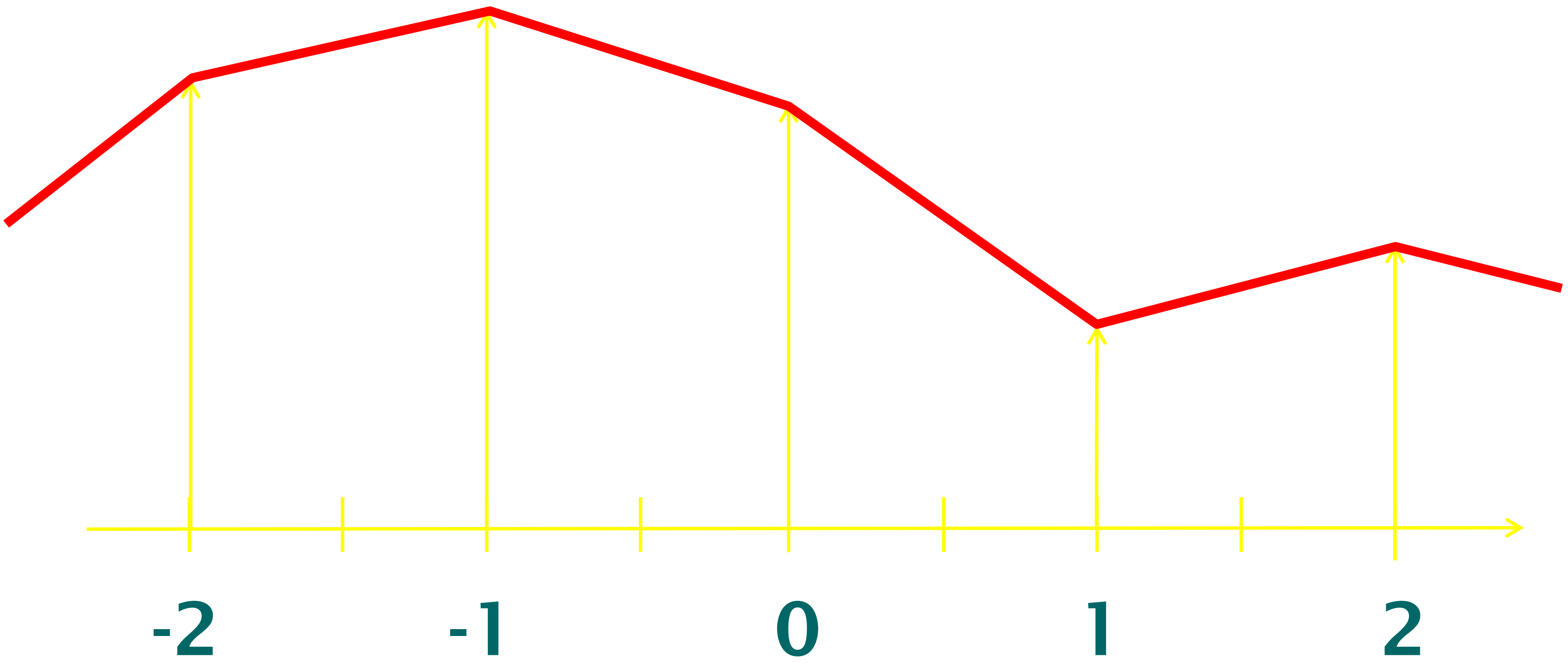
Interpolation



Interpolation



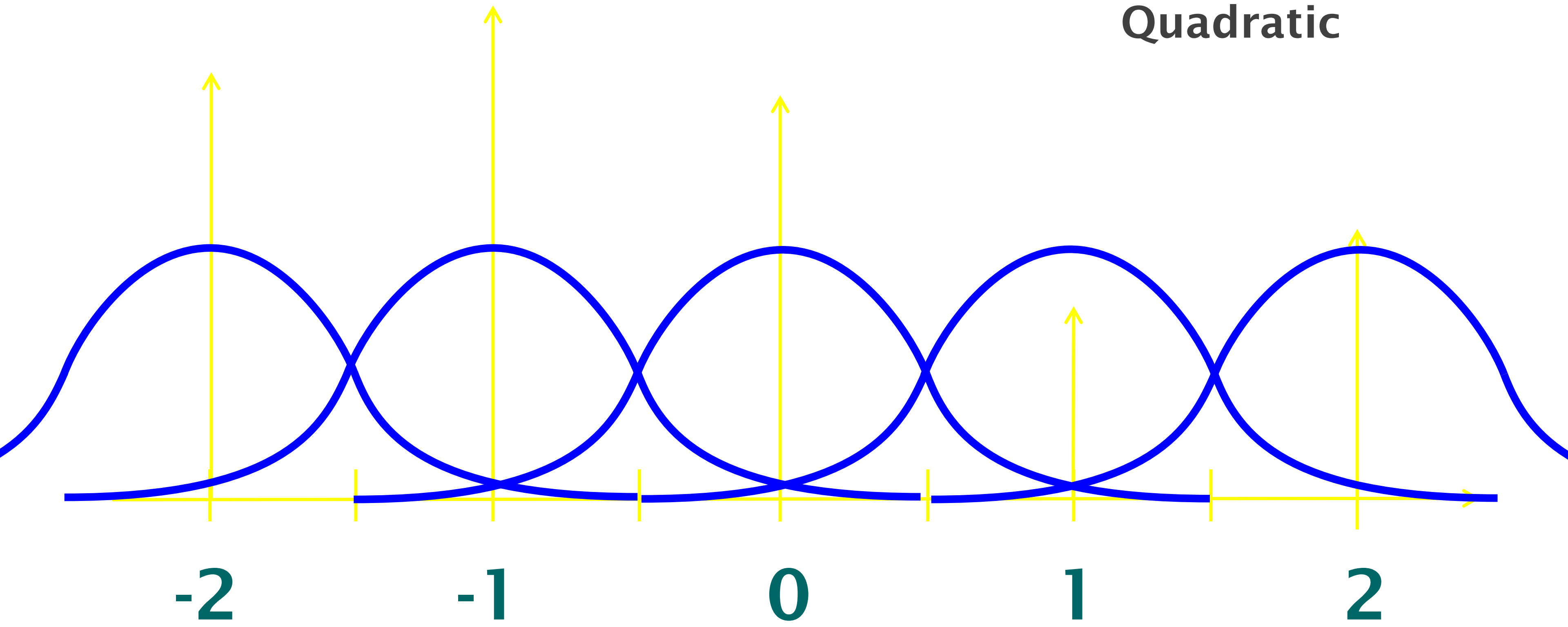
Interpolation



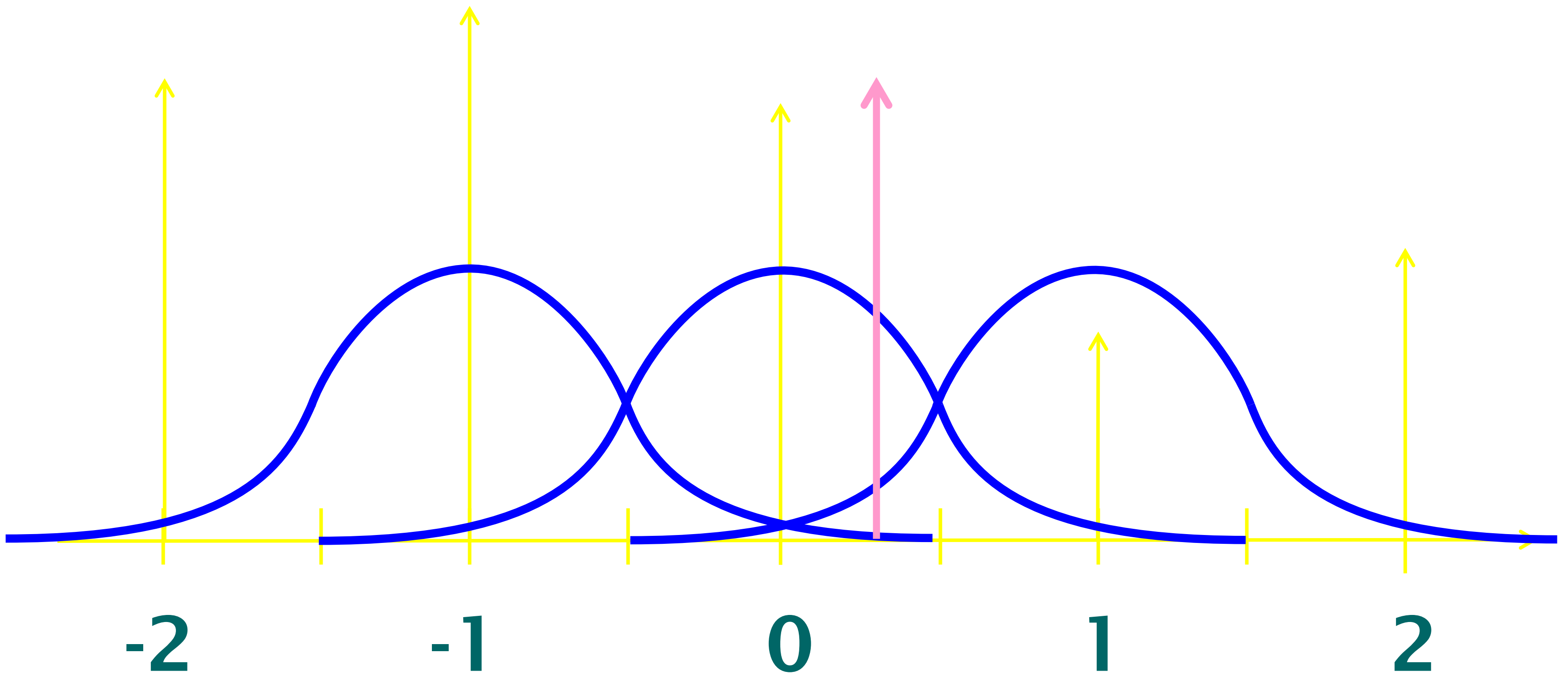
Interpolation

Second Order

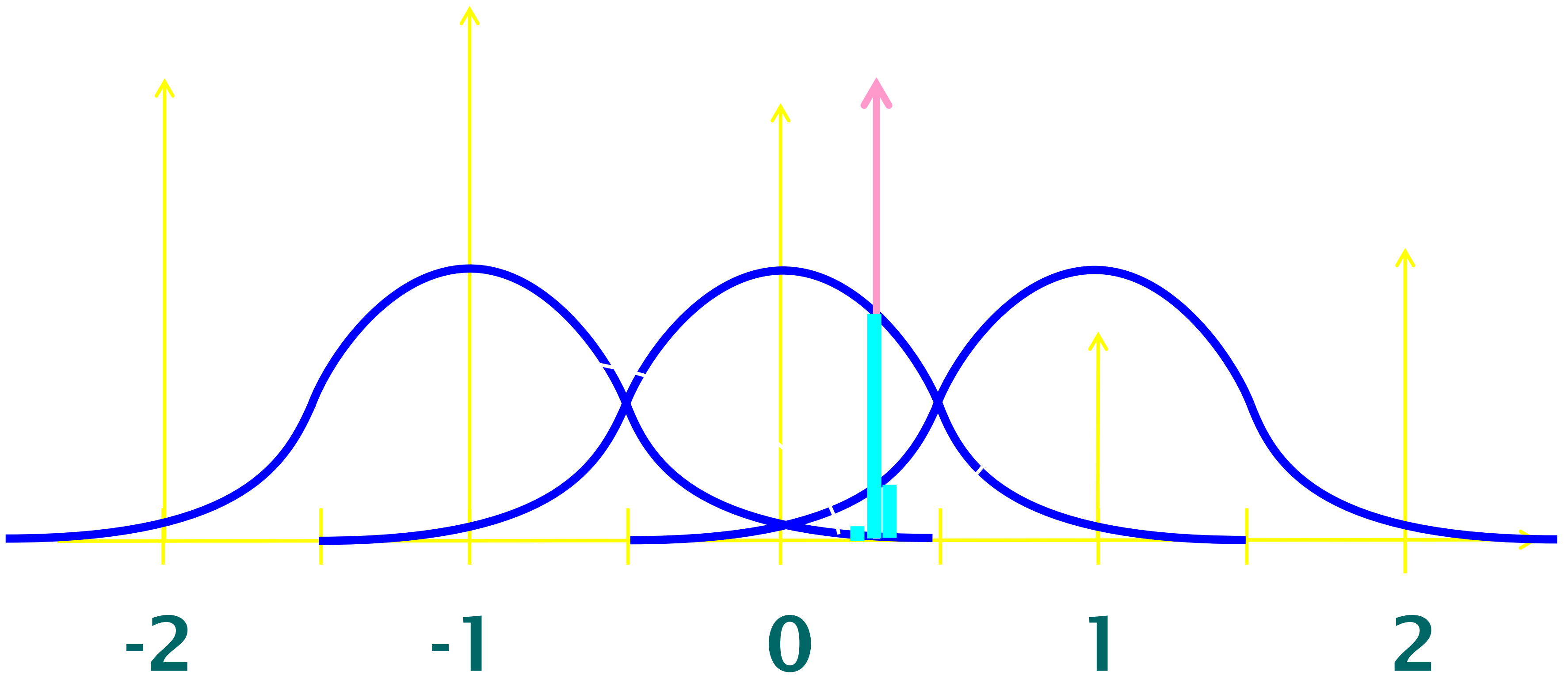
Quadratic



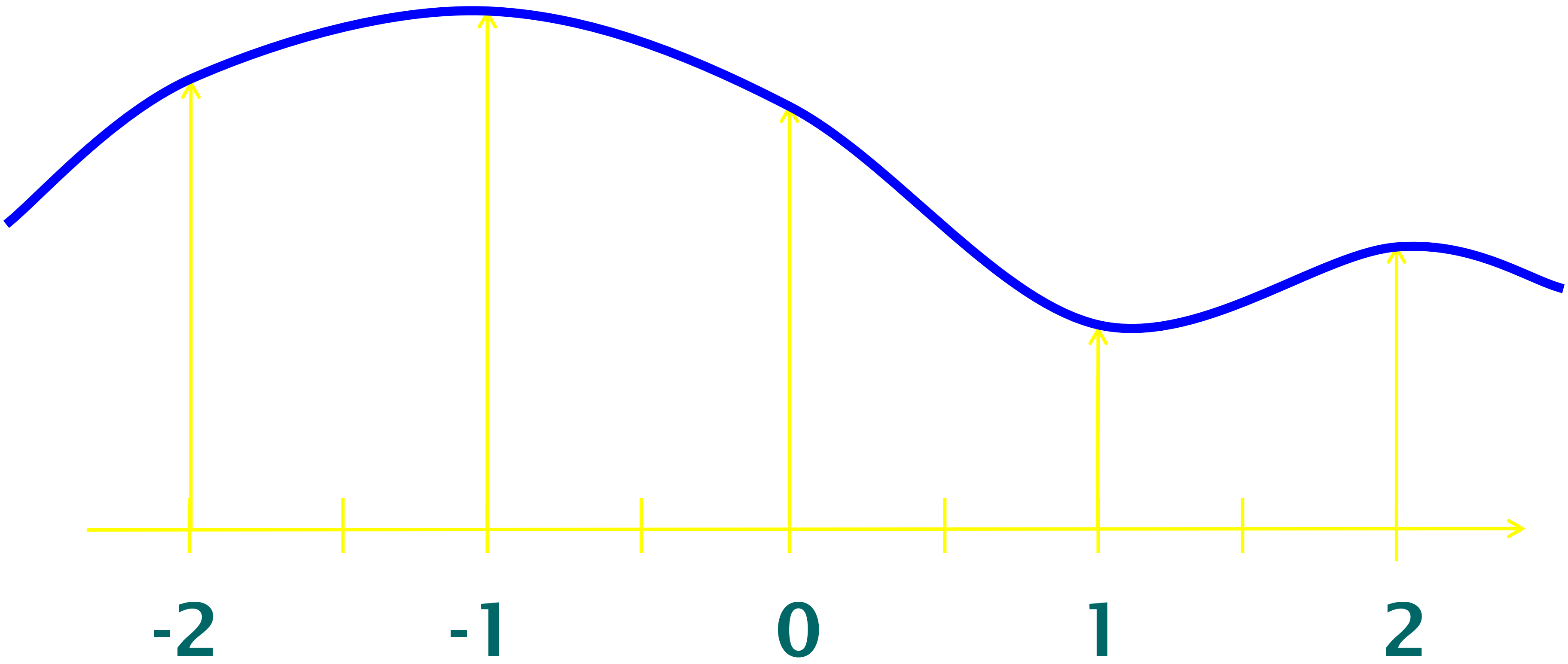
Interpolation



Interpolation



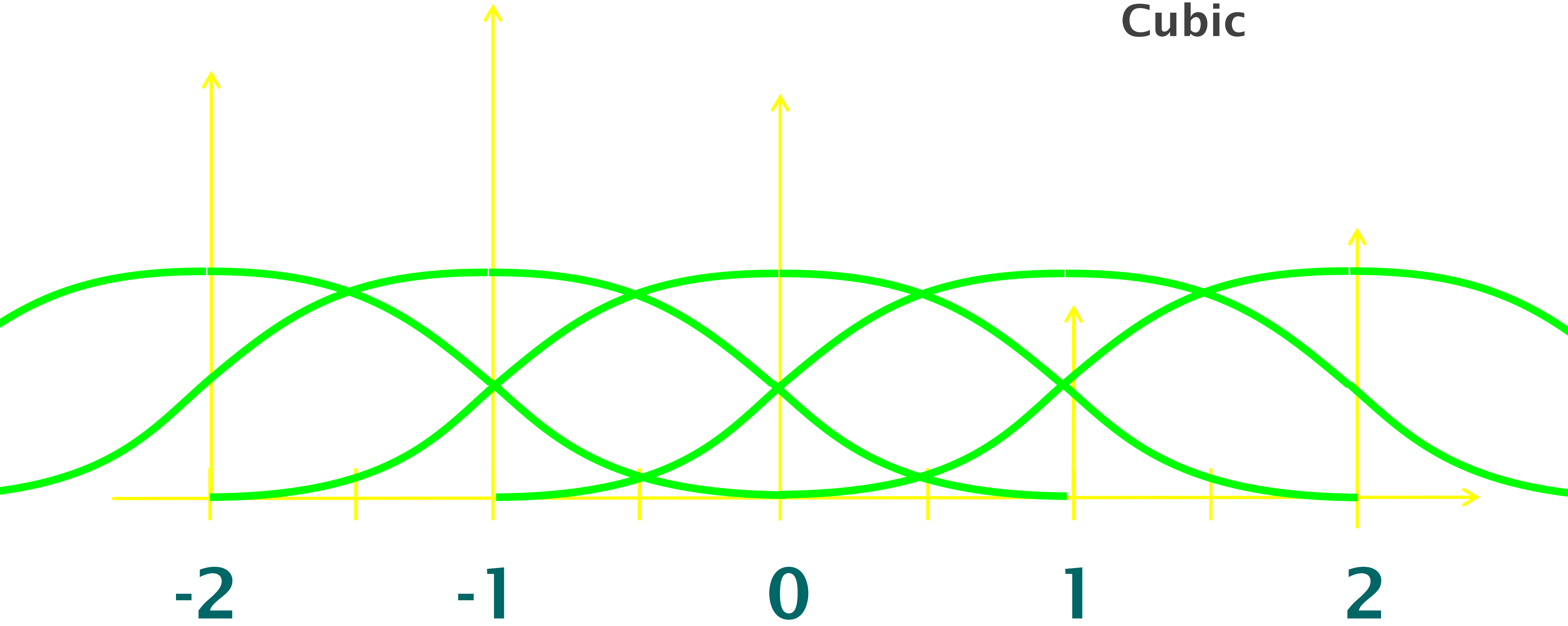
Interpolation



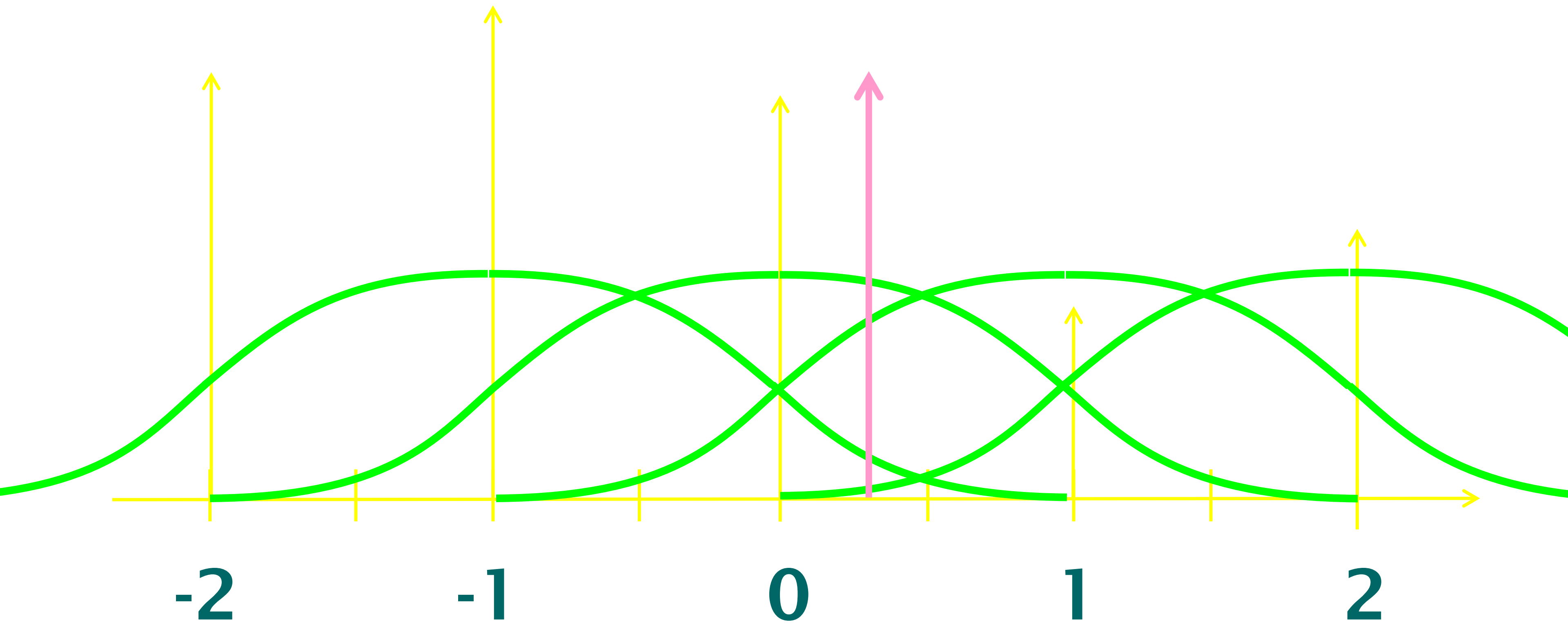
Interpolation

Third Order

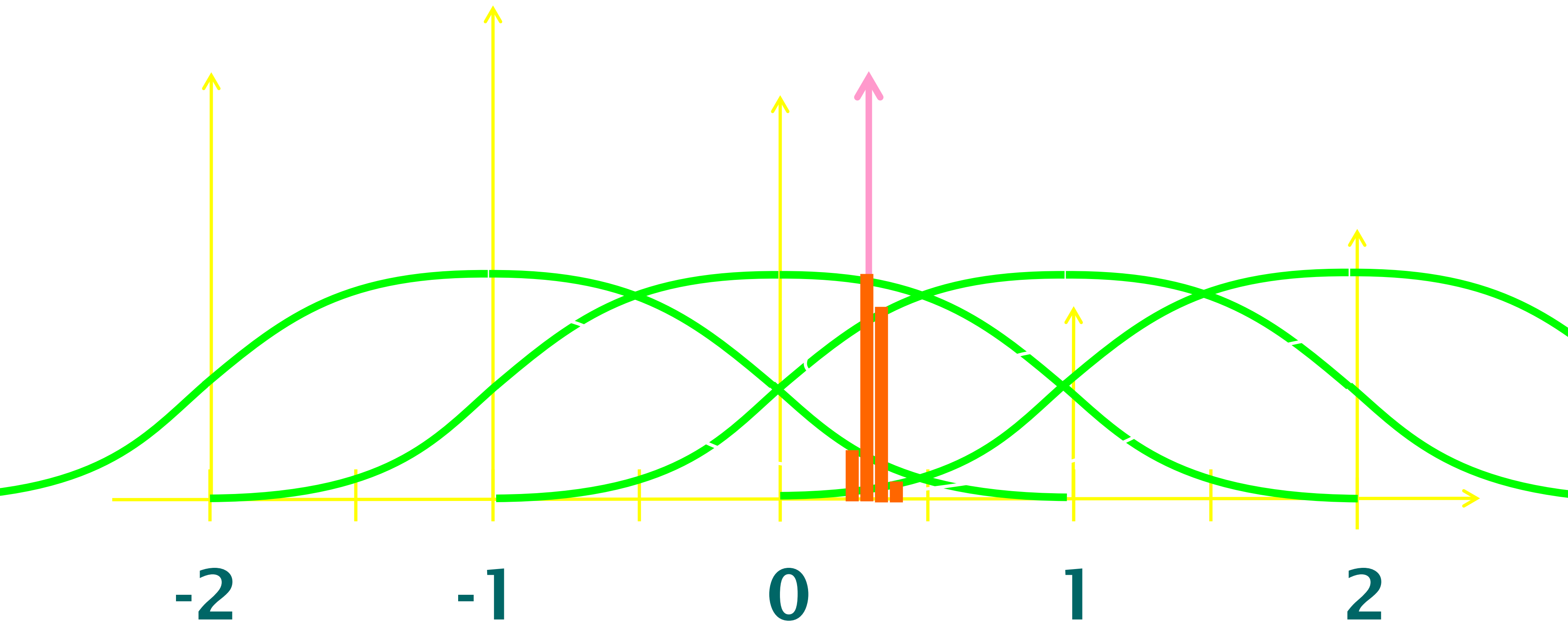
Cubic



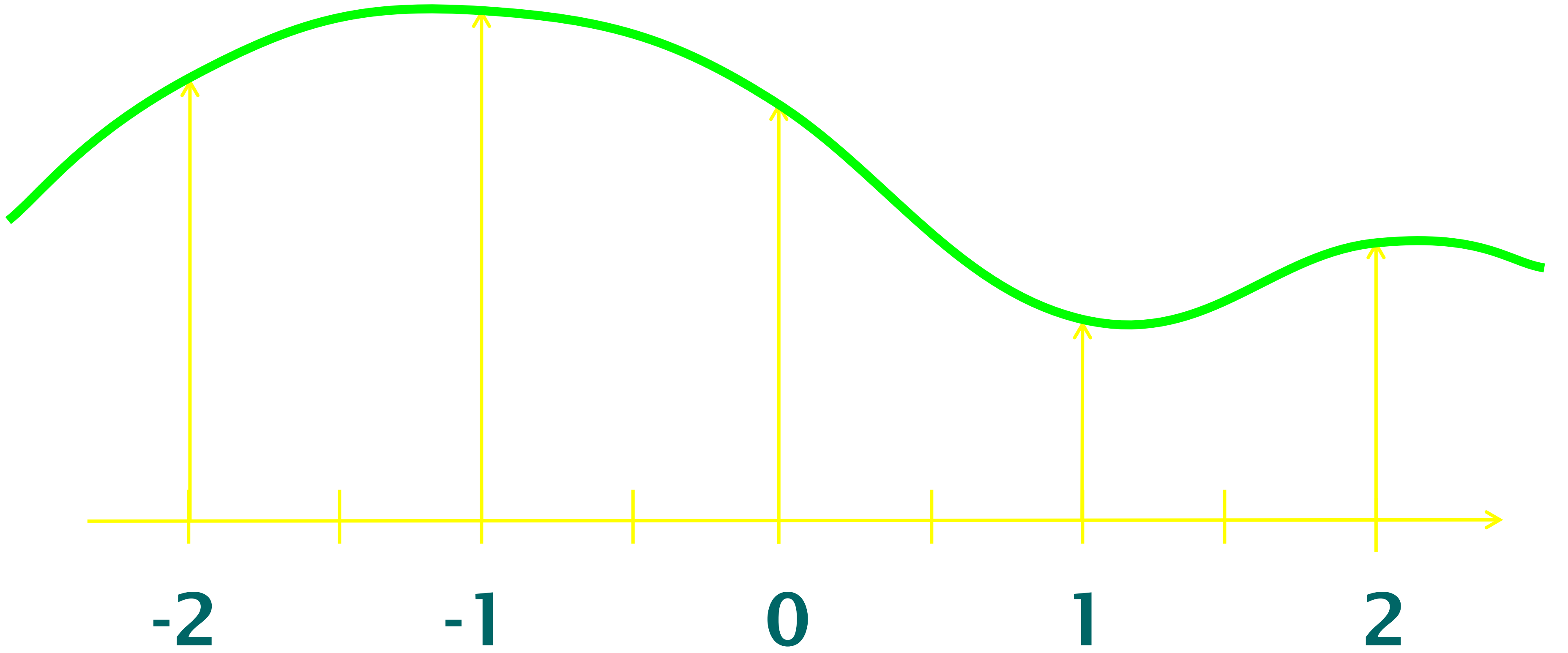
Interpolation



Interpolation



Interpolation



Type of Resampling	Computational Complexity
Nearest-Neighbor	$O(n^2)$
Bilinear Interpolation	$O(n^2)$
Cubic Convolution	$O(n^2)$
Cubic Spline, Direct Computation	$O(n^4)$
Cubic Spline, Using FFT	$O(n^3 \log n)$
Radial Functions with Local Support	$O(n^4)$
Gaussian, Using FFT	$O(n^3 \log n)$

ACCURACY EVALUATION

Localization error - displacement of features

- due to detection method

Matching error - false matches

- ensured by robust matching (hybrid)
- consistency check, cross-validation

Alignment error - difference between model and reality

- mean square error
- test point error (excluded points)
- comparison (“gold standard”)

TRENDS AND FUTURE

complex local transforms

multimodal data, 3D data sets

brute force approaches

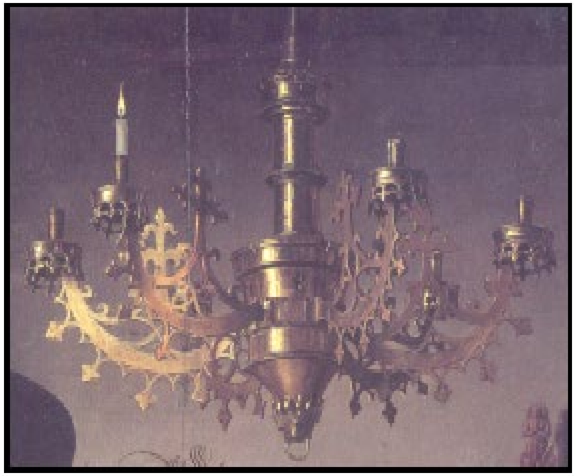
CNN

expert systems

Analysis of the Arnolfini Portrait

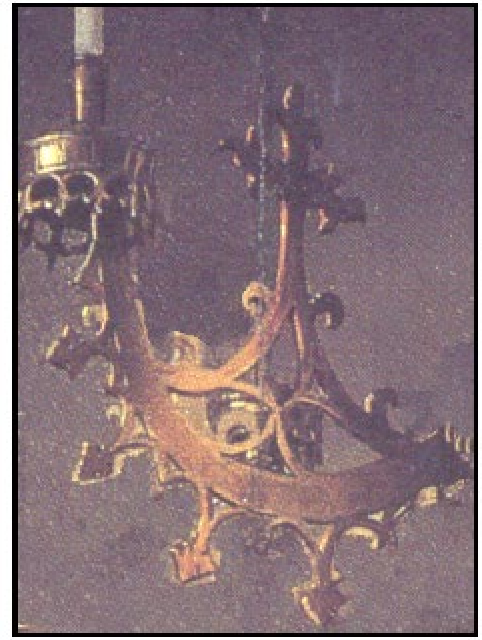
(Jan van Eyck)

From Criminisi et al., Microsoft Research



a

b



c

d

e

APPLICATIONS DIFFERENT TIMES

Medieval mosaic conservation, Prague

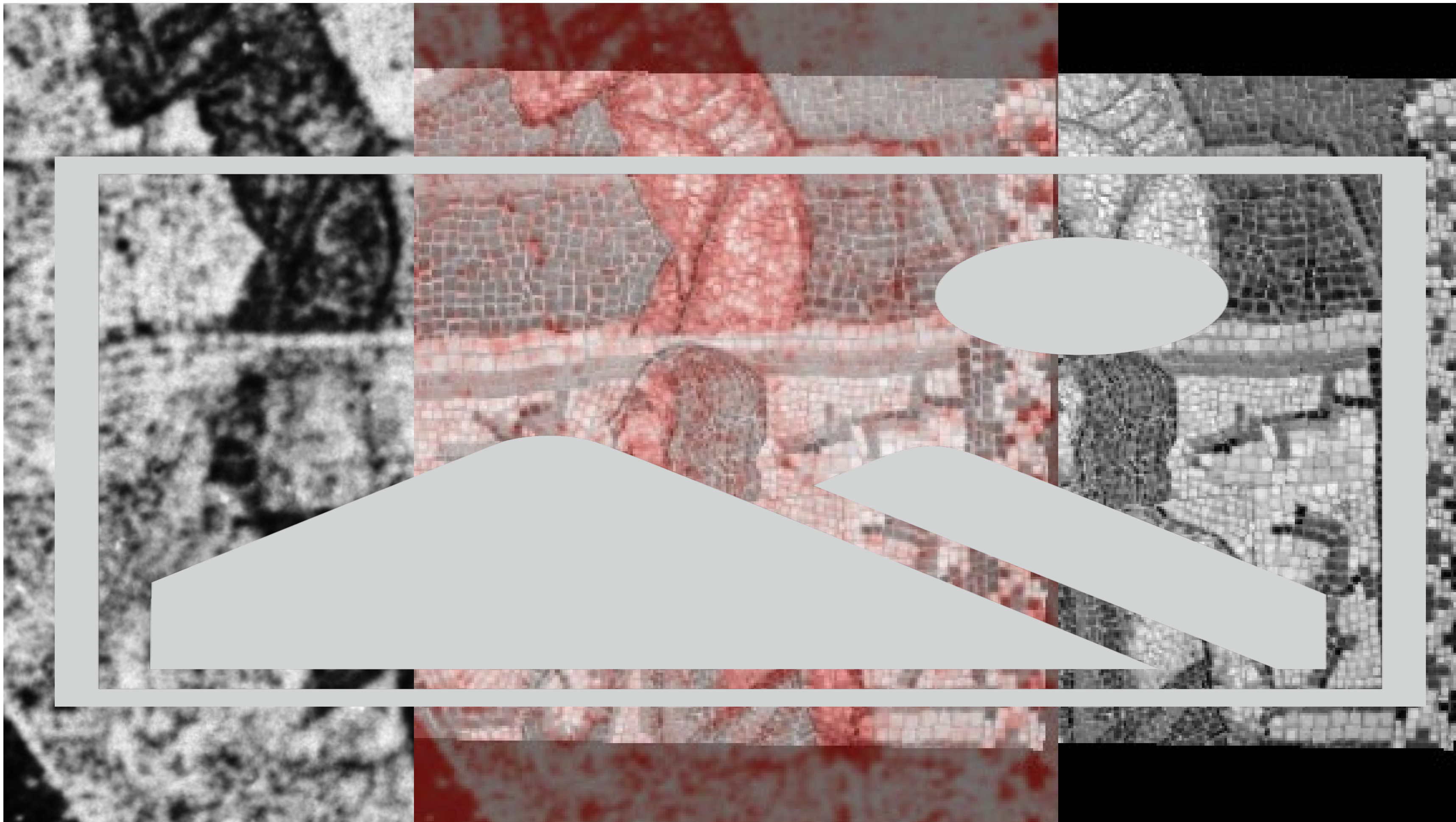
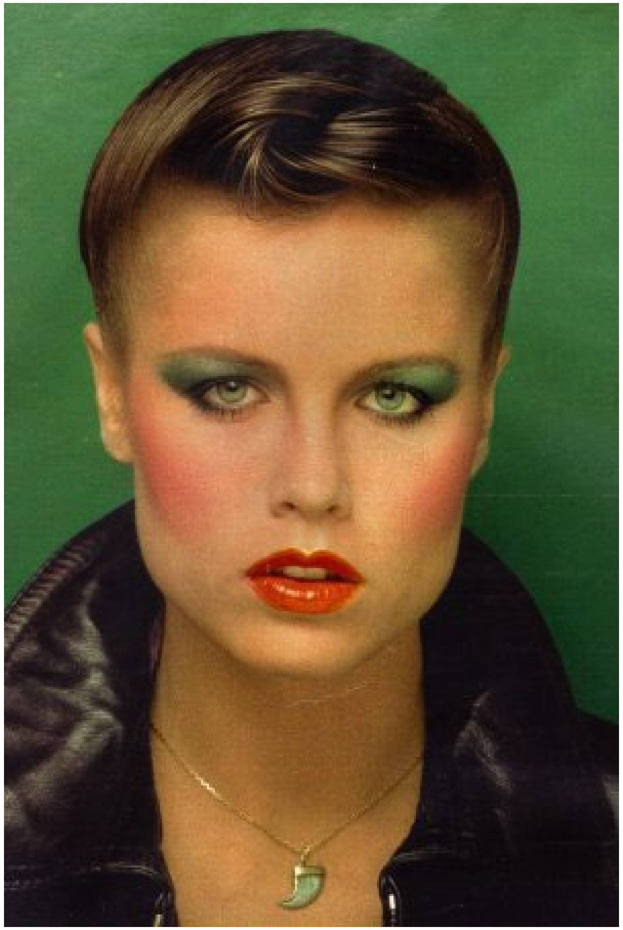
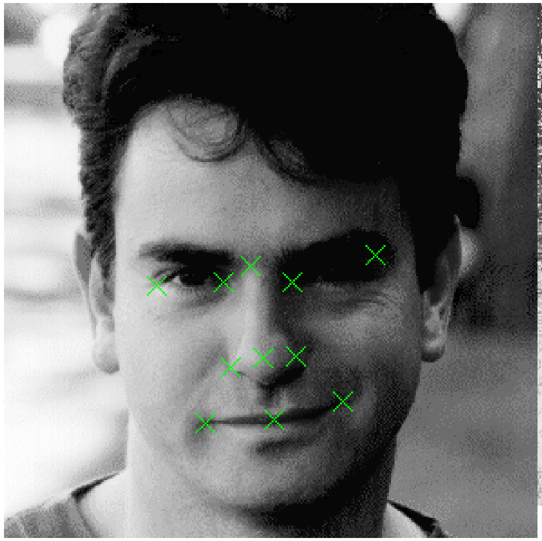
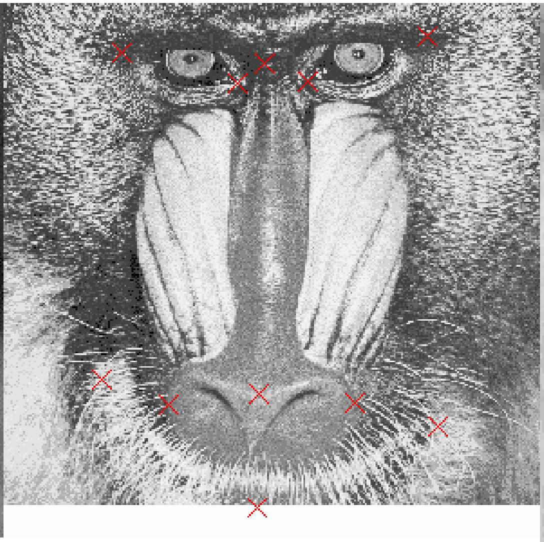


Image warping



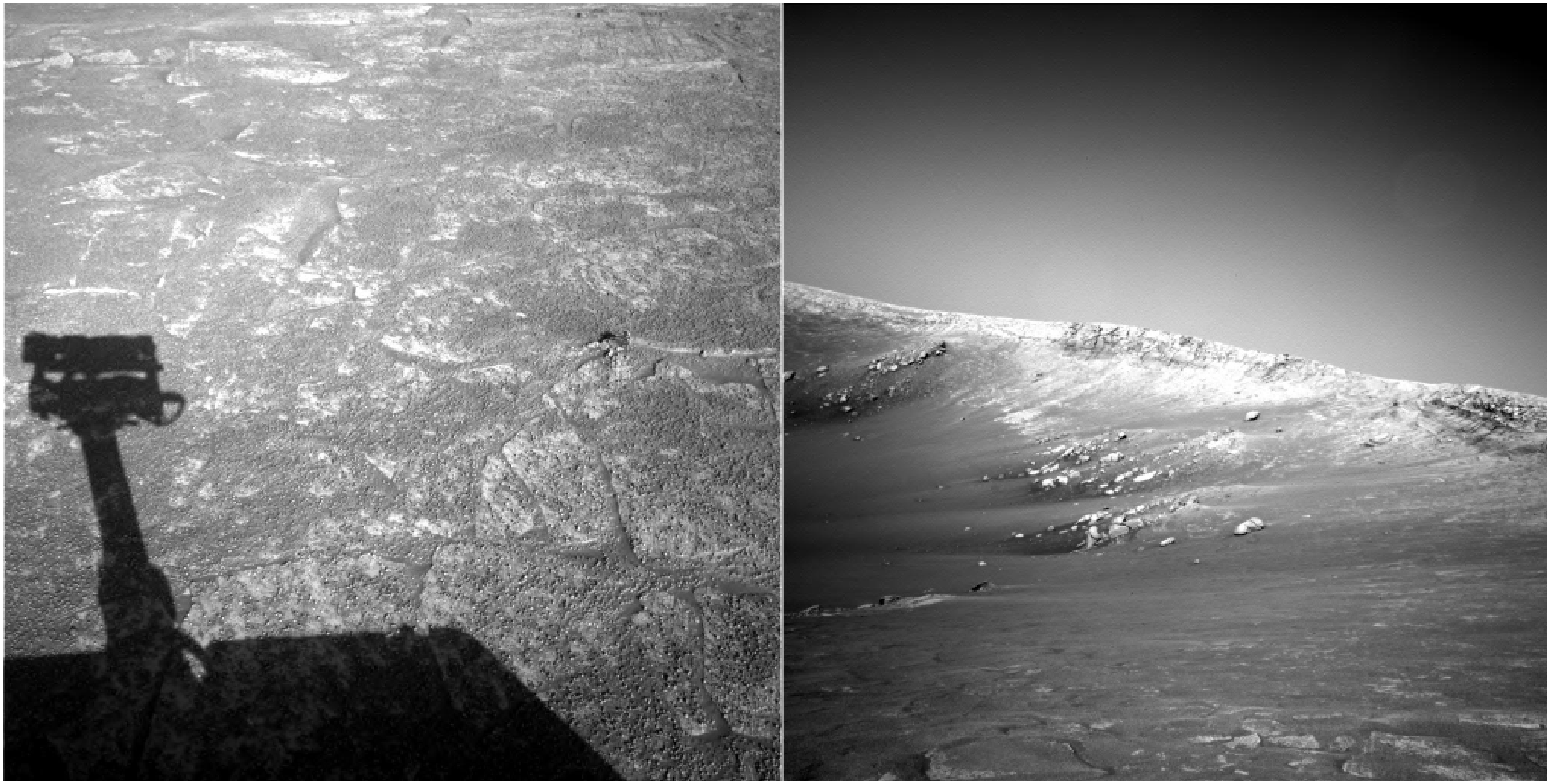
Human to animal warping by TPS

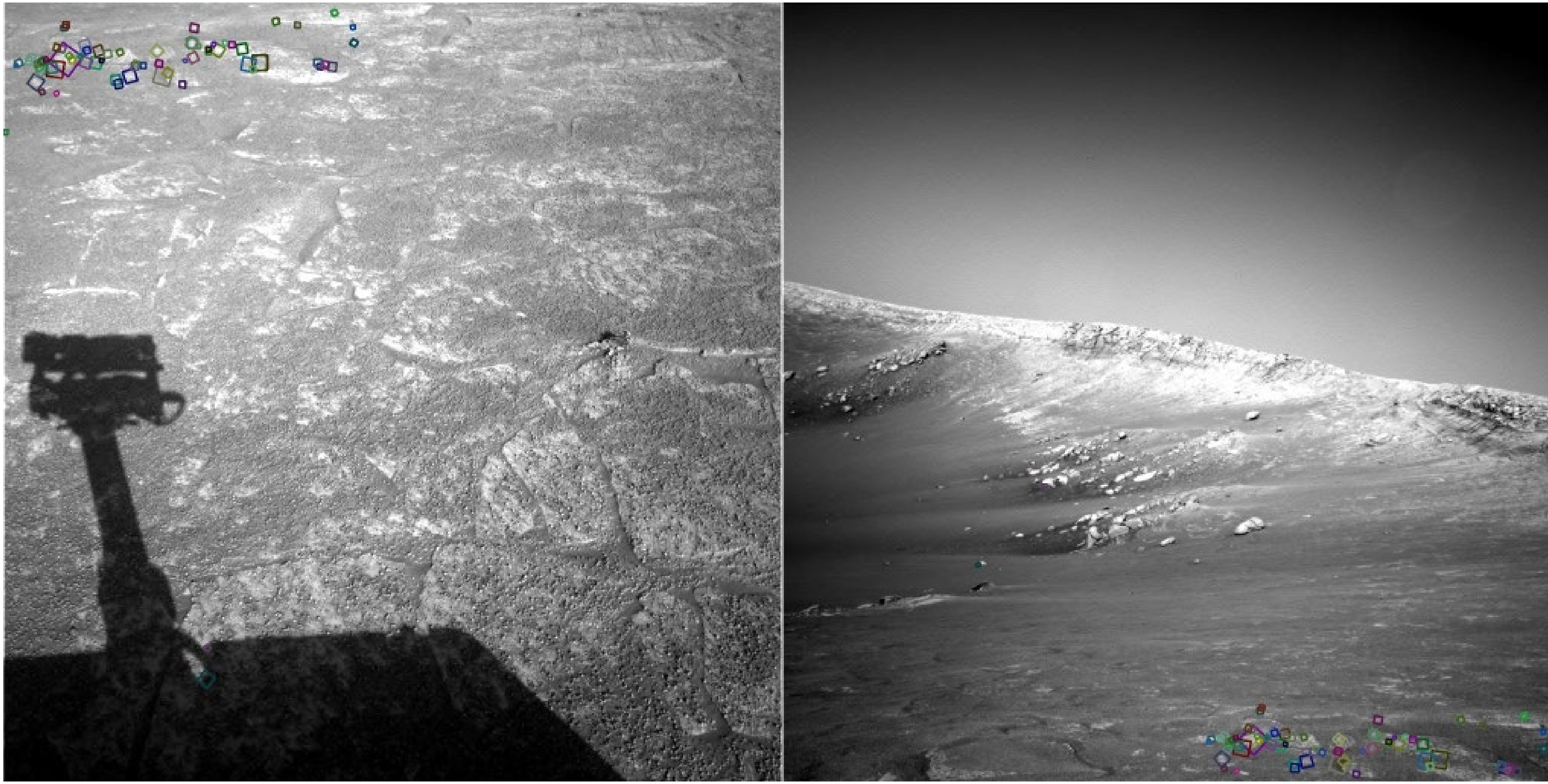




Realistic warping

Morphing = warping + blending





SIFT – scale invariant feature transform

Distinctive image features from scale-invariant keypoints. David G. Lowe, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

Hodně příznaků

Opakovatelné

Reprezentativní (orientace, měřítko)

Rychlý výpočet

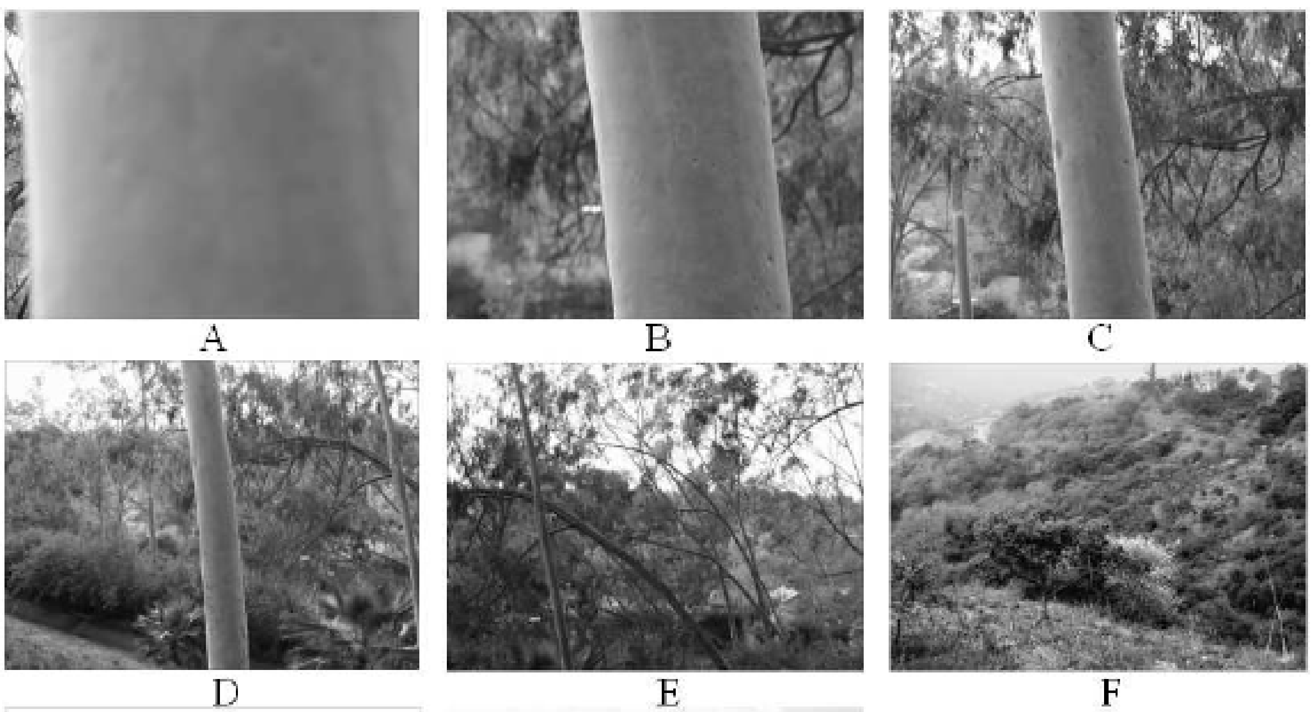
SIFT – scale invariant feature transform

1. Nalézt extrémy
2. Vylepšit počet a polohu
3. Odstranit vliv otočení a změny měřítka
4. Popsat

SIFT – scale invariant feature transform

1. Nalézt extrémy

Příznaky různé v různých rozlišeních - scale



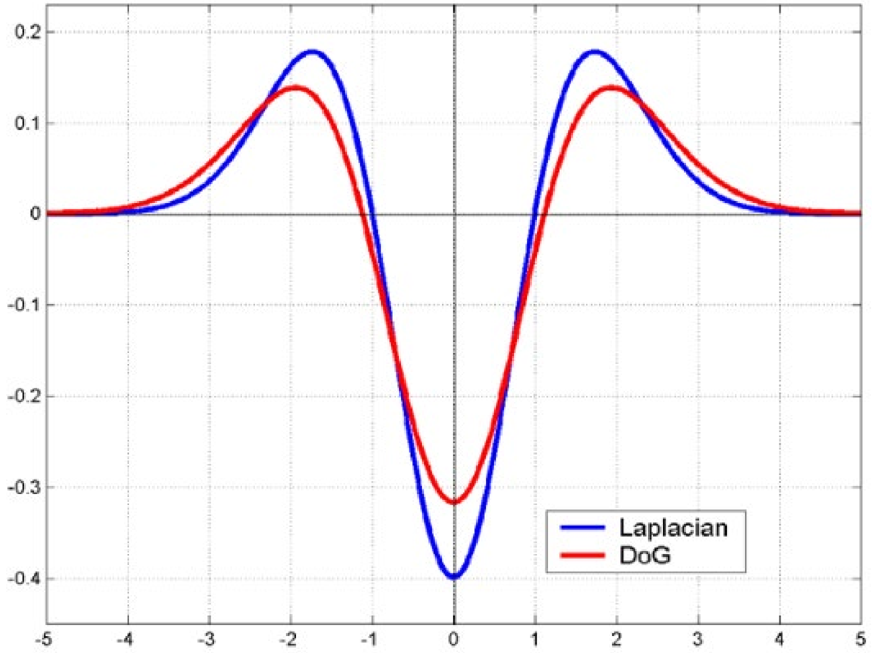
SIFT – scale invariant feature transform

1. Nalézt extrémy

scale

LoG

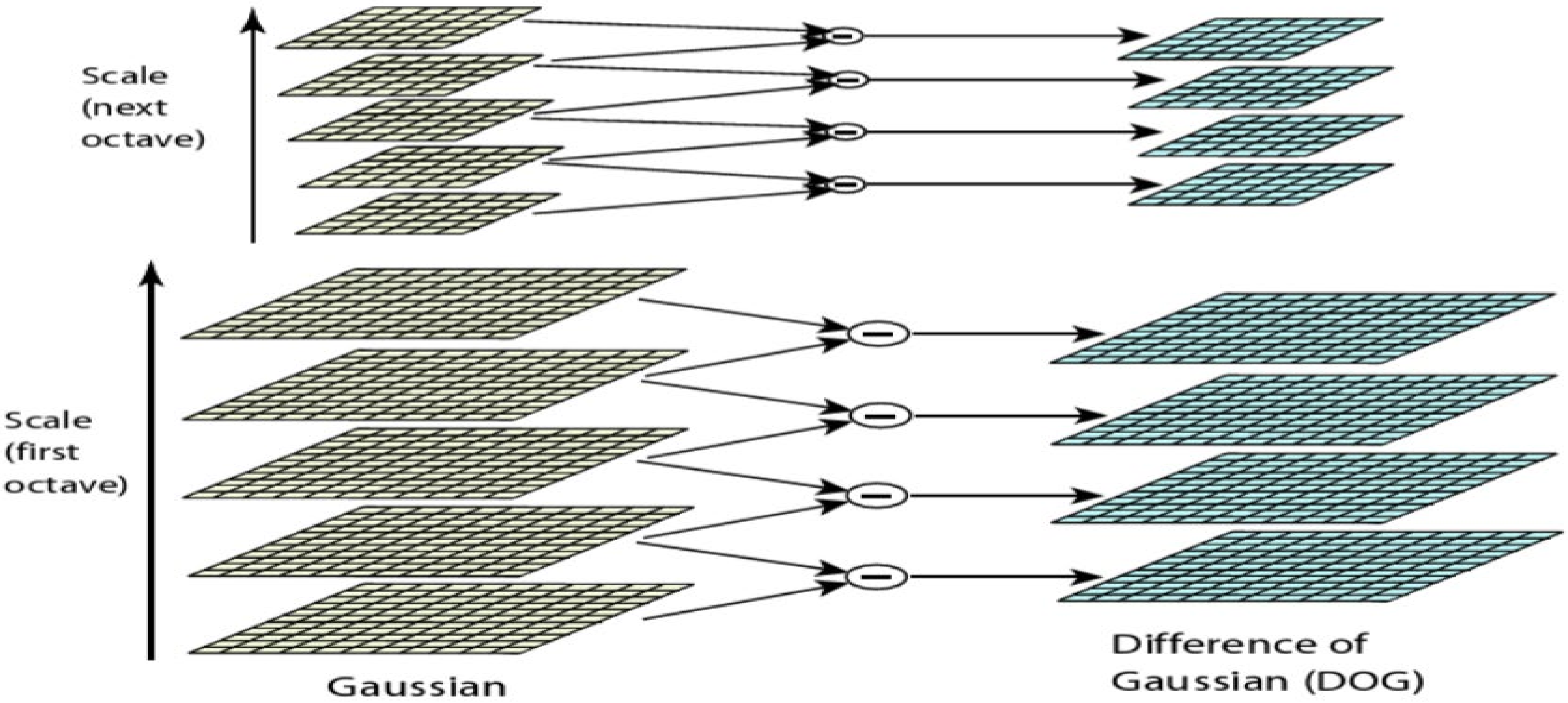
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



DoG - difference of Gaussians

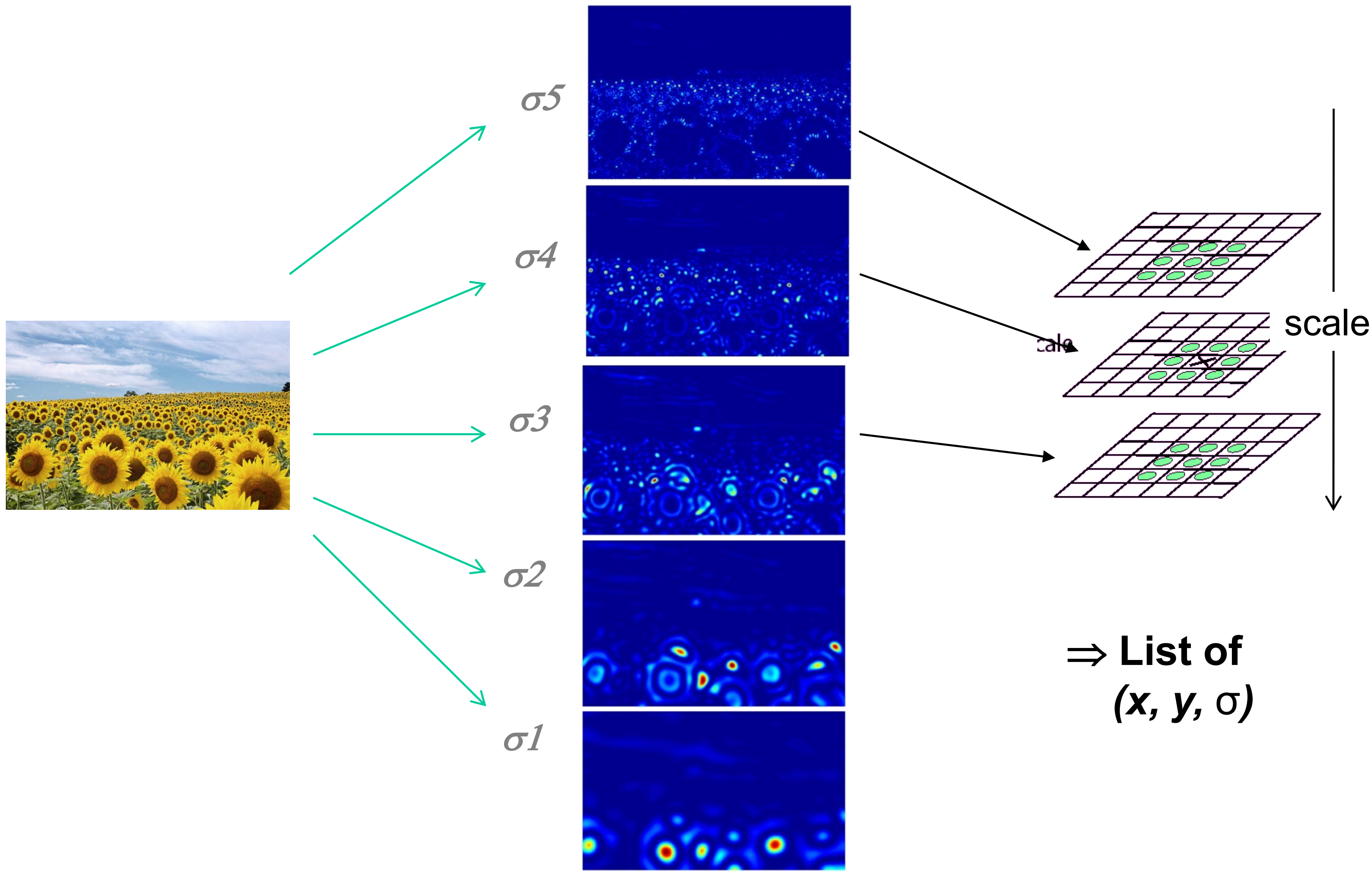
$$D(\sigma) \equiv \left(G(k\sigma) - G(\sigma)\right) * I$$

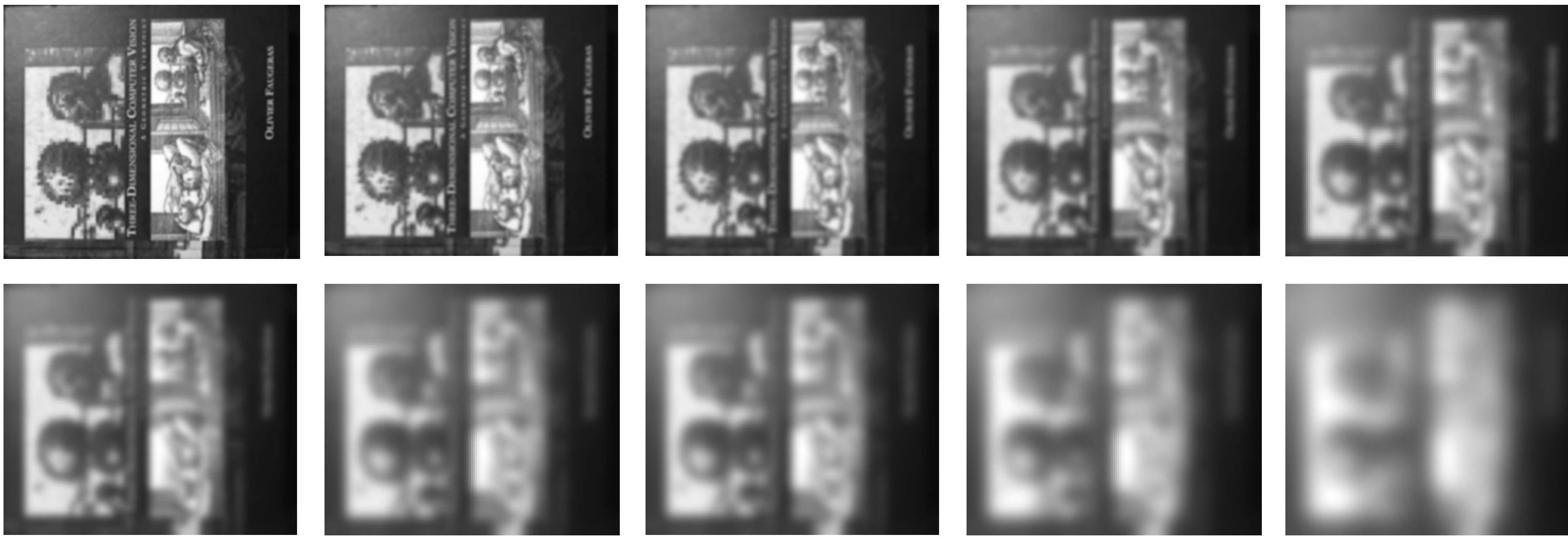
SIFT – v jedné oktávě 3 škály (Lowe)



Všechny lokální extrémy na 3x3x3 okolí

Non-maximum suppression

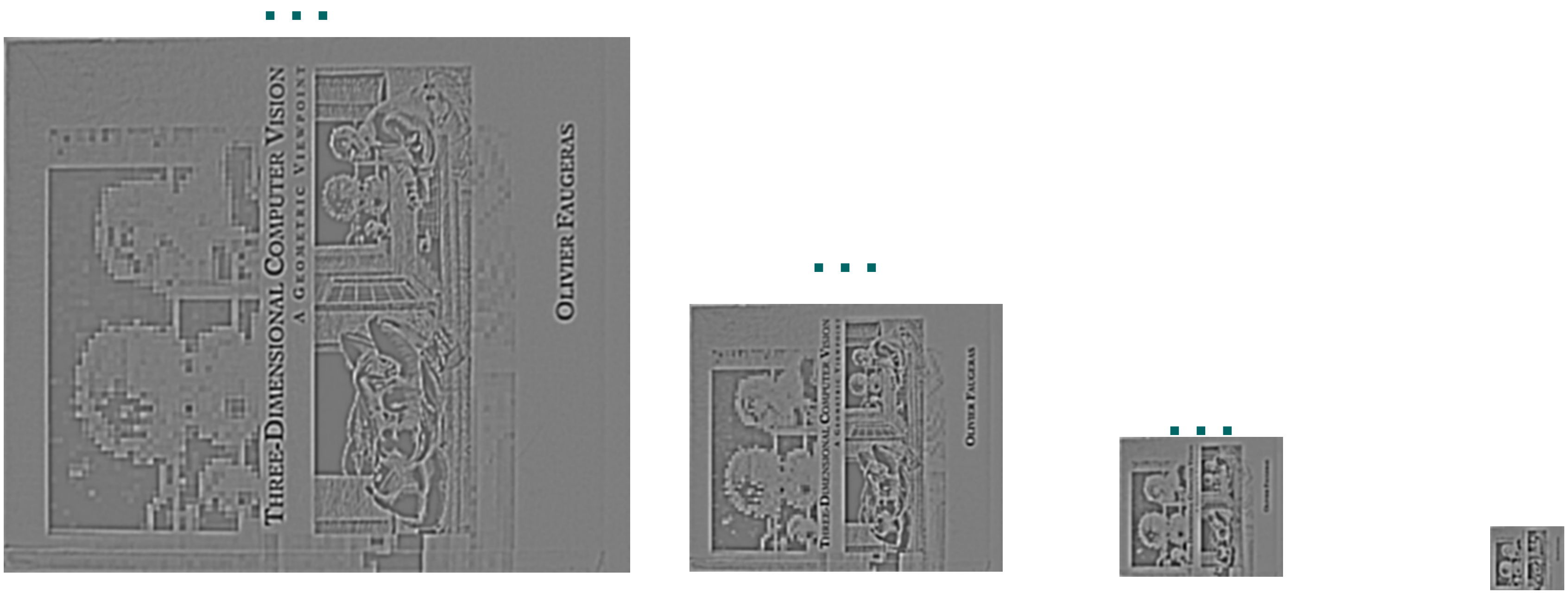

σ5
σ4
σ3
σ2
σ1
scale
⇒ List of
(x, y, σ)


Three-Dimensional Computer Vision
A Geometric Viewpoint
Olivier Faugeras

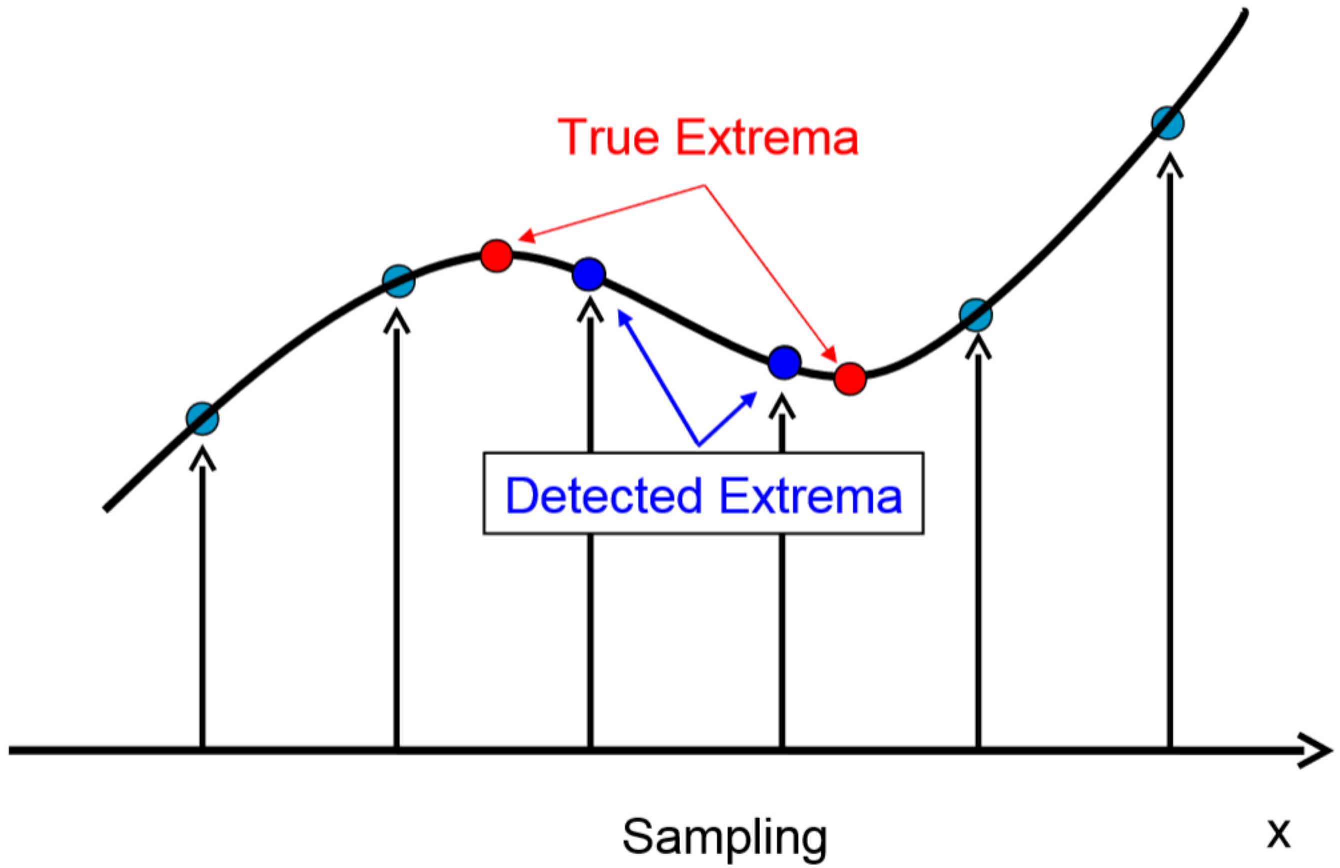

THREE-DIMENSIONAL COMPUTER VISION
A GEOMETRIC VIEWPOINT
OLIVIER FAUGERAS

Scale space images

Difference-of-Gaussian images

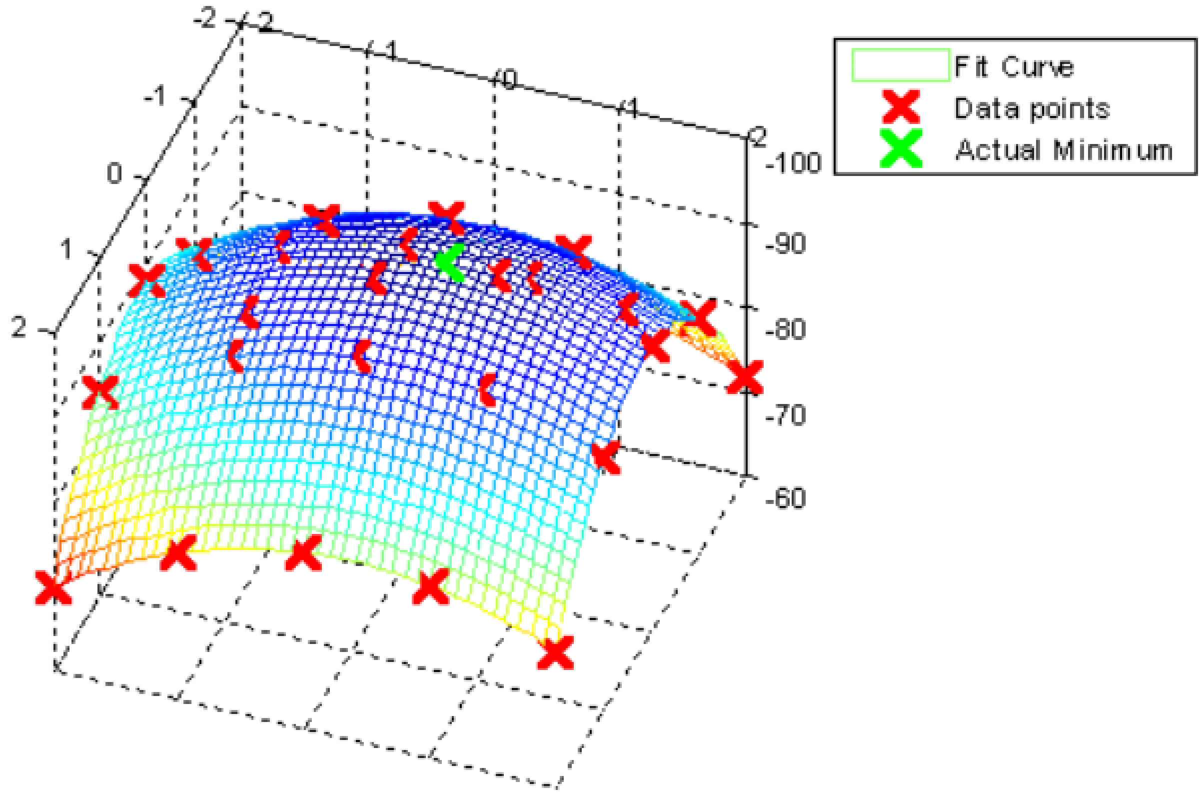


SIFT – scale invariant feature transform



interpolace

SIFT – scale invariant feature transform



interpolace

SIFT – scale invariant feature transform

Čištění kandidátů - ~~nízký kontrast~~

- ~~extrémy co jsou hrany~~

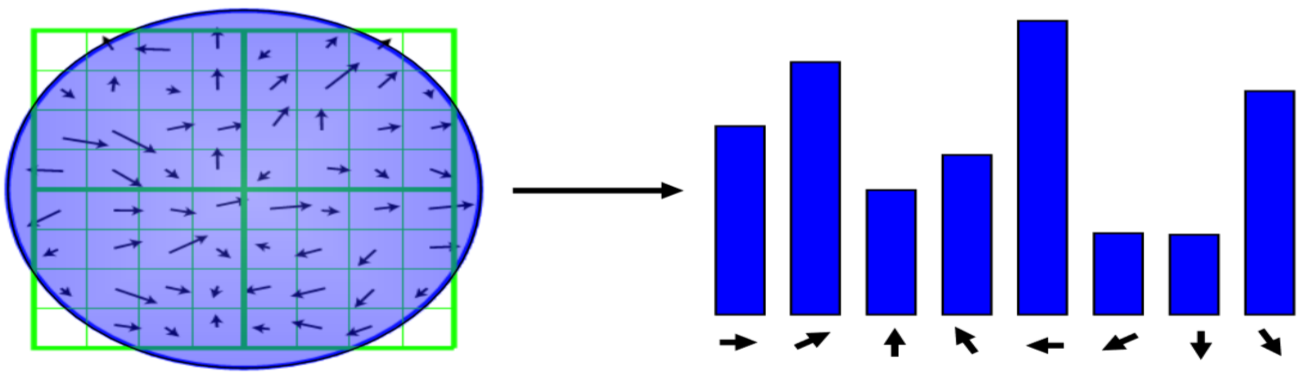
(nízká křivost)

832 -> 729 -> 536

SIFT – scale invariant feature transform

Deskriptory – invariance k affinní, šum, osvětlení

- rychlé, distinktivní

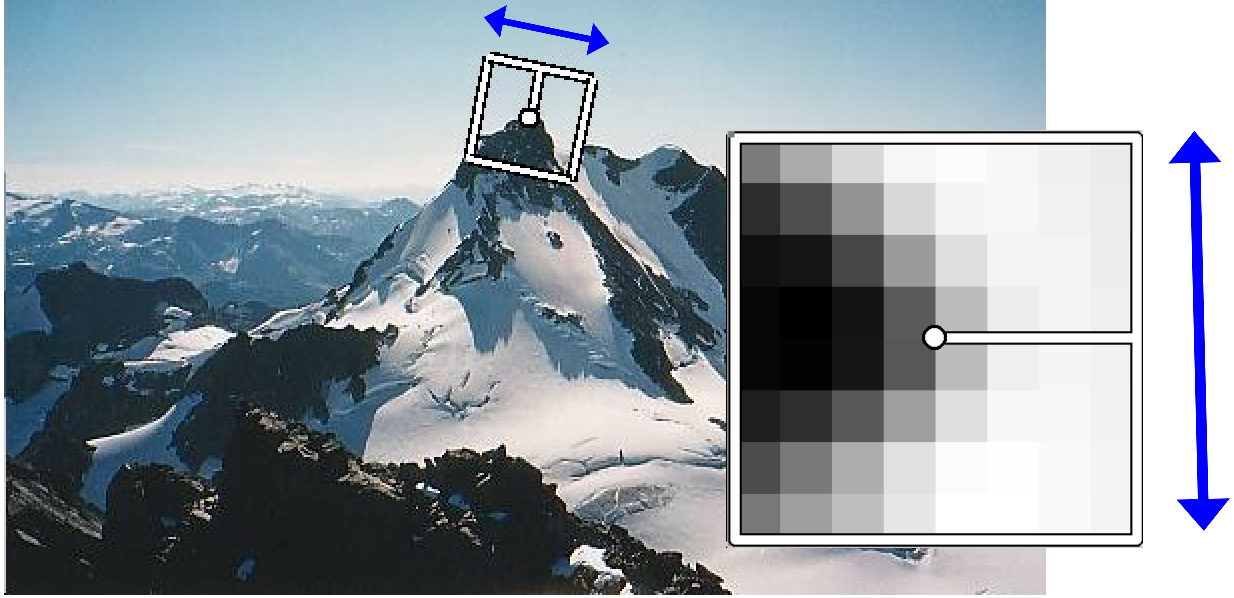


Normalizace škála a rotace

odhad velikosti gradientu a směru ->

pro nejvýznamnější peak - histogram gradientů (36 binů)

pokud hodne podobne maximu -> další feature



SIFT – scale invariant feature transform

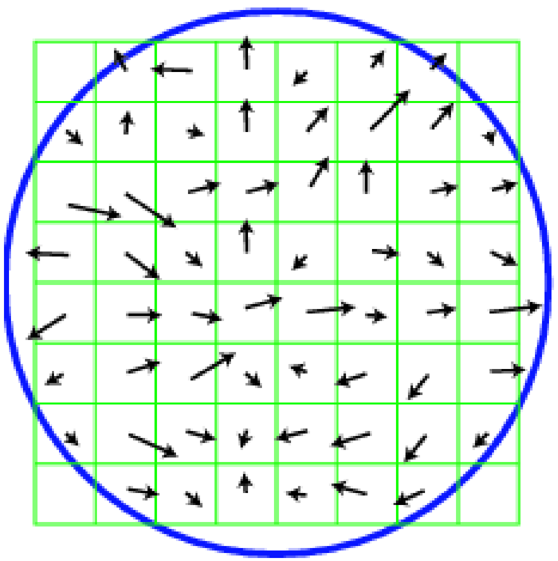
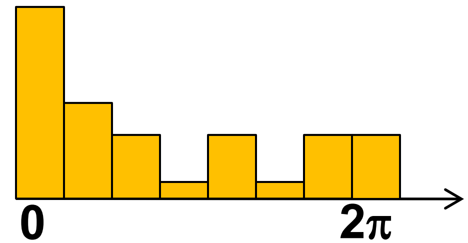
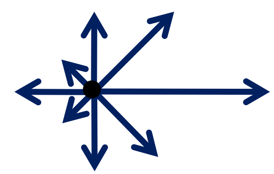


Image gradients



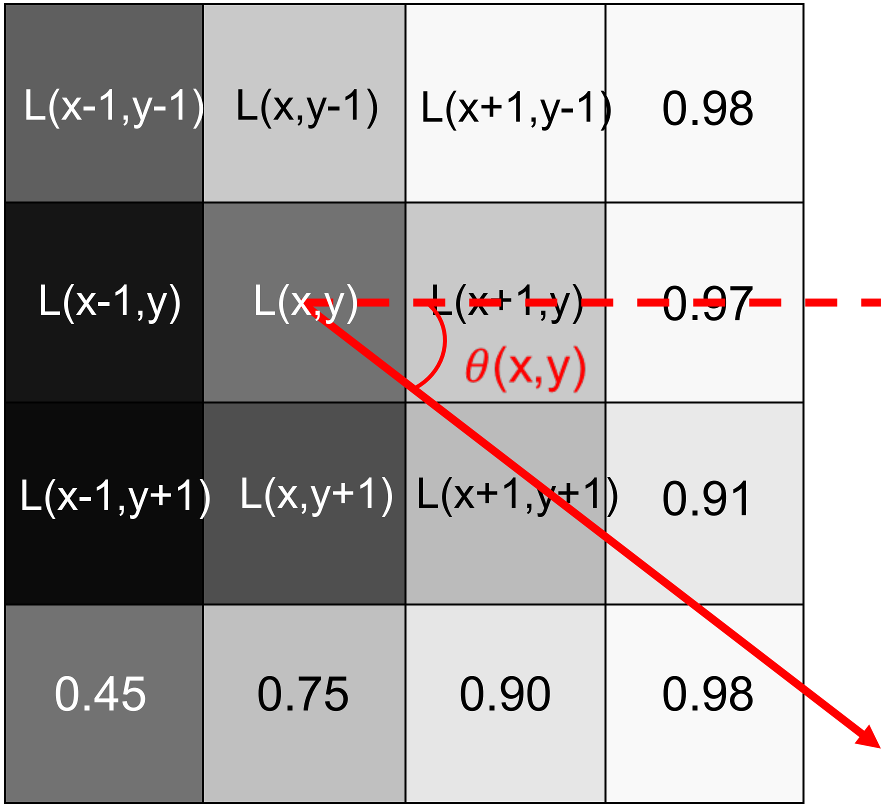
angle histogram



Numeric Example

0.37	0.79	0.97	0.98
0.08	0.45	0.79	0.97
0.04	0.31	0.73	0.91
0.45	0.75	0.90	0.98

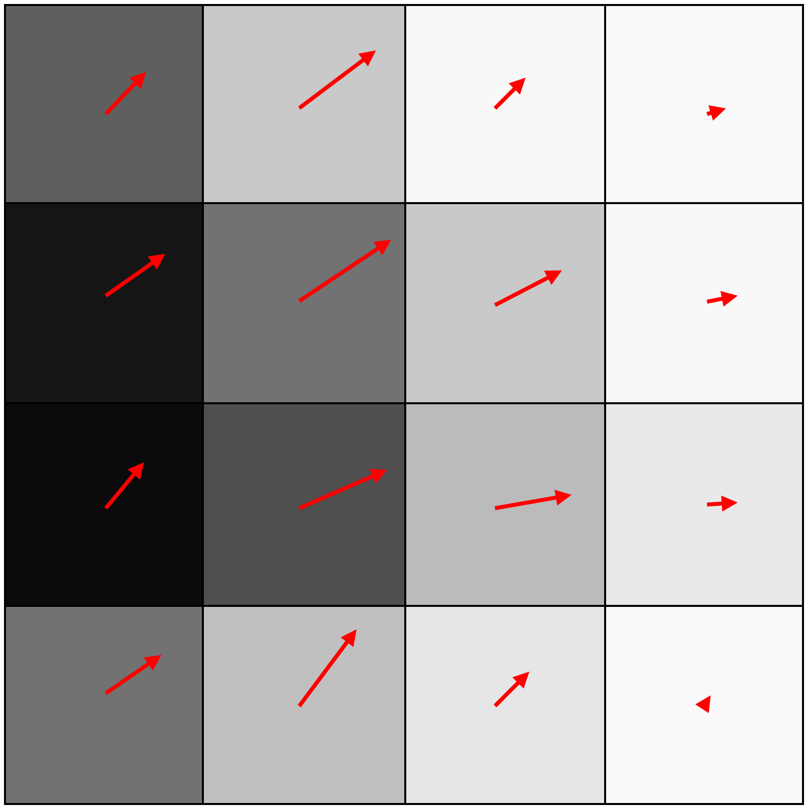
by Yao Lu



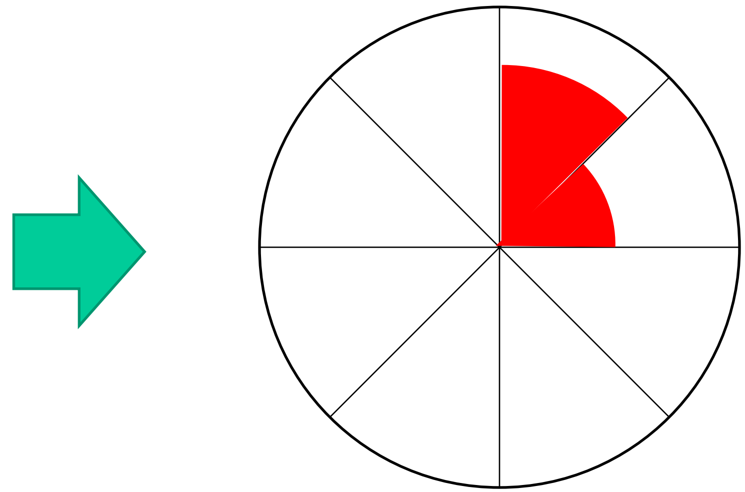
$$\text{magnitude}(x,y)=\sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2+\left(L(x,y+1)-L(x,y-1)\right)^2}$$

$$\theta(x,y)=atan\left(\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)\right)$$

by Yao Lu



Orientations in each of the 16 pixels of the cell

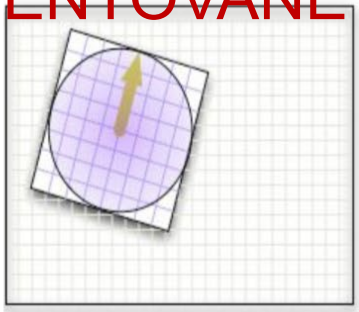


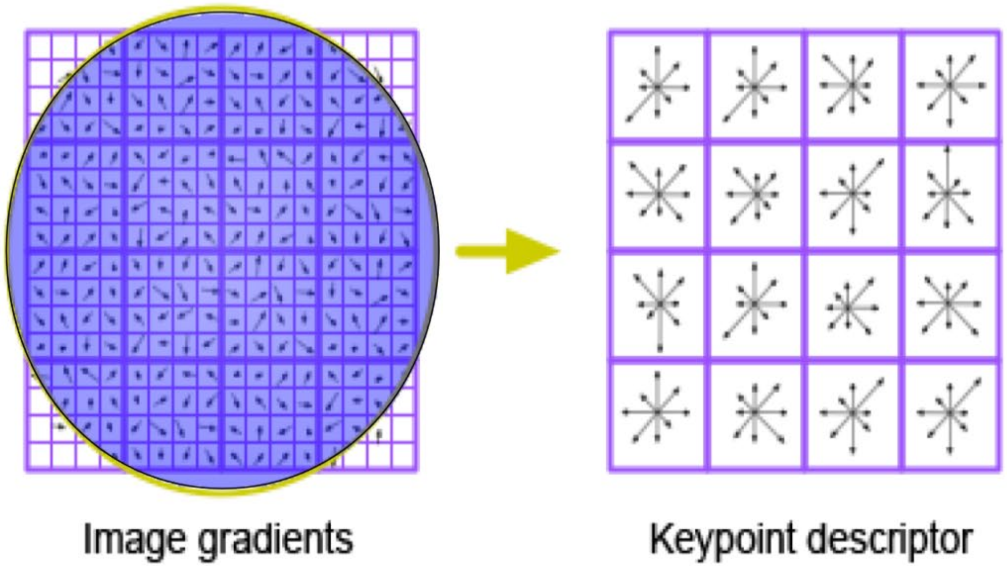
The orientations all ended up in two bins: 11 in one bin, 5 in the other. (rough count)

5 11 0 0 0 0 0 0

SIFT – scale invariant feature transform

ORIENTOVANÉ !





určí orientaci a velikost v nejbližším scale pro max peak

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Histogram orientací

16x16 - 4x4 okna, 8 směrů -> 128 vektor příznaků


10
PACK
maxell
CD-R

Image Fusion

Input: Several images of the same scene

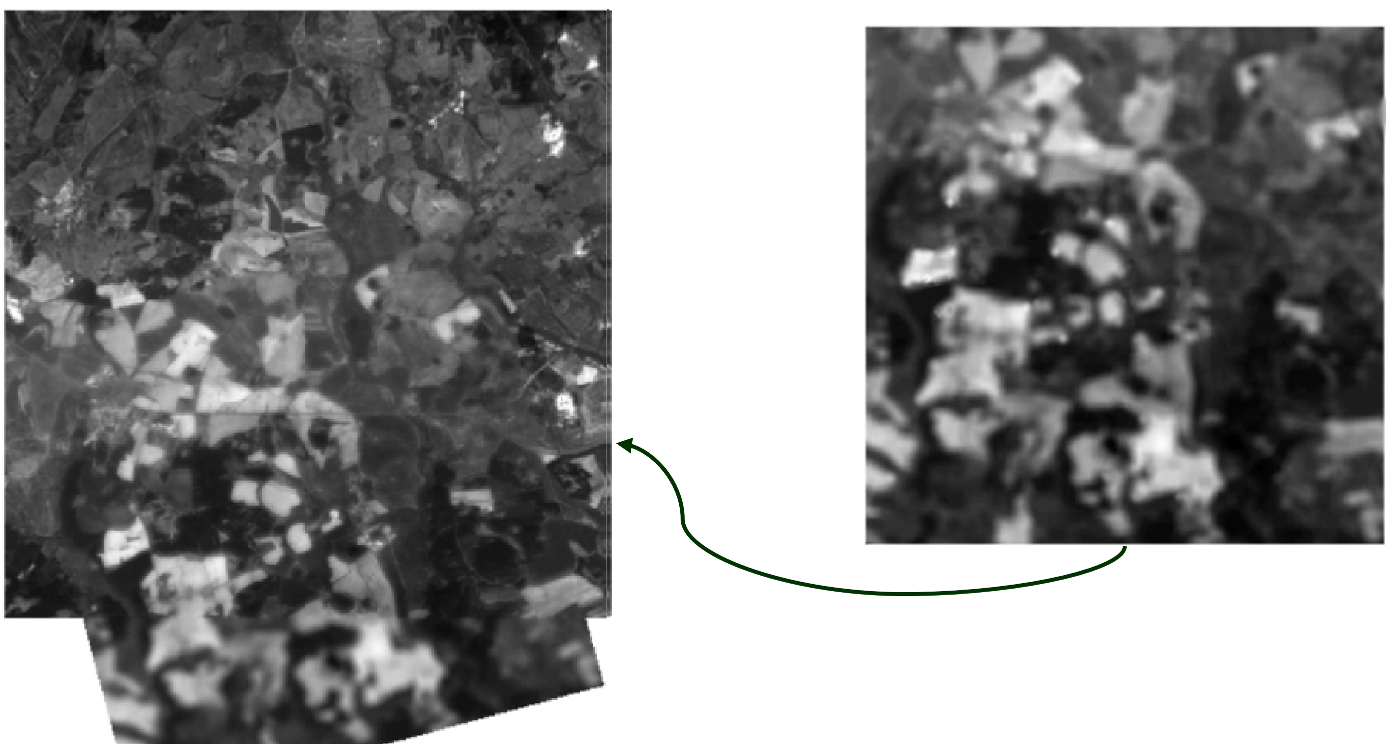
Output: One image of higher quality

The definition of “quality” depends on the particular application area

Basic fusion strategy

Acquisition of different images

Image-to-image registration



Fusion categories

Multisetting fusion

Multiview fusion

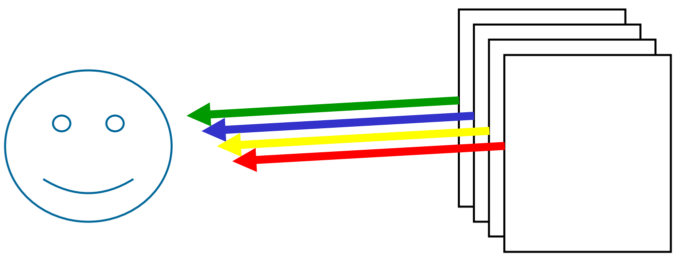
Multitemporal fusion

Multimodal fusion

Fusion for image restoration

Multisetting Fusion

Images taken by the same sensor with different settings



Goal: To combine complementary information by image compositing

Multifocus fusion

Multifocus fusion

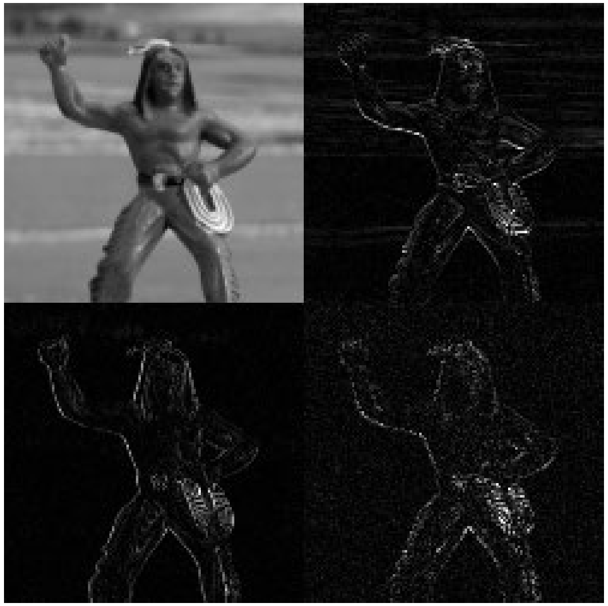
The original image can be divided into regions such that every region is in focus in at least one channel

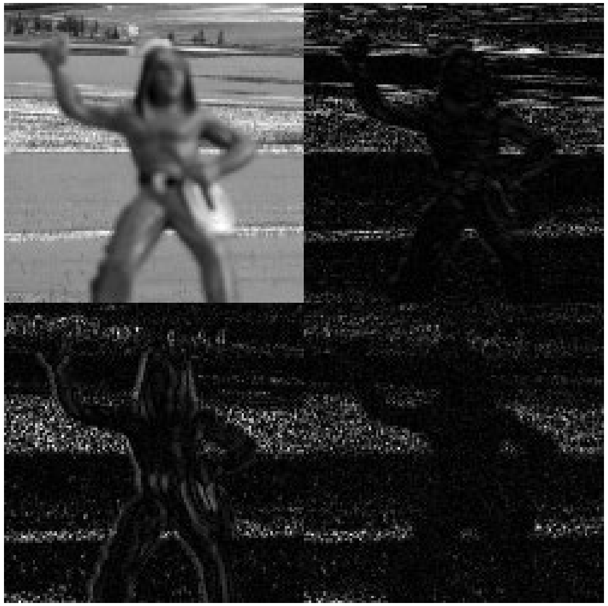
Goal:Image everywhere in focus

Idea: Identify the regions in focus (by maximizing proper focus measure) and combine them together

Artificial example

Images with different areas in focus





Fused image

Multiexposure fusion

high dynamic range images

foreground

background

Courtesy of Image Fusion Systems Research

Fusion categories

Multisetting fusion

Multiview fusion

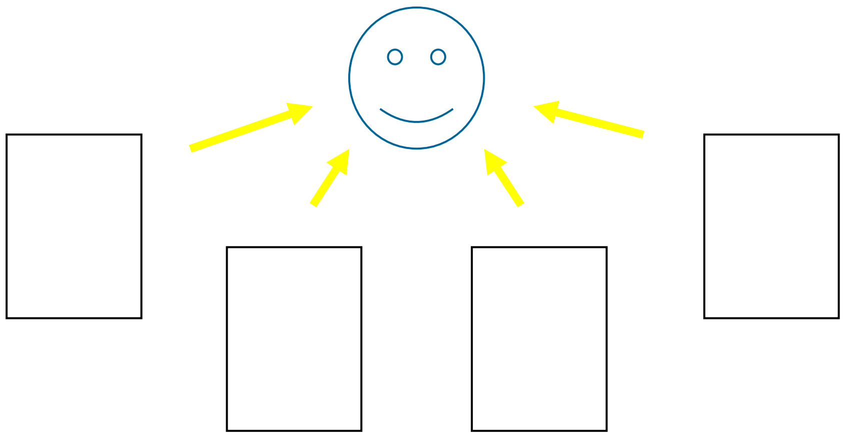
Multitemporal fusion

Multimodal fusion

Fusion for image restoration

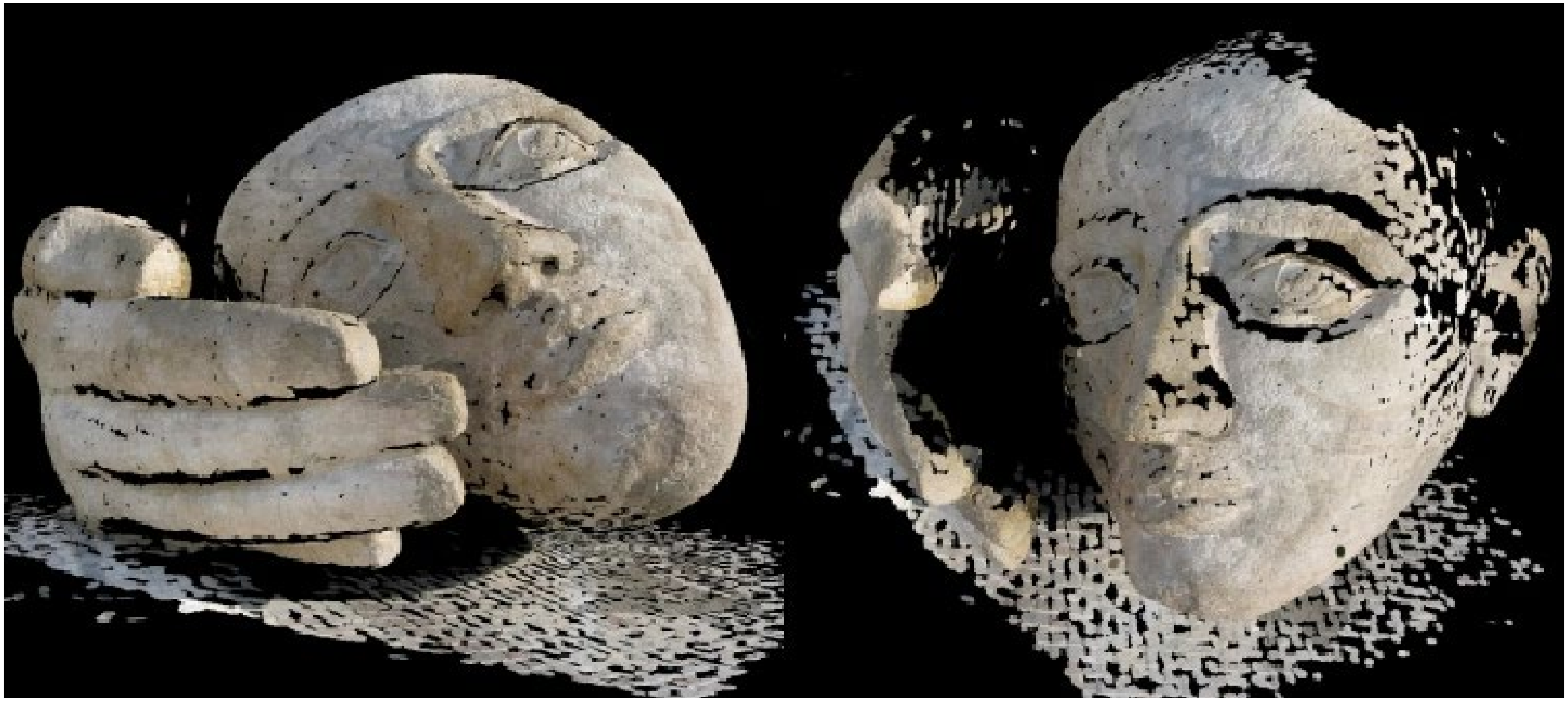
Multiview Fusion

Images of the same modality, taken at the same time but from different places or under different conditions



Goal: to supply complementary information from different views

Multiview fusion - stereo



Courtesy of CMP, CVUT, Prague

Fusion categories

Multisetting fusion

Multiview fusion

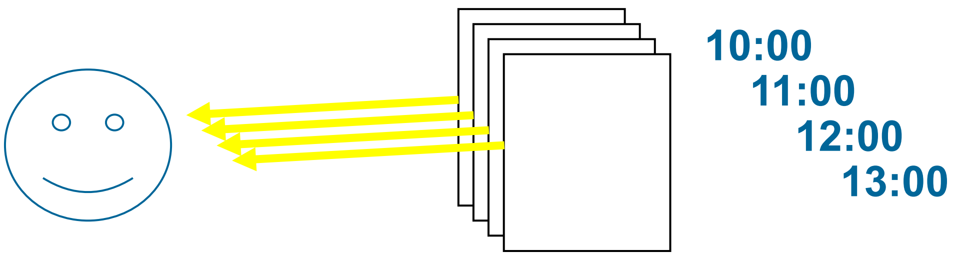
Multitemporal fusion

Multimodal fusion

Fusion for image restoration

Multitemporal Fusion

Images of the same scene taken at different times (usually of the same modality)



Goal: Change detection, noise suppression, image synthesis

Methods: Subtraction, false color synthesis, time averaging, image blending

Synthesis of artificial images

Fusion categories

Multisetting fusion

Multiview fusion

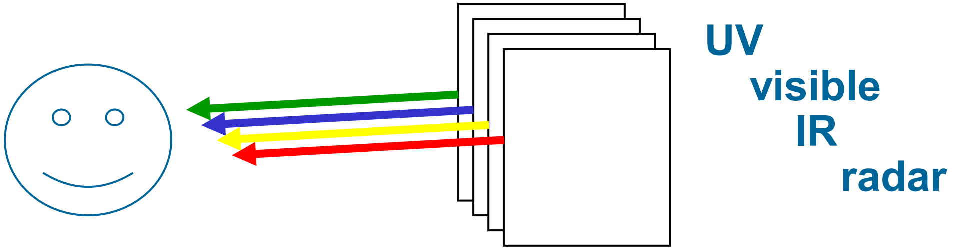
Multitemporal fusion

Multimodal fusion

Fusion for image restoration

Multimodal Fusion

Images of different modalities: PET, CT, MRI, visible, infrared, ultraviolet, etc.



Goal: To emphasize band-specific information

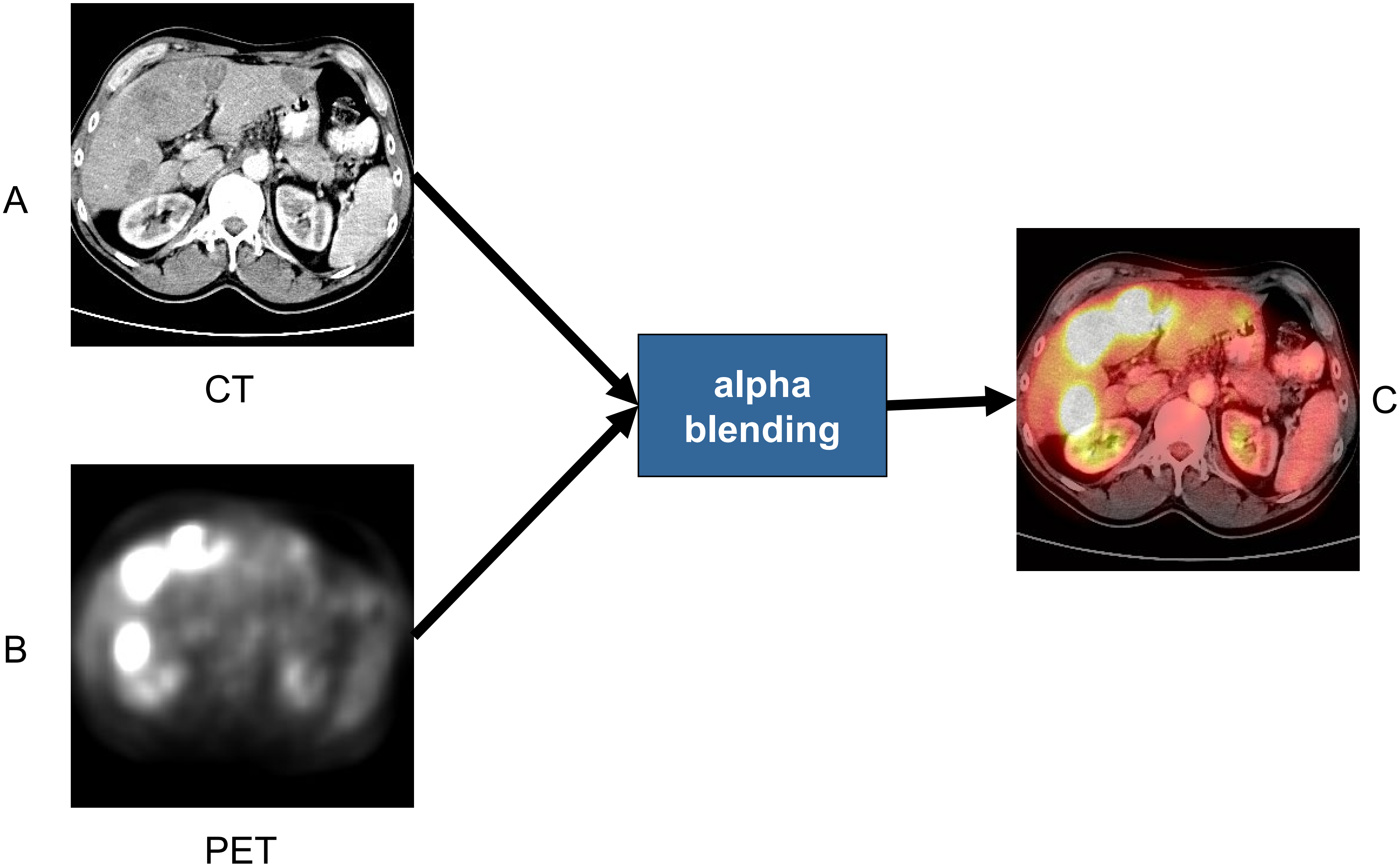
Multimodal Fusion

Pixel-wise fusion

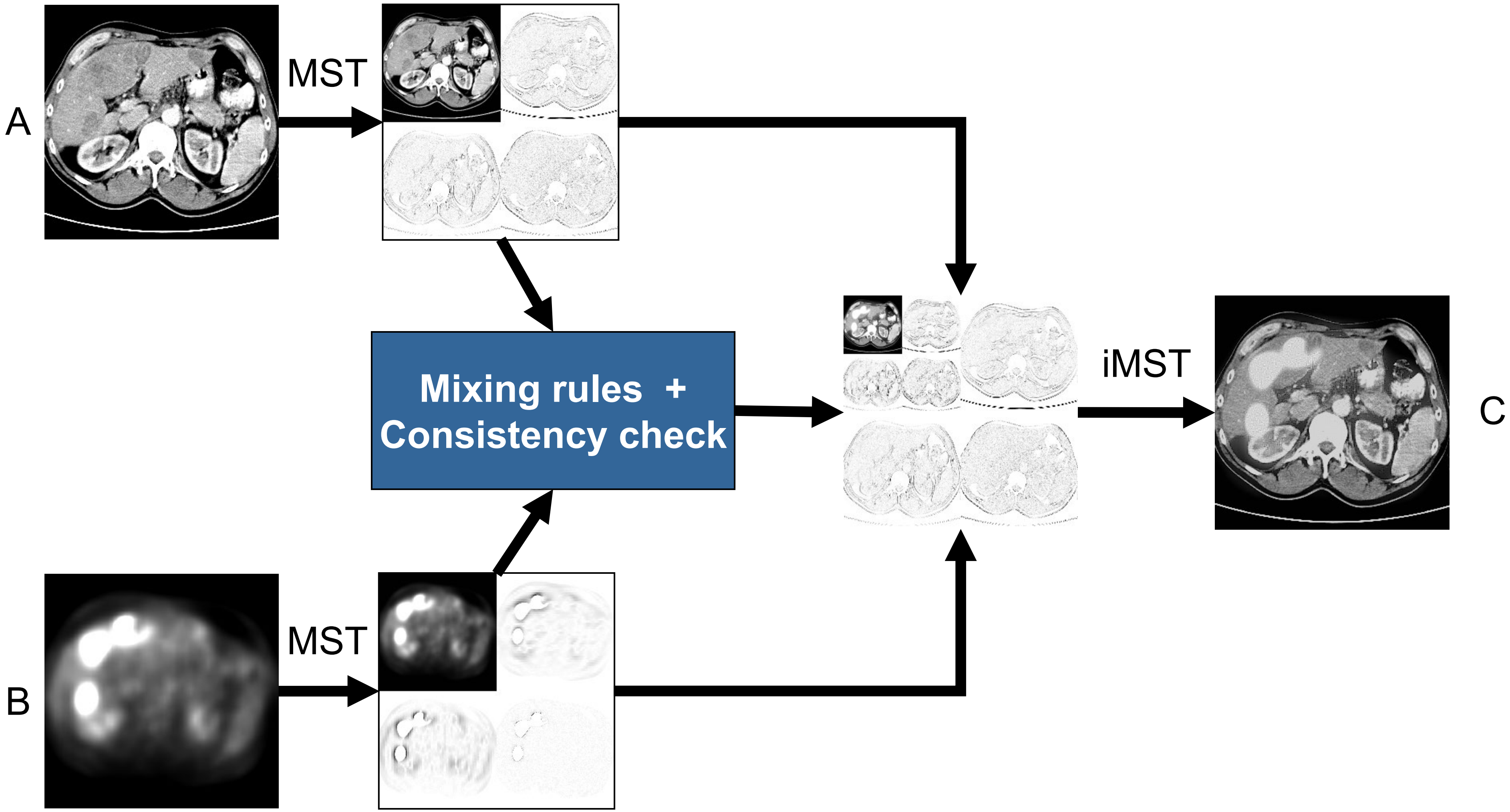
Fusion in transform domains

Object-level fusion

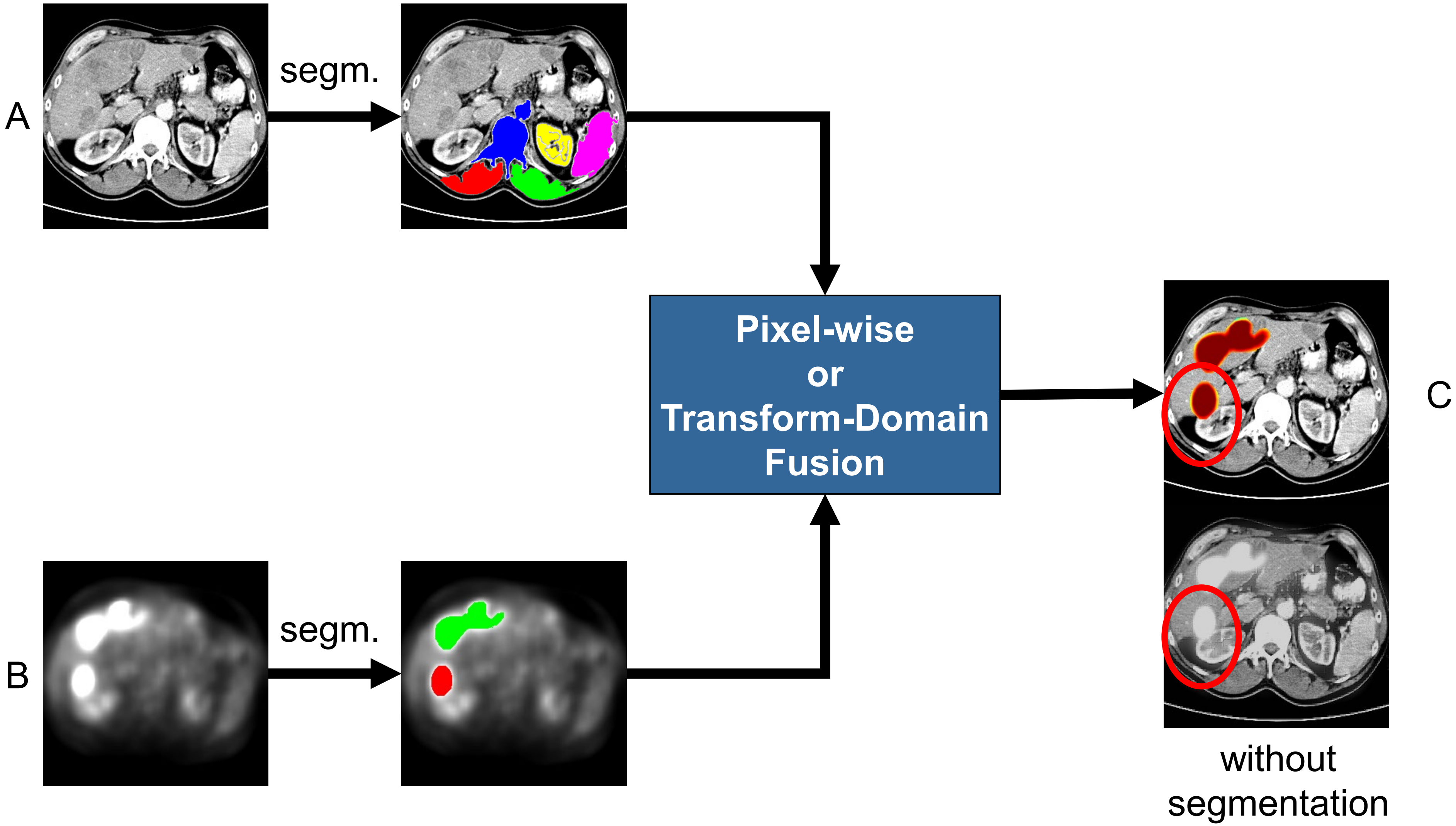
Pixel-wise fusion



Transform domain

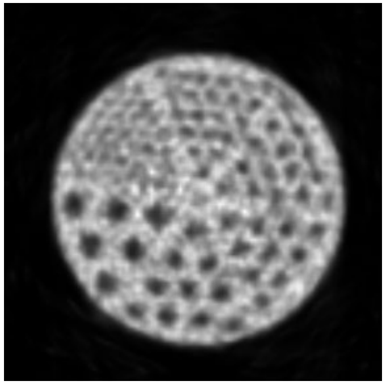


Object-level



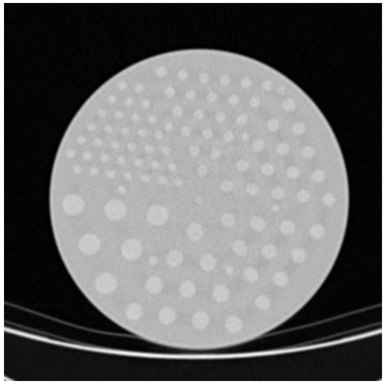
Multimodal Fusion object level fusion

PET



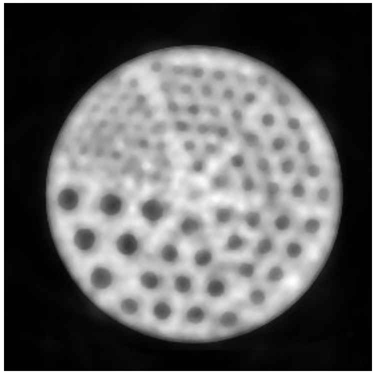
multimodal fusion for quality enhancement

CT



Jaszczak SPECT Phantom

MAP restoration

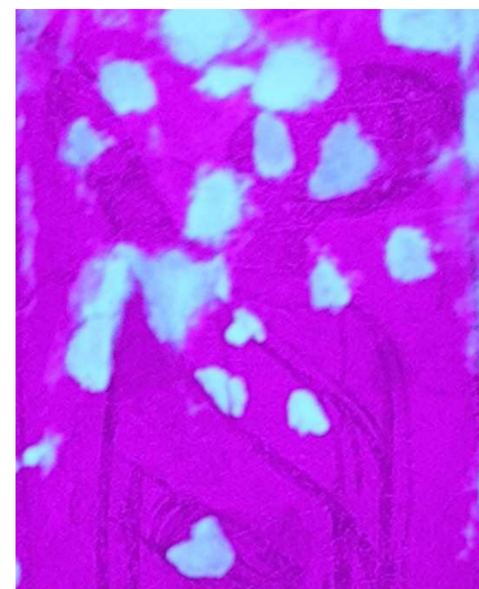


Multimodal Fusion
art conservation applications

ultraviolet
wide band

visible light

ultraviolet
narrow band

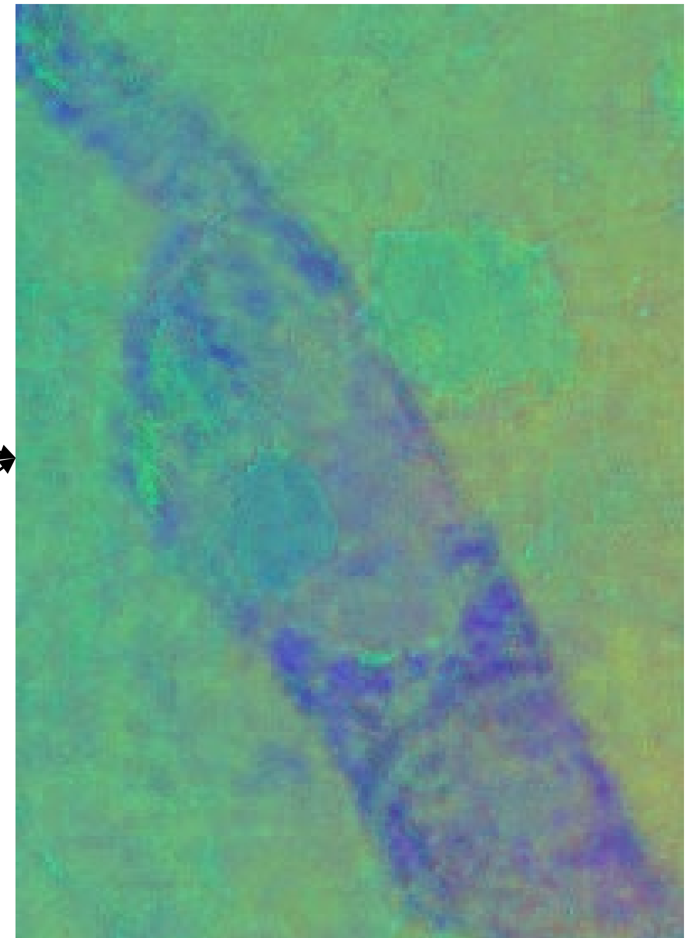


Multimodal Fusion art conservation applications

PCA -based fusion

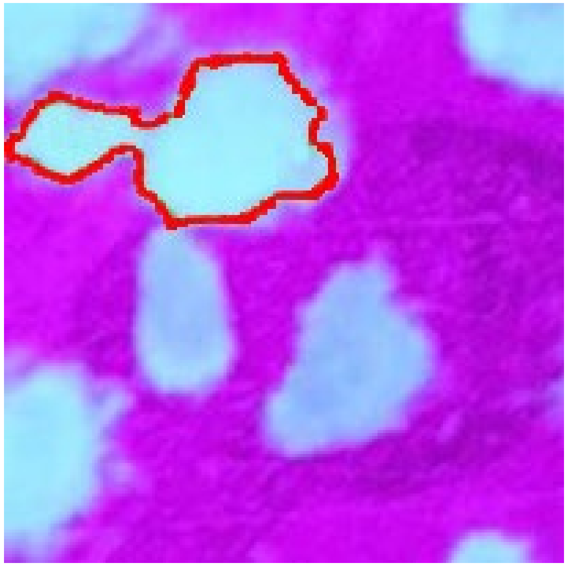
VIS

UV



Multimodal Fusion
art conservation applications

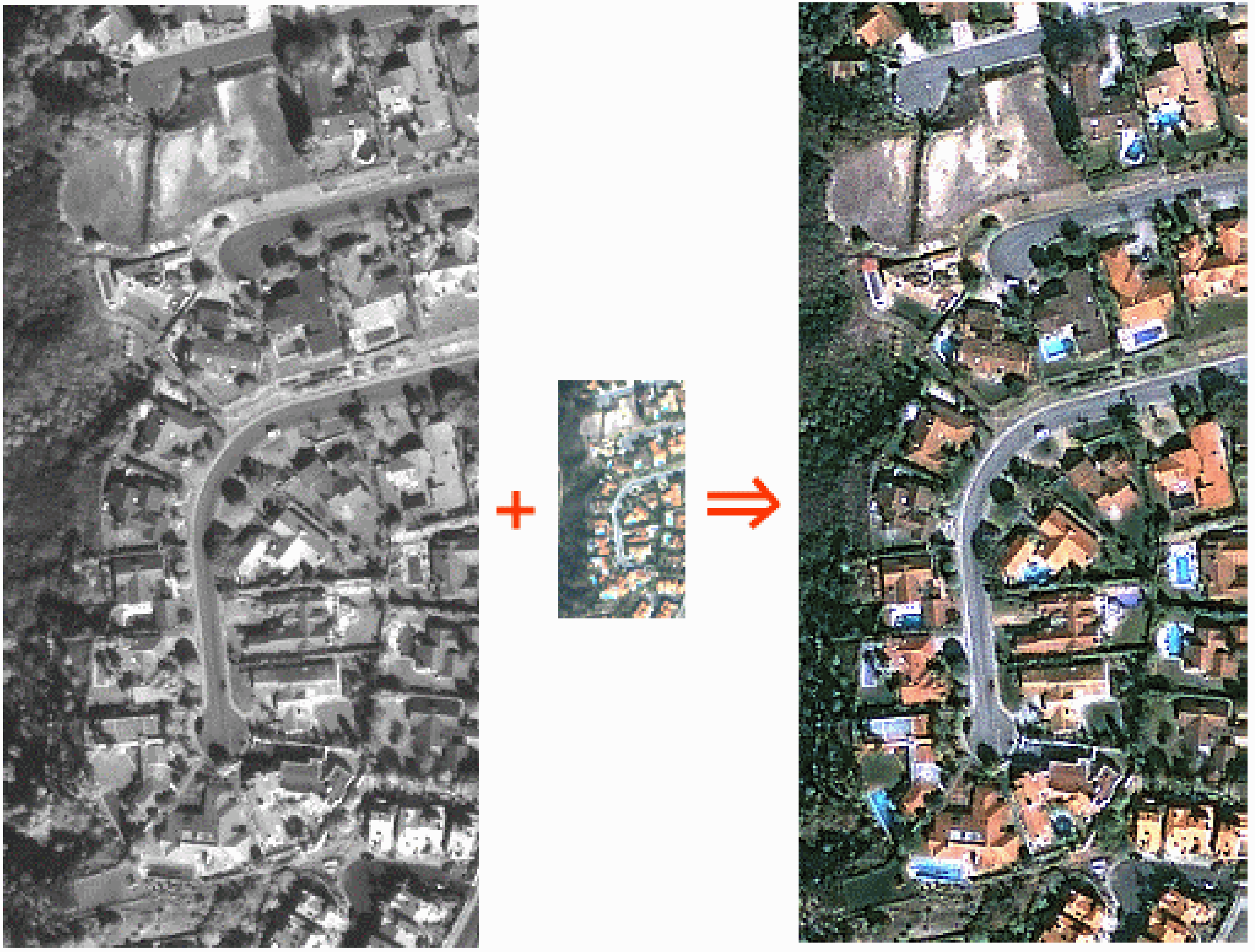
fusion for change detection



Multimodal fusion of images with different resolution

One image with high spatial resolution, the other one with low spatial but high spectral resolution.

Goal: An image with high spatial and spectral resolution



Multimodal fusion of images with different resolution

Goal: An image with high spatial and spectral resolution

Methods: Replacing intensity in IHS

Replacing intensity in PCA

Replacing high frequencies

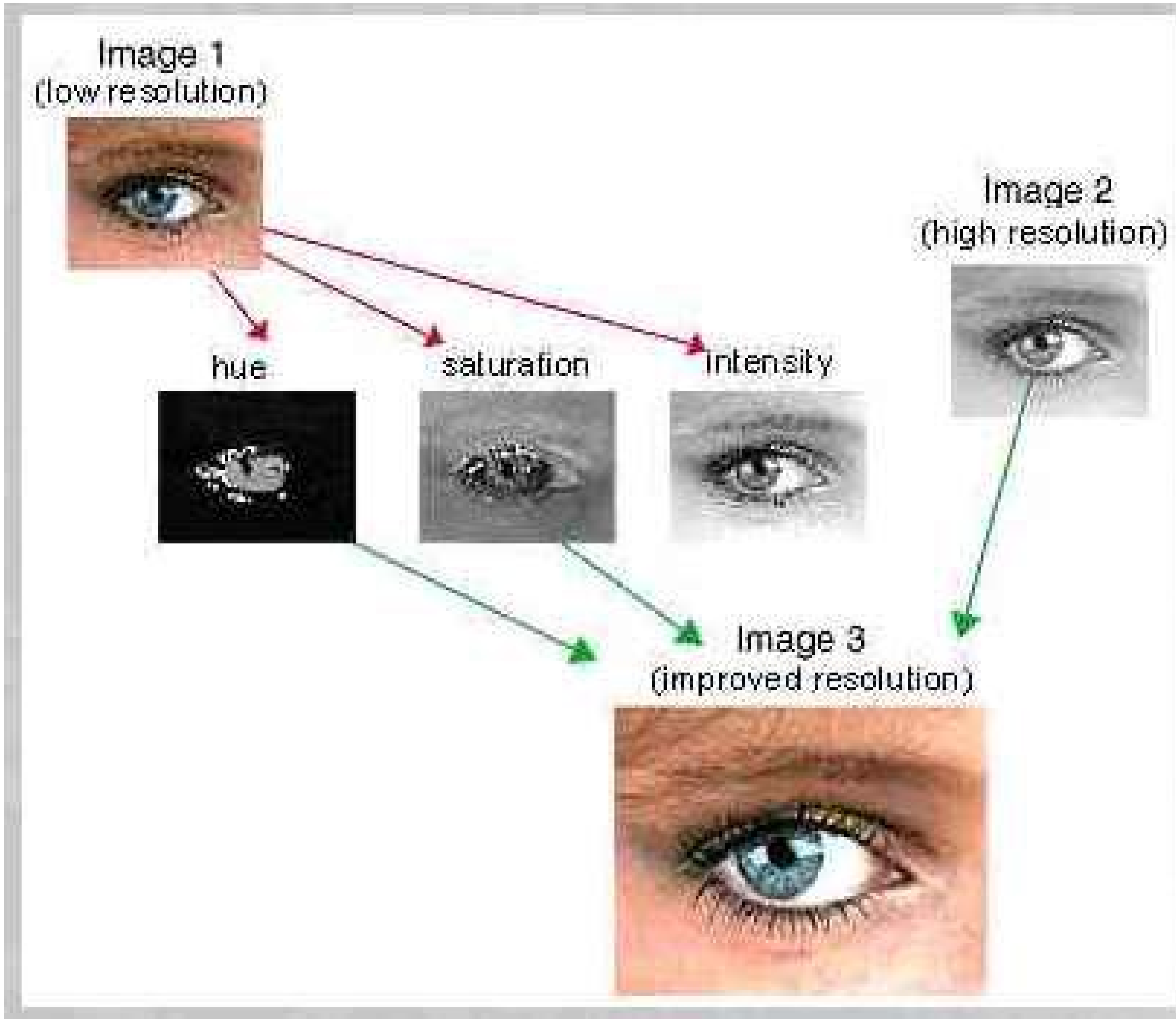
Replacing bands in WT

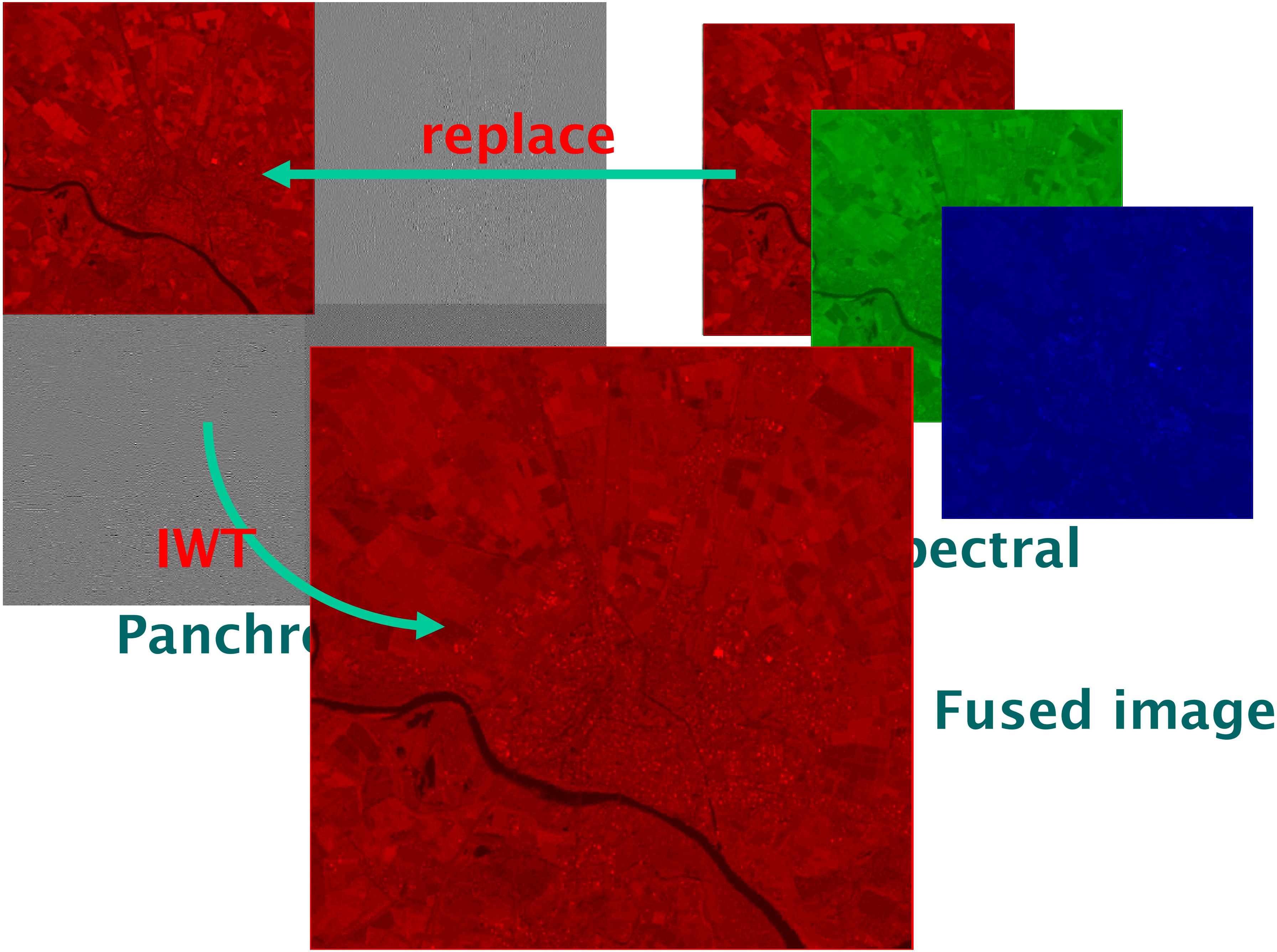
IHS transformation
RGB image -> HIS

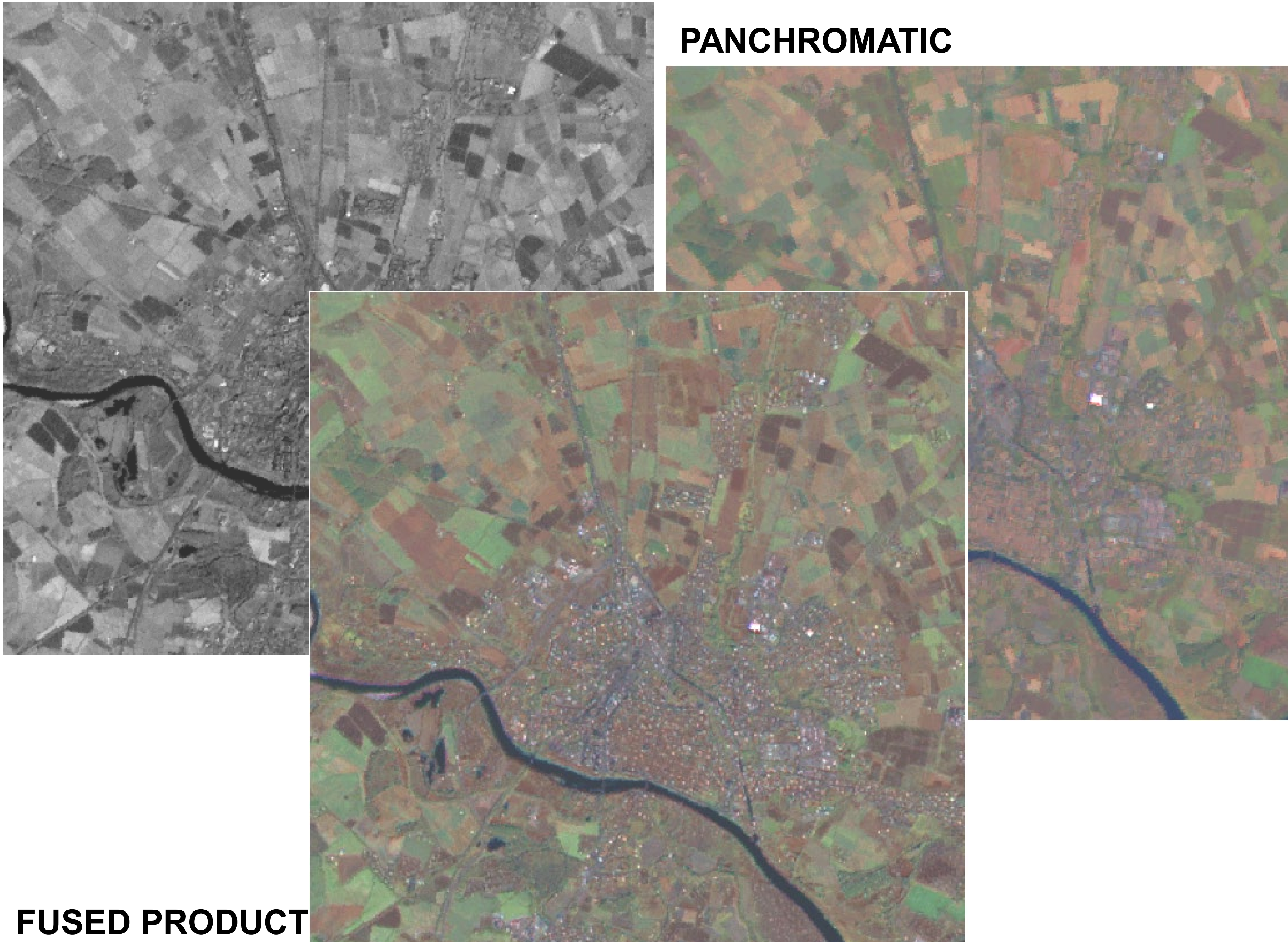
Hue,Saturation, Intensity

I -> PAN

HIS -> RGB




replace
IWT
Panchr
pectral
Fused image


PANCHROMATIC
FUSED PRODUCT

Original HRPI (panchromatic band)

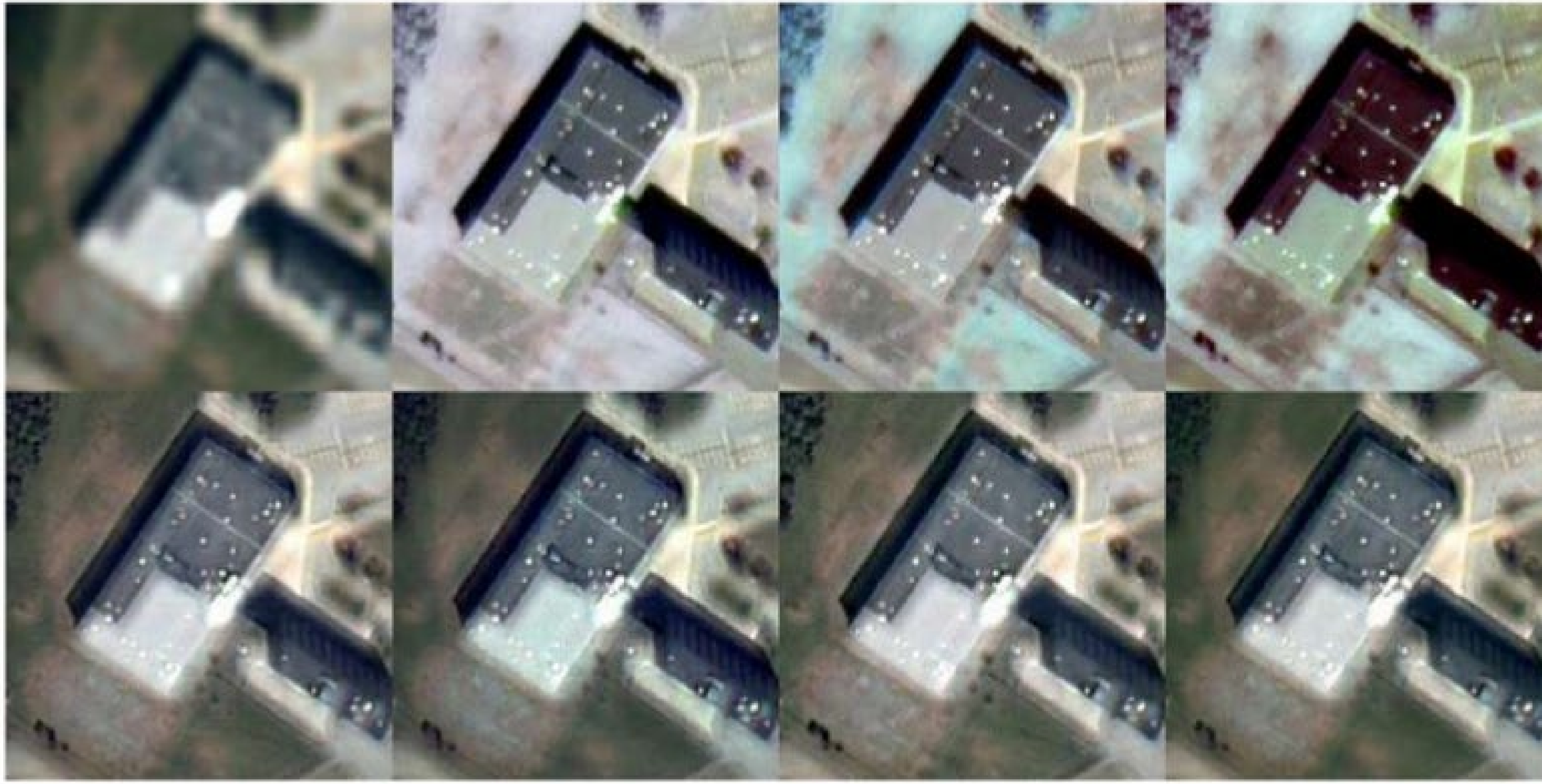
Original LRMIs (RGB) (resampled at 1-m pixel size).

HS

Result of the HPF method

PCA method

Wavelety



Morfologie

- Předzpracování – odšumování, skeletonizace, konvexní obal

- Segmentace

- Rozpoznávání – plocha, hranice

Morfologie
Strukturní element

- adekvátní studovanému objektu

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

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

Morfologie
Základní operace

Eroze

Dilatace

drží tvar,

lehce vyhlazený vzhledem k

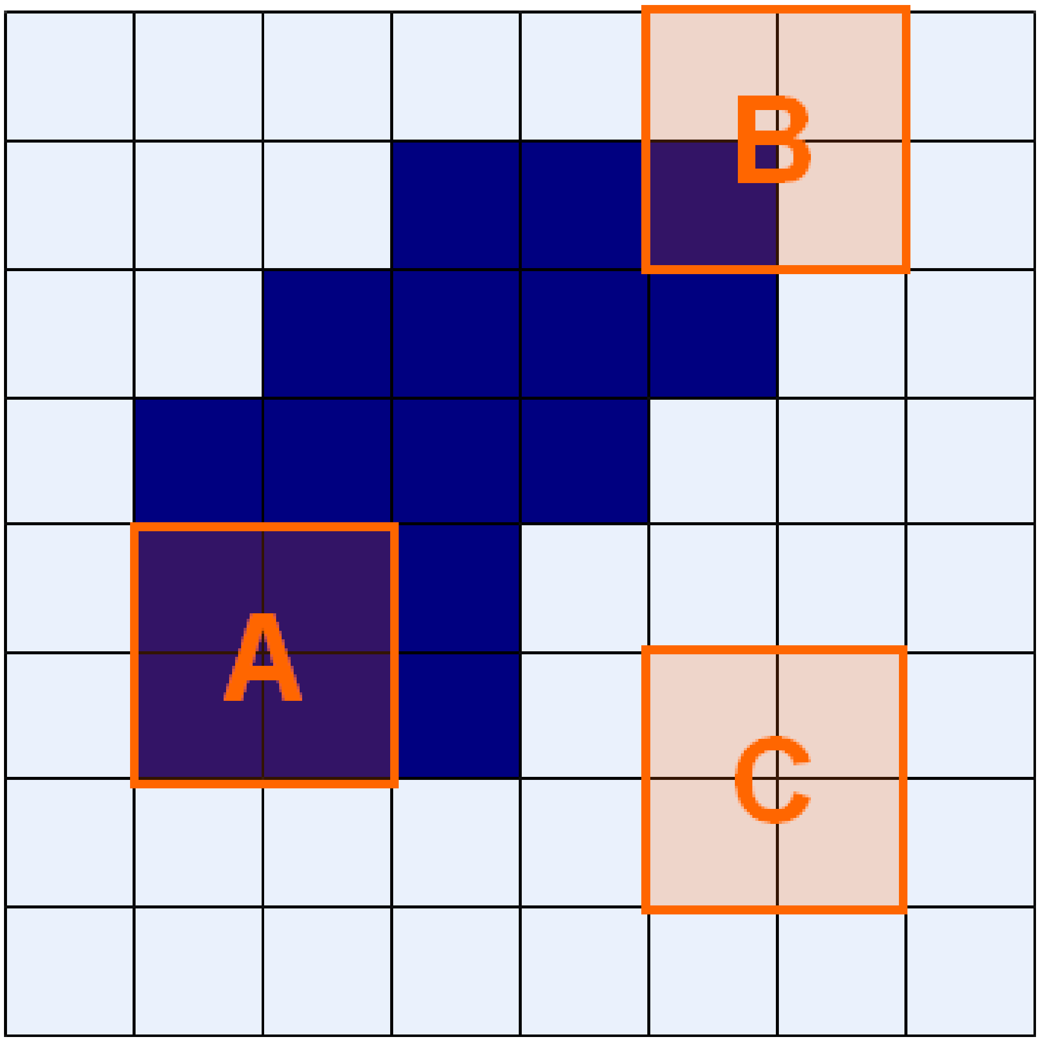
Kombinace –

otevření objektu

uzavření pozadí

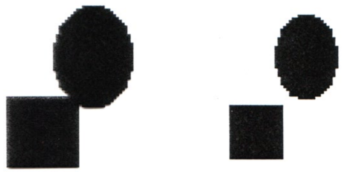
Morfologie – dilatace a eroze

Fit & Hit & Miss



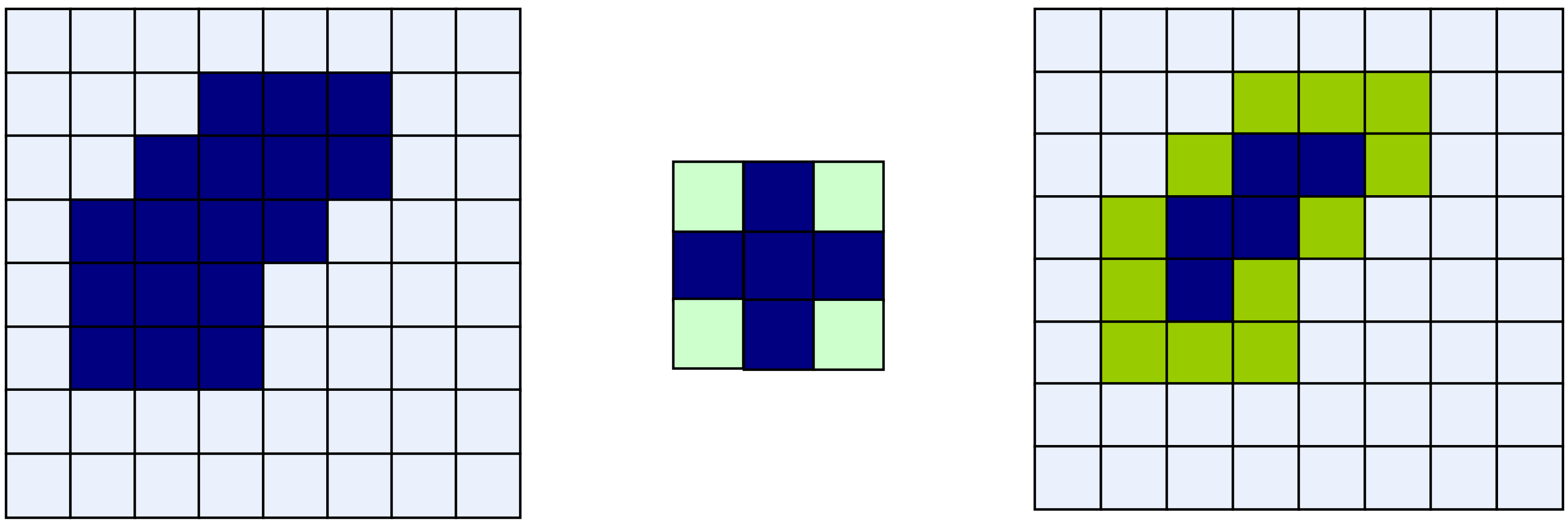
Morfologie – dilatace a eroze

Eroze



f ⊖ s

$$g(x,y)=\begin{cases}1 \text{ jestliže } s \text{ FIT } f\\ 0 \text{ jinak}\end{cases}$$



Morfologie – dilatace a eroze

Eroze

$$E = \mathbb{Z}^2$$

$$A \ominus B = \{z \in E | B_z \subseteq A\}$$

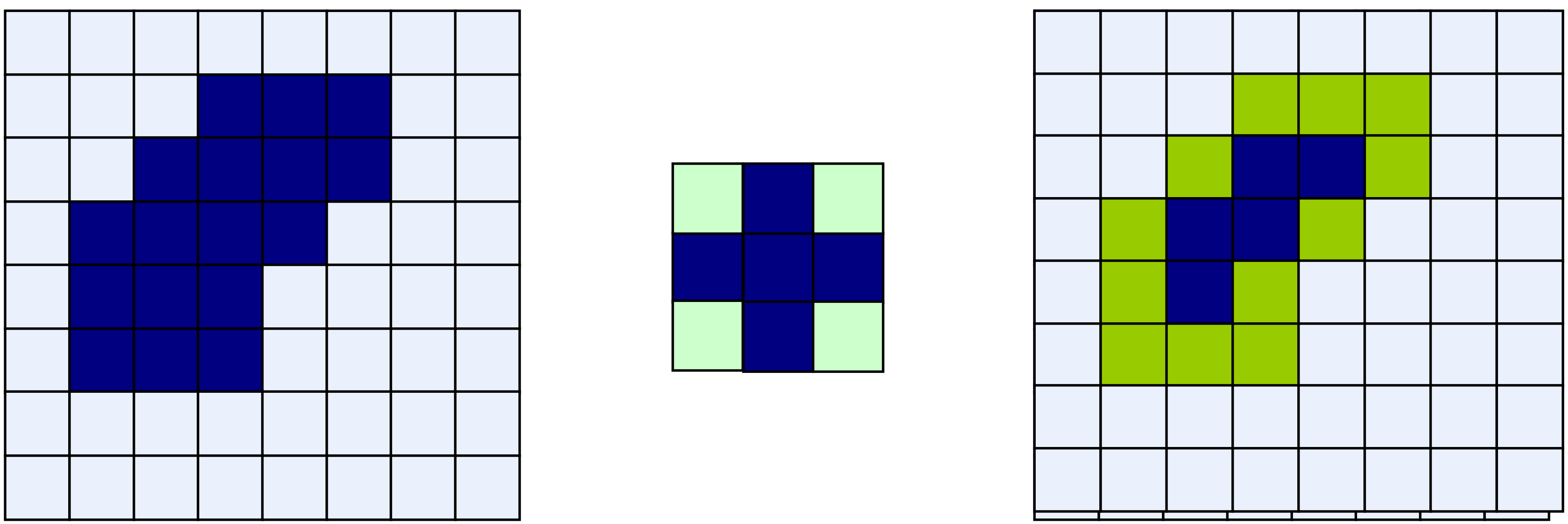
$$B_z = \{b + z | b \in B\}$$

Morfologie – dilatace a eroze

Dilatace

f ⊕ s

$$g(x,y) = \begin{cases} 1 \text{ jestliže } s \text{ HIT } f \\ 0 \text{ jinak} \end{cases}$$



Morfologie – dilatace a eroze

Dilatace

$$A \oplus B = \{z \in E \,|\, (B)_z \cap A \neq \varnothing\}$$

Morfologie – dilatace a eroze
vlastnosti

$$A \oplus B = B \oplus A$$

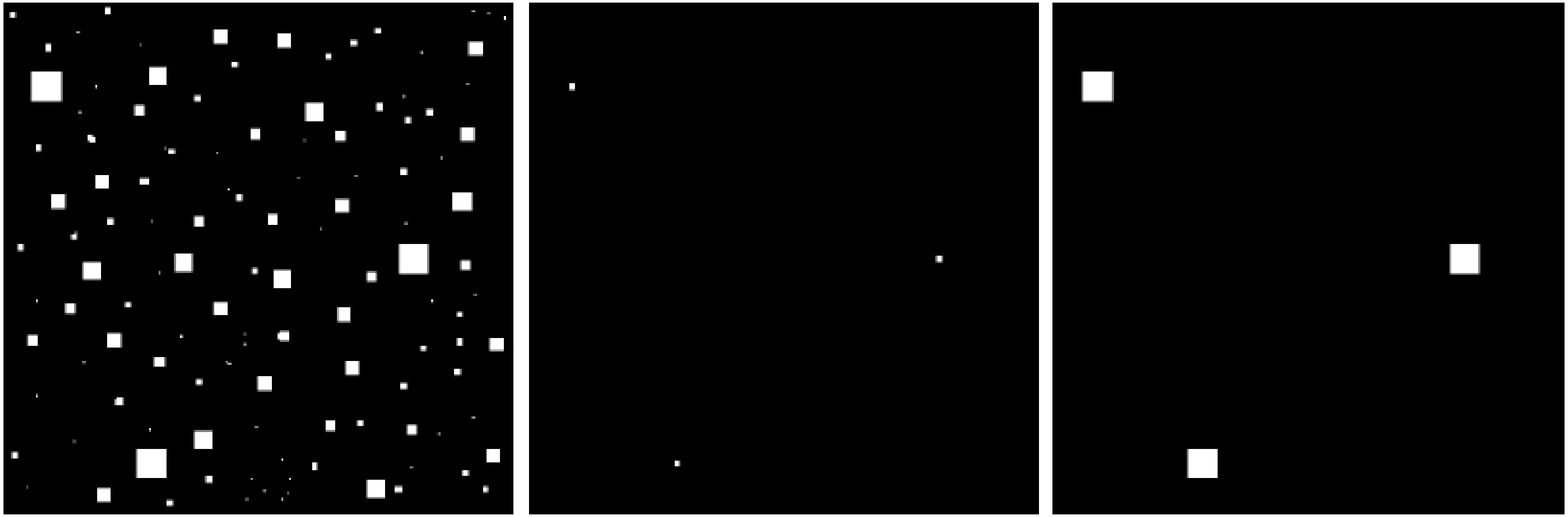
$$(A \oplus B) \oplus C = A \oplus (B \oplus C))$$

eroze - neplatí komutativita, asociativita

Eroze – odstranění struktur daného tvaru

Dilatace – zaplnění děr daného tvaru

Morfologie – složené operace



Morfologie – složené operace

-opakování základních operací EROZE a DILATACE,
 stejný SE

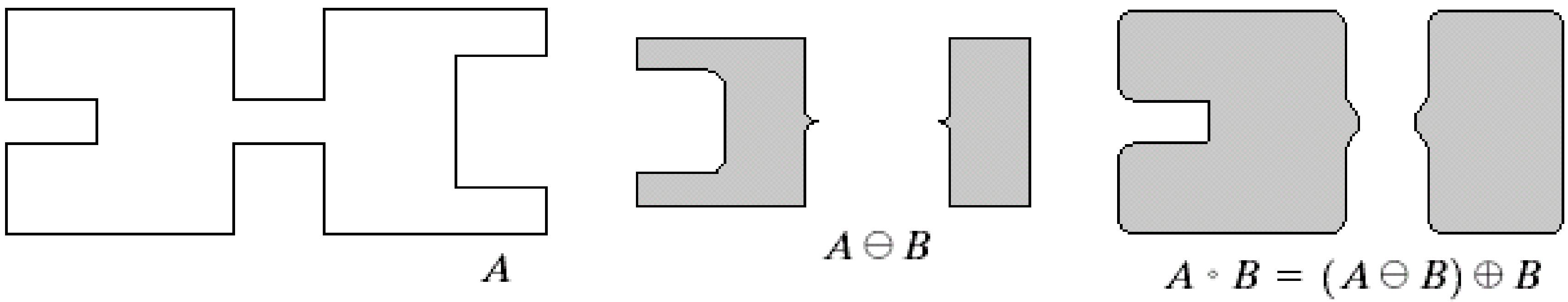
- OTEVŘENÍ = EROZE -> DILATACE

- UZAVŘENÍ = DILATACE -> EROZE

Morfologie – OTEVŘENÍ

$f \circ s = (f \ominus s) \oplus s$

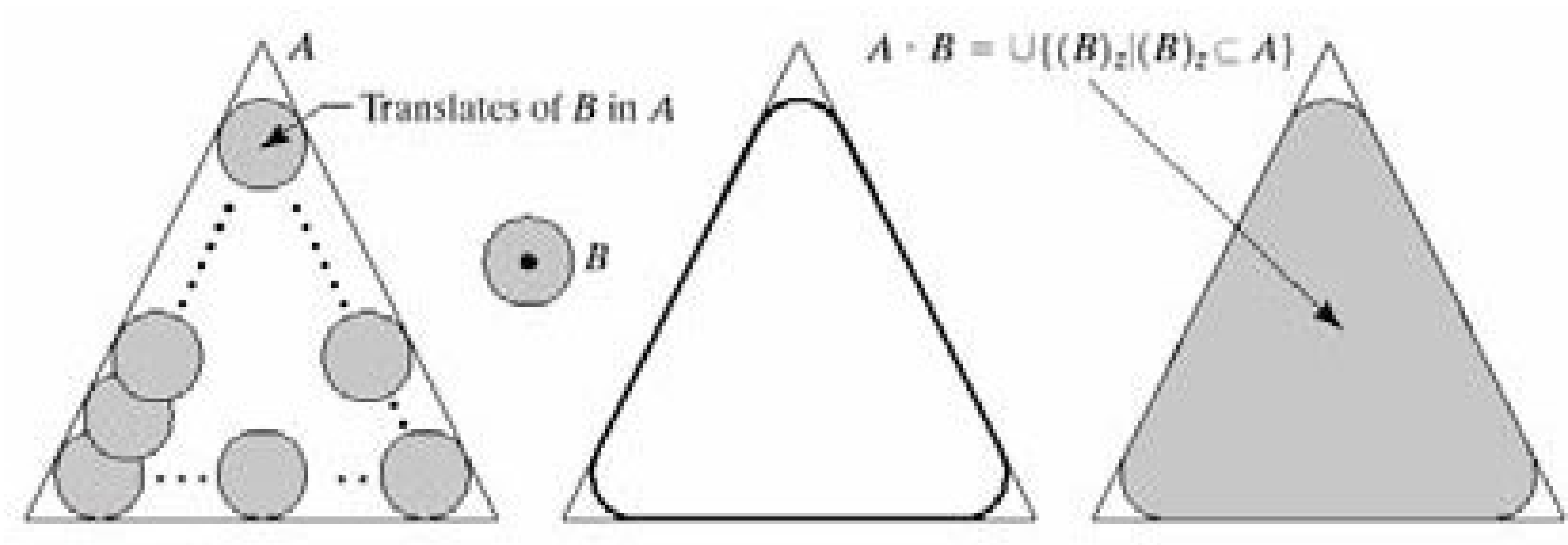
vyhlazuje hranice, rozděluje tenká spojení, odstraňuje malé objekty, ale zachovává tvar

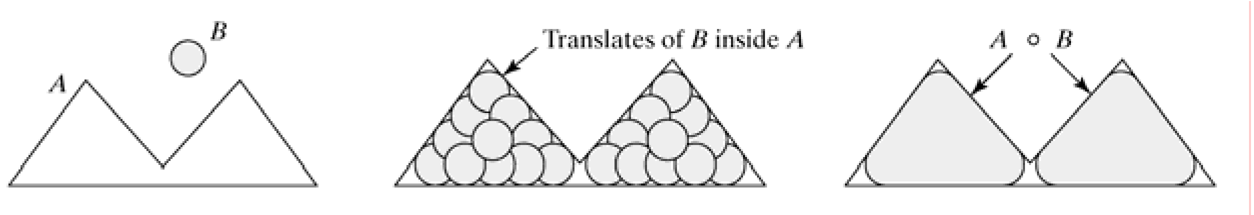


$$A \circ B = \cup \{(B)_z \mid (B)_z \subseteq A\}$$

sjednocení všech posunů B, které pasují do A

Morfologie – OTEVŘENÍ

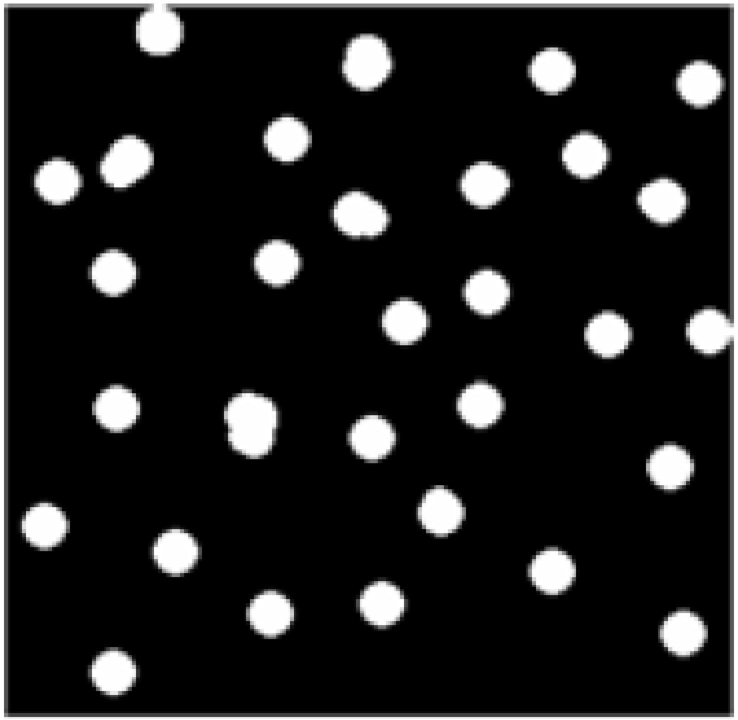




-idempotentní

$$(A \circ B) \circ B = A \circ B.$$

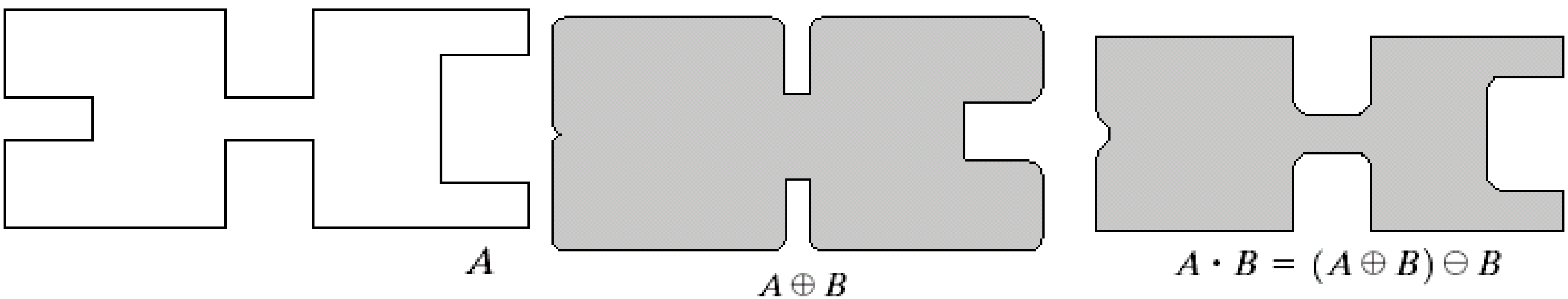
Morfologie – OTEVŘENÍ



Morfologie – UZAVŘENÍ

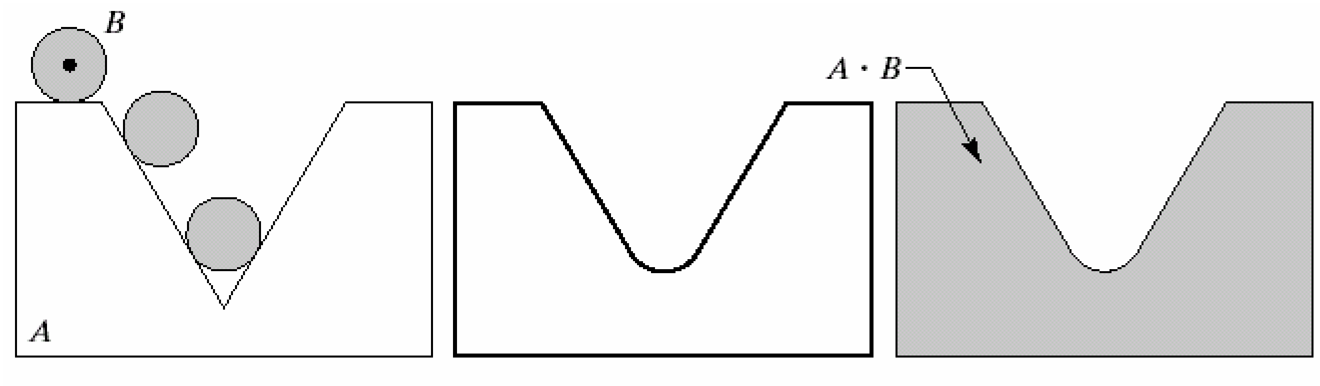
$$f \cdot s = (f \oplus s) \ominus s$$

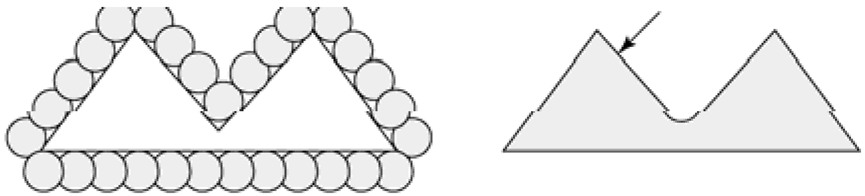
vyhlazuje hranice, odstraňuje malé díry, zaplňuje malé předěly, ale zachovává tvar



Doplněk sjednocení všech posunů B, které se nepřekrývají s A

Morfologie – UZAVŘENÍ





-idempotentní $(A \bullet B) \bullet B = A \bullet B$

Morfologie – UZAVŘENÍ

Morfologie – vlastnosti

Otevření A°B je podmnožina A

jestliže C je podmnožina D,

pak C °B je podmnožina D °B

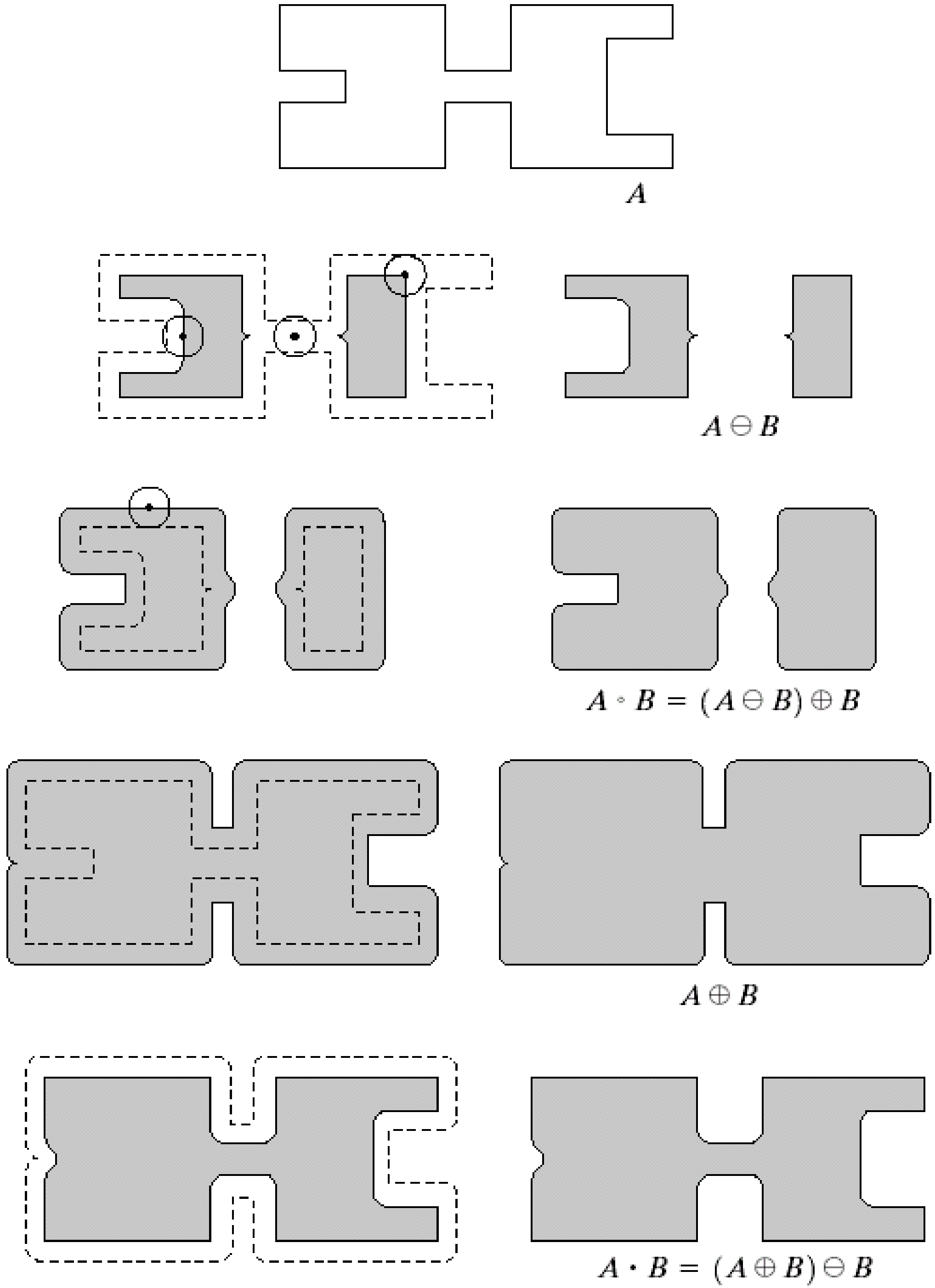
Uzavření A je podmnožina A•B

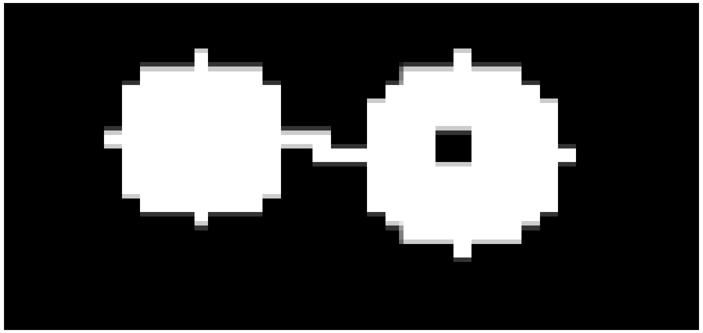
jestliže C je podmnožina D,

pak C •B je podmnožina D •B

Otevření a uzavření jsou duální vzhledem k doplňku a zrcadlení

$$(A \bullet B)^c = (A^c \circ \hat{B})$$

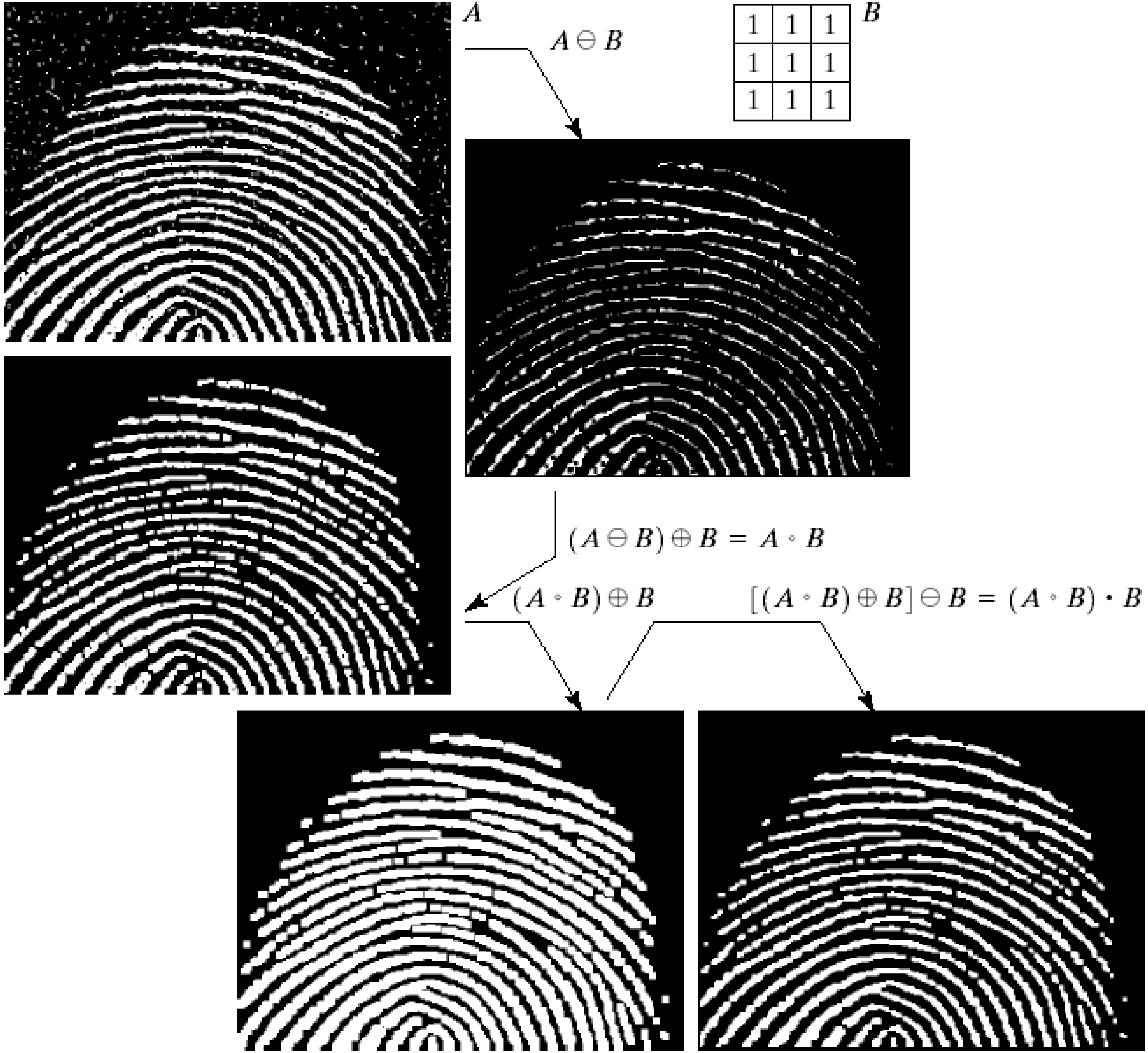

A
$A \ominus B$
$A \circ B = (A \ominus B) \oplus B$
$A \oplus B$
$A \bullet B = (A \oplus B) \ominus B$



A

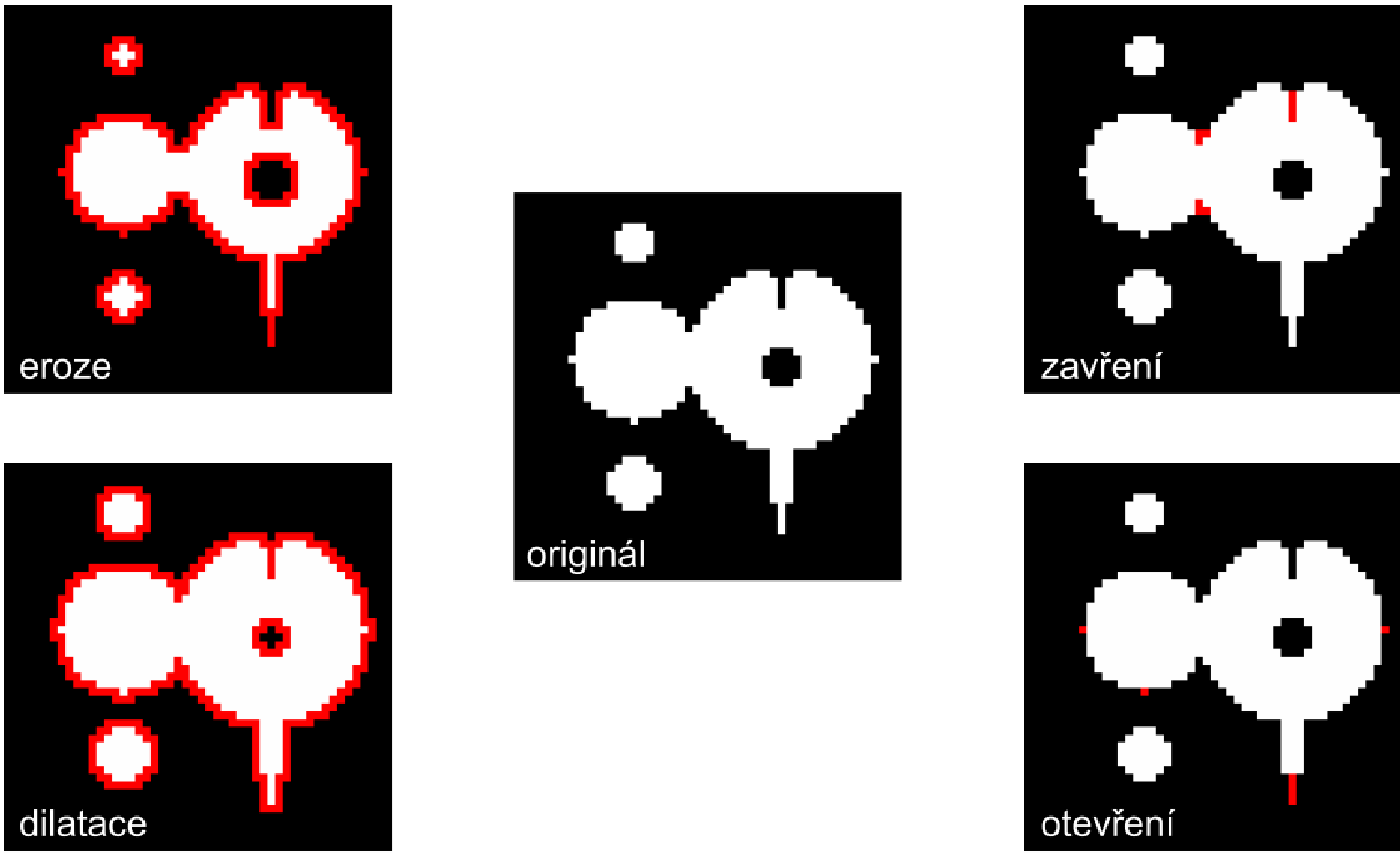
opening of A
→removal of small protrusions, thin connections, ...

closing of A
→ removal of holes



A
A ⊖ B
B
1 1 1
1 1 1
1 1 1
(A ⊖ B) ⊕ B = A ∘ B
(A ∘ B) ⊕ B
[(A ∘ B) ⊕ B] ⊖ B = (A ∘ B) • B

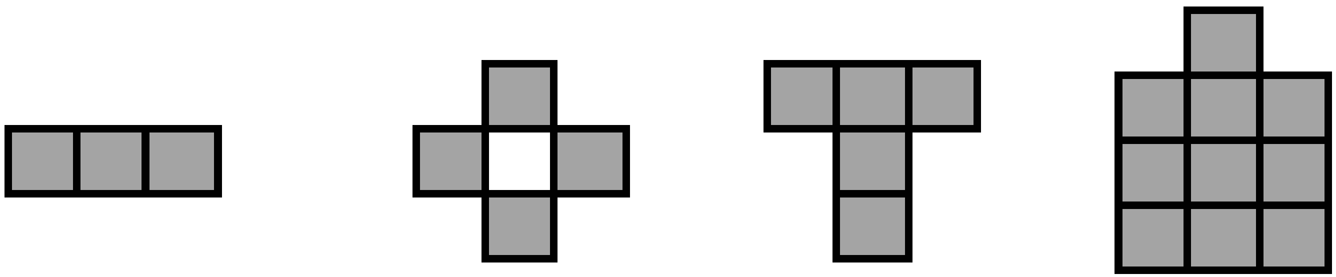
Porovnání operací



Morfologie – Hit or Miss

$$\circledast$$

Detekce požadovaného tvaru   ”template matching



Strukturální element – objekt (B1) a pozadí (B2)

Pasuje B1 do objektu *a současně* B2 nepasuje do objektu, tedy pasuje do pozadí ?

Morfologie – Hit or Miss

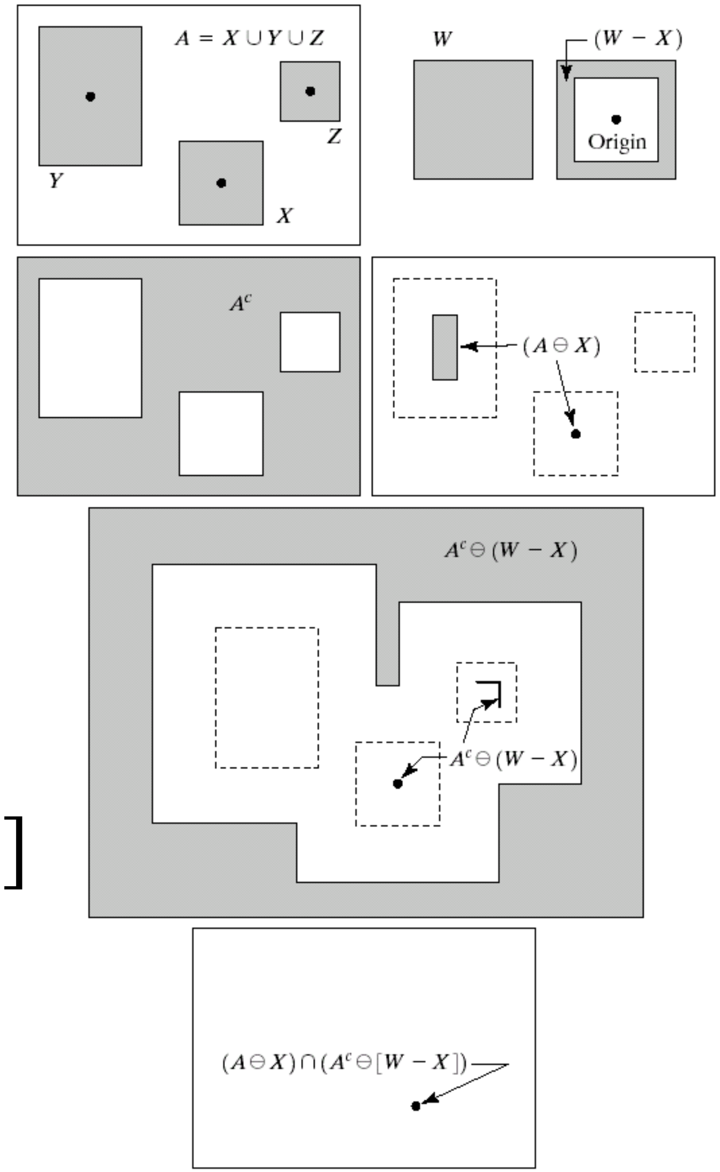
A, okno W a pozadí (W-X)

doplněk A, eroze A s X

eroze doplňku A s (W-X)

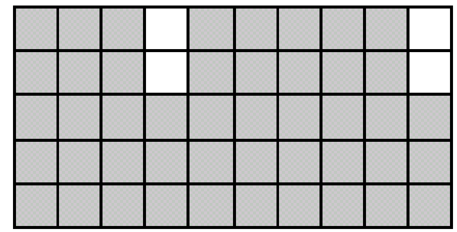
průnik s pozicí X

$$A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)]$$

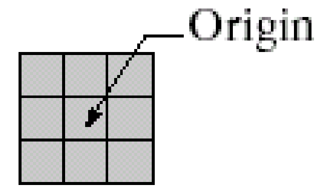


Morfologie – detekce hranice

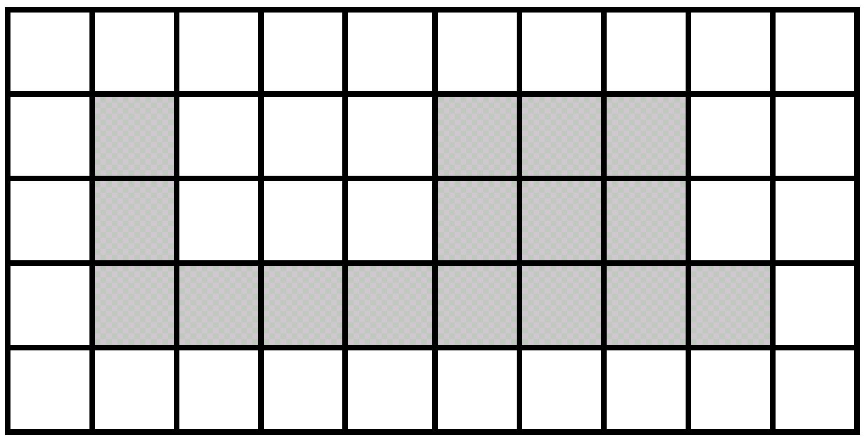
$$\beta(A) = A - (A \ominus B)$$



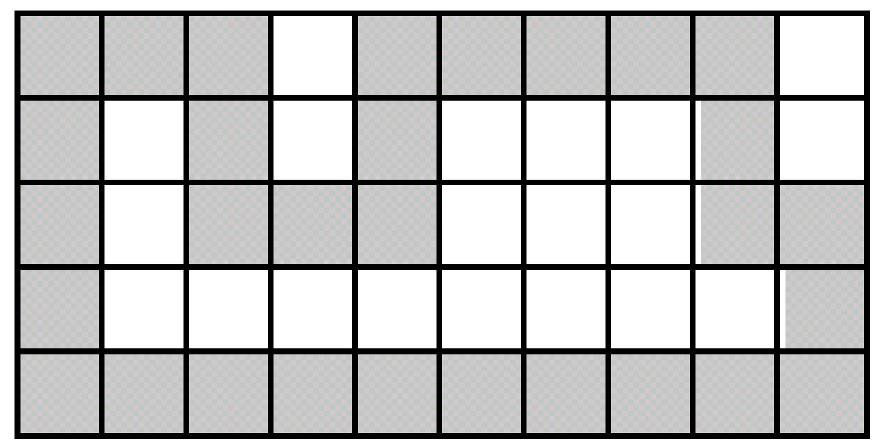
A



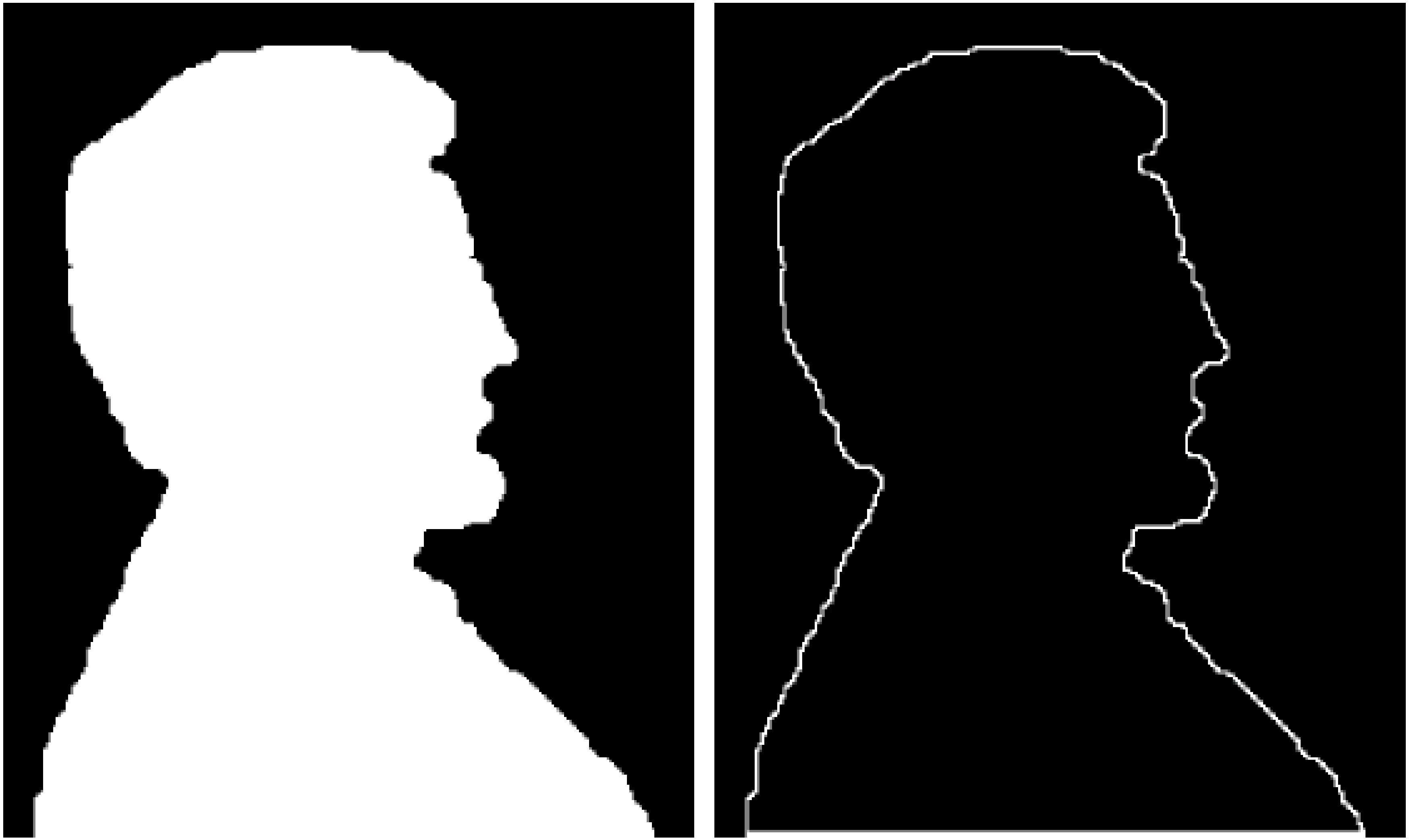
B



$A \ominus B$

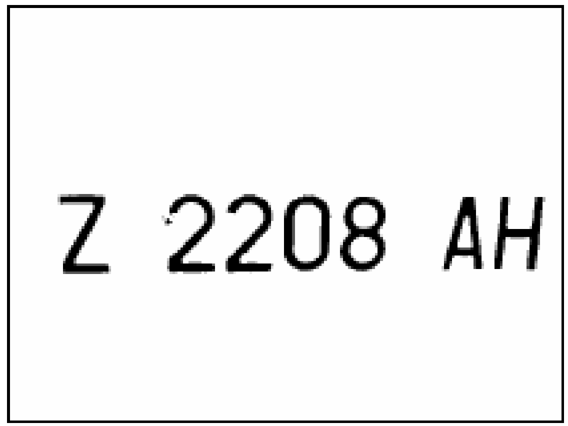


$\beta(A)$

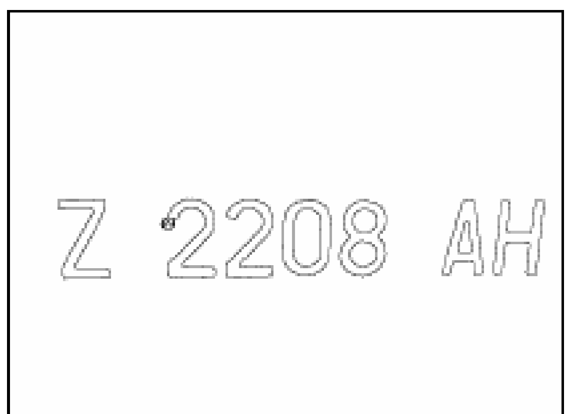


A

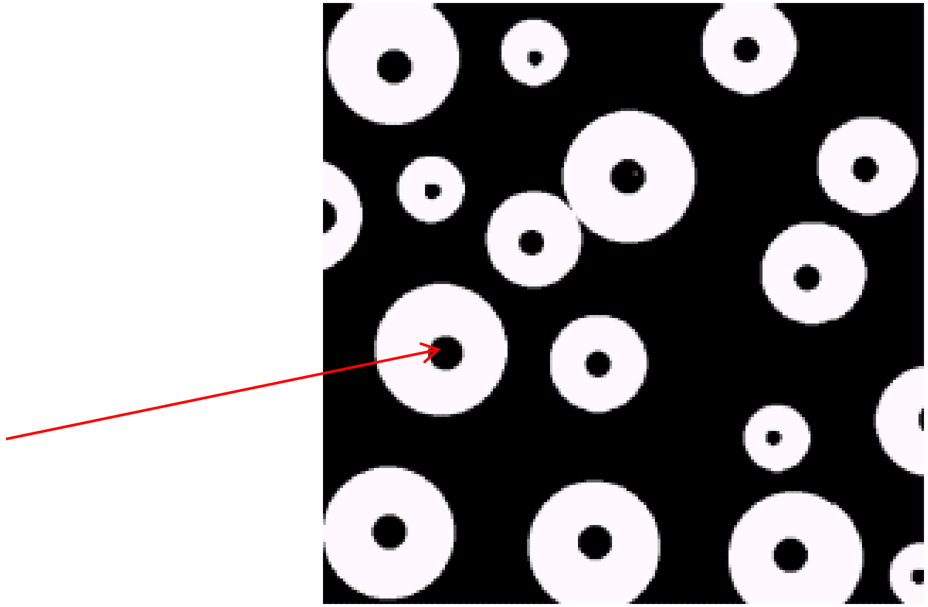
$A \ominus B$



$\beta(A)$

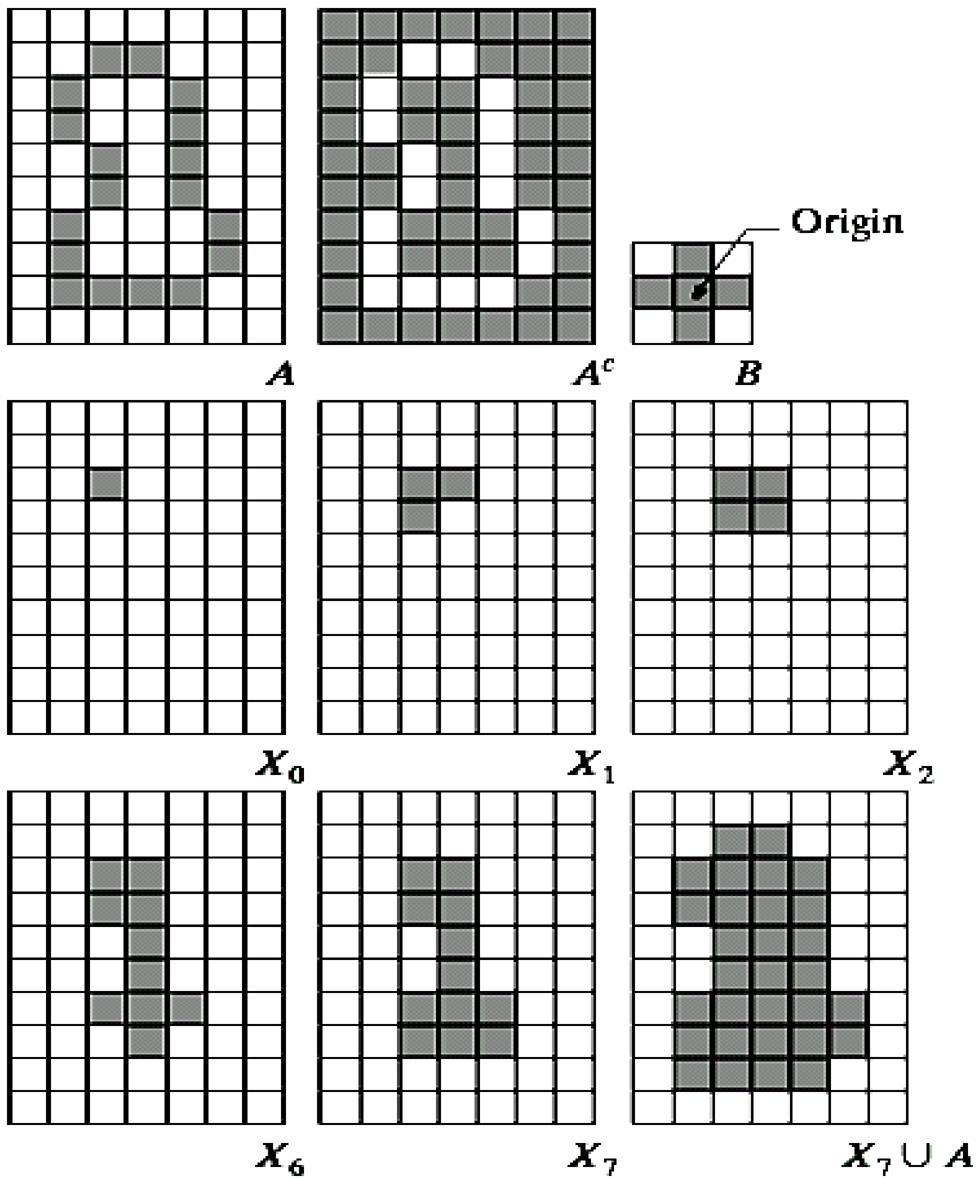


Morfologie – zaplňování mezer

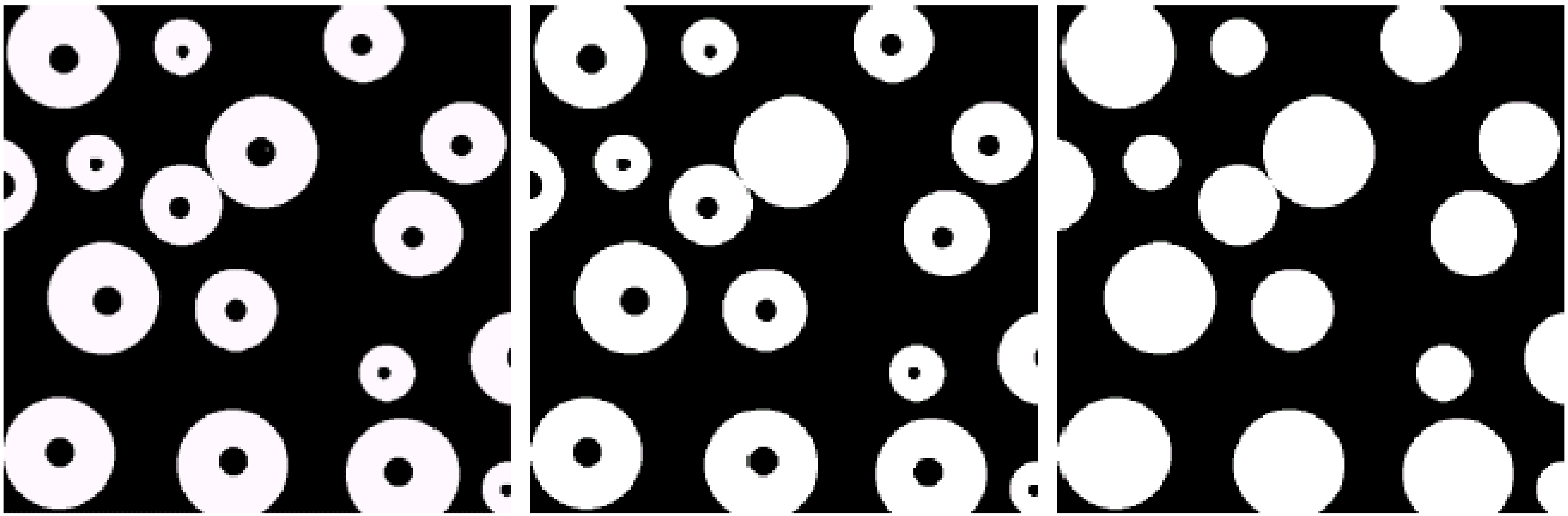


$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

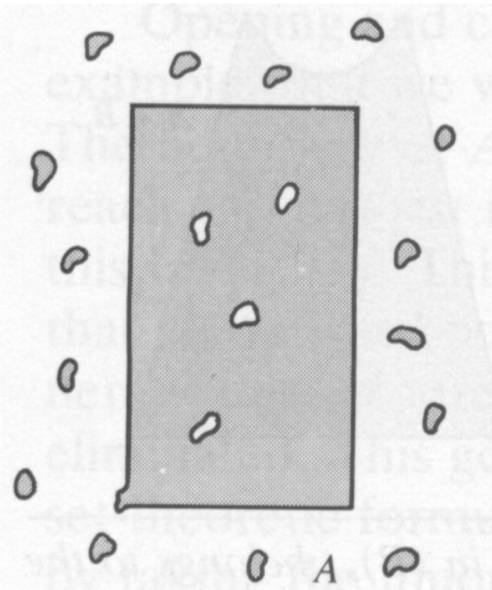
iterativně

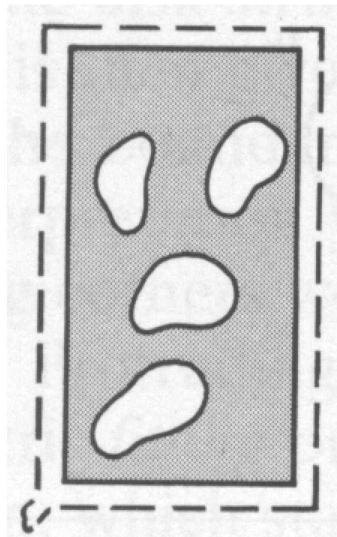


$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

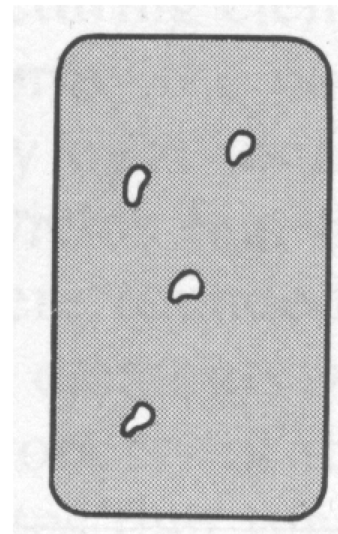


Morfologie – vyhlazování

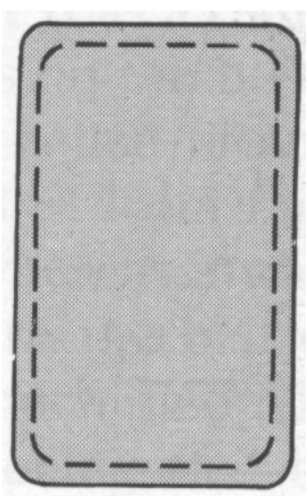




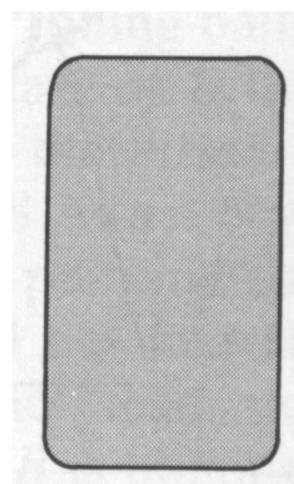
eroze



otevření



dilatace

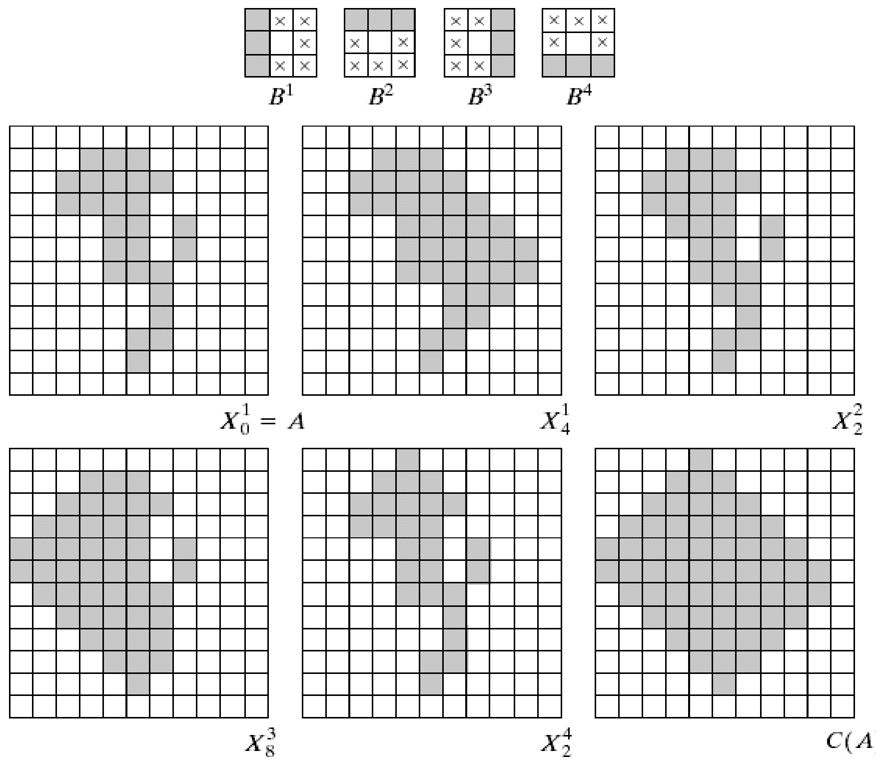


uzavření

Morfologie – konvexní obal

$$X_{k+1}^{i} = (X_{k}^{i} \circledast B^{i}) \cup A \quad i = 1,2,3,4 \text{ and } k = 1,2,3,\ldots$$

A je konvexní, jestliže každá přímá spojnice dvou bodů z A je v A.



(a) Structuring elements. (b) Set *A*. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull.

$$C(A) = \bigcup_{i=1}^{4} D^{i}$$

Morfologie – thinning

$$A \otimes B = A - (A \circledast B)$$
$$= A \cap (A \circledast B)^c$$

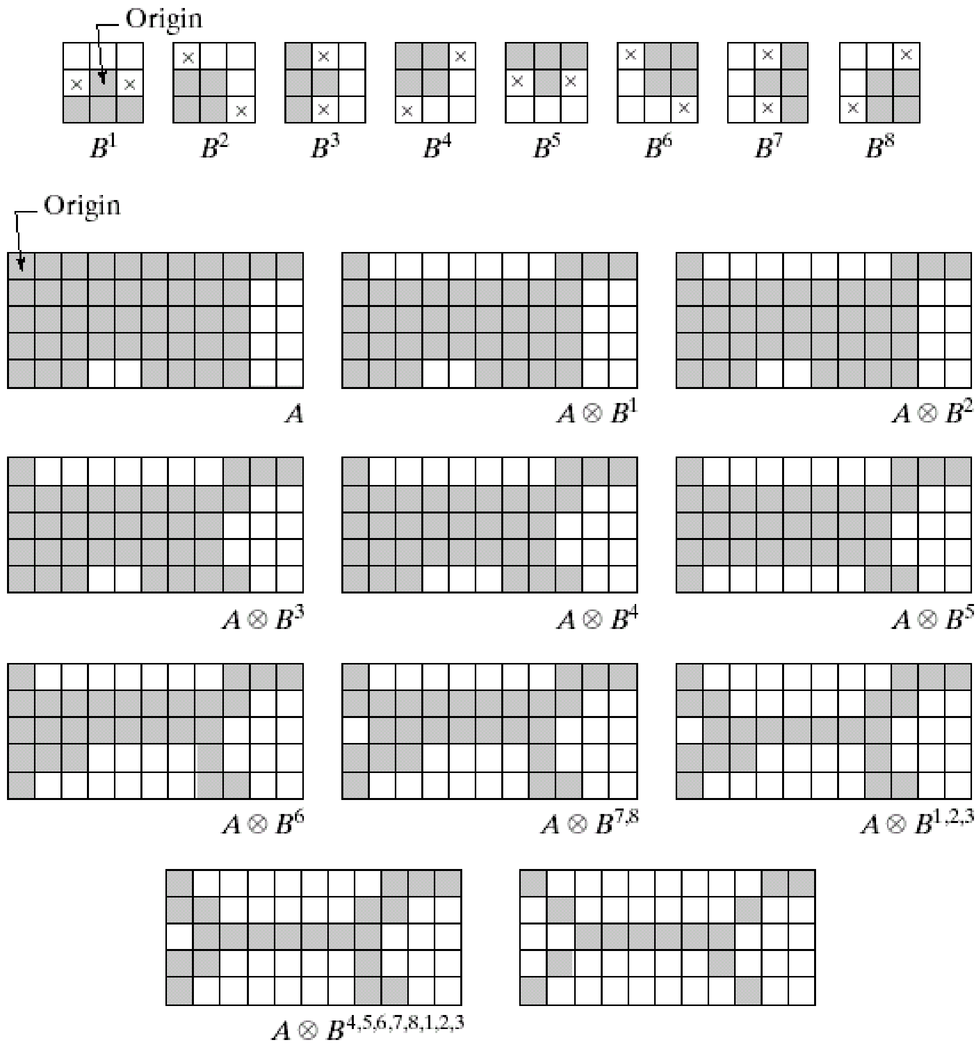
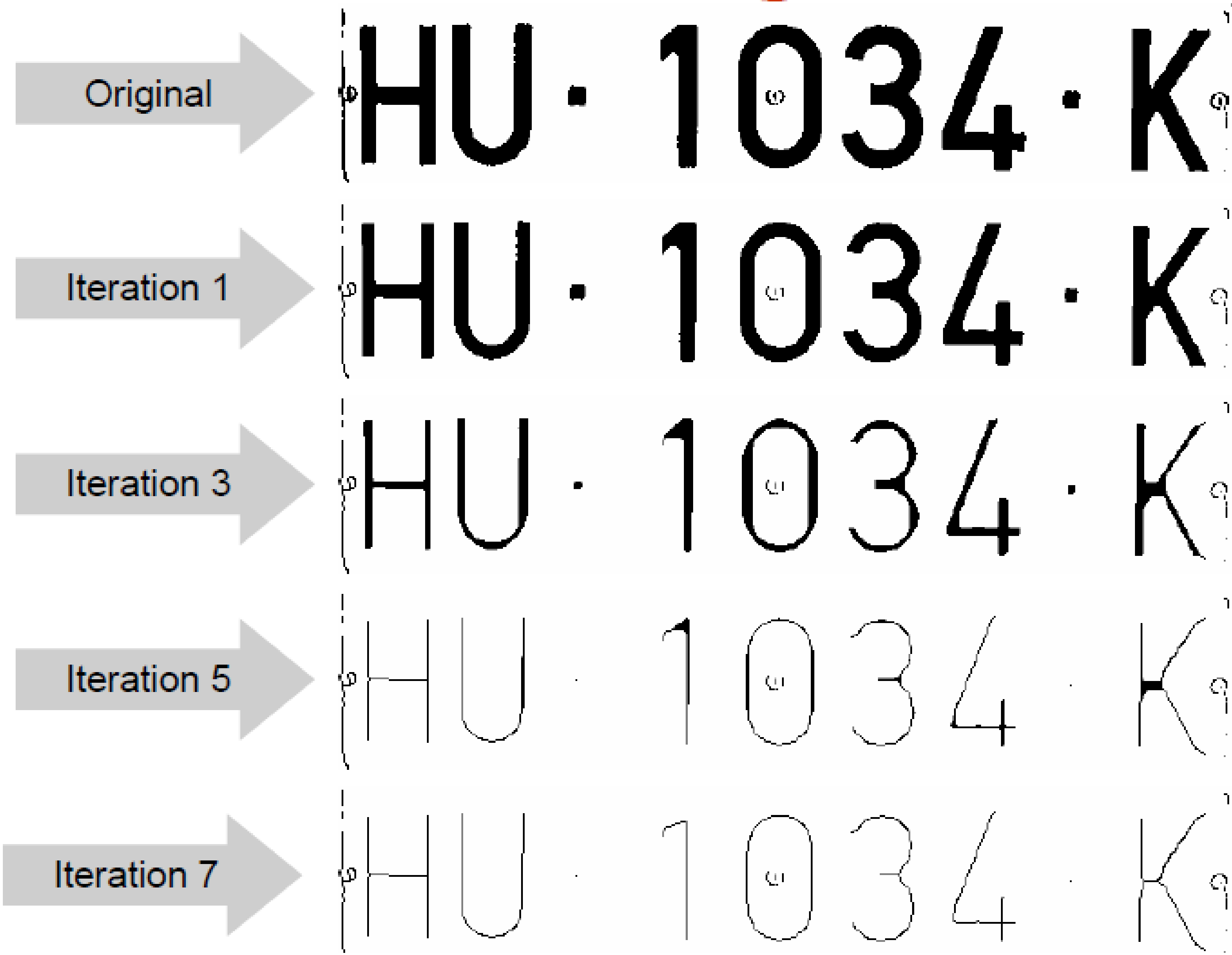


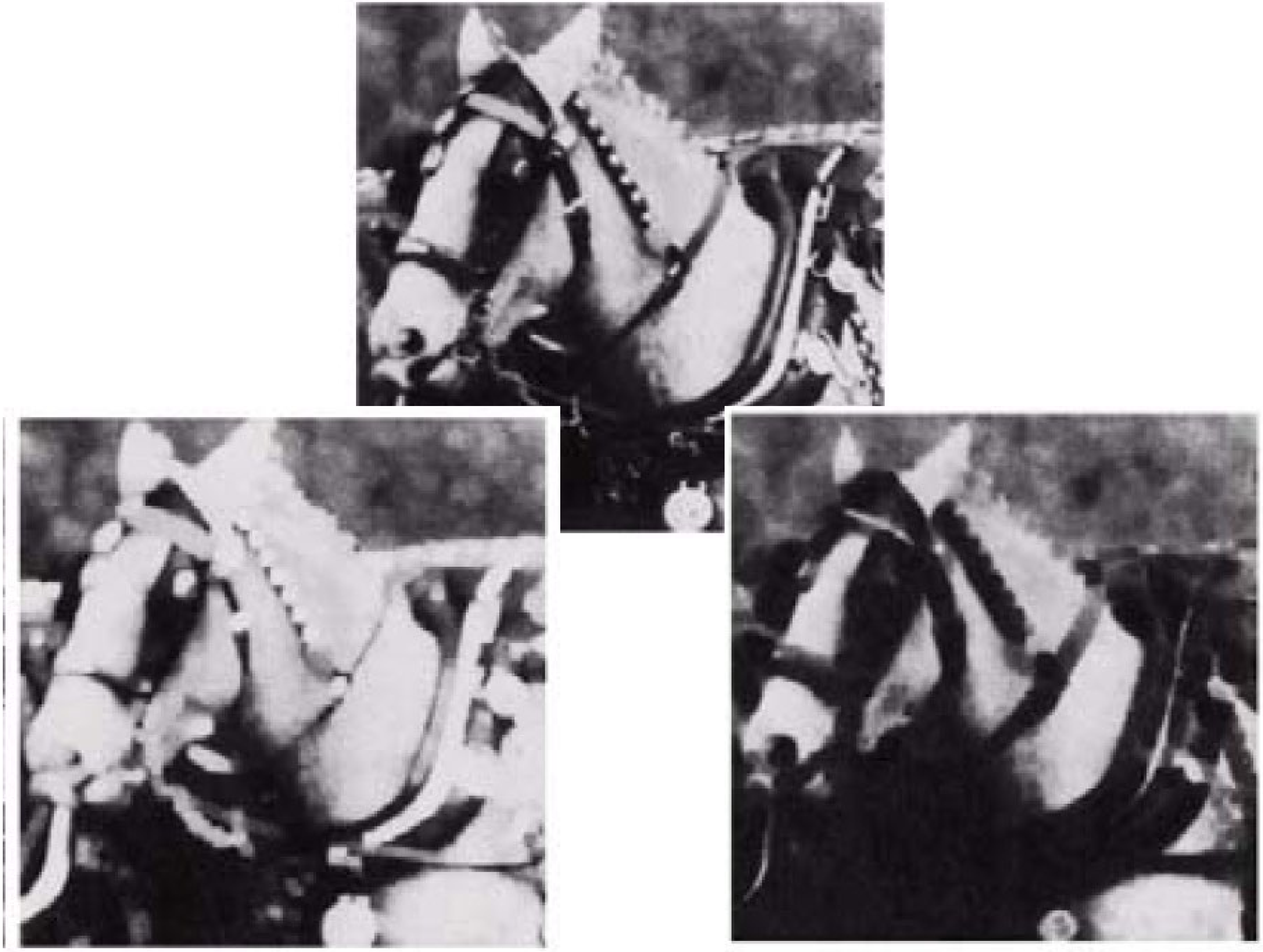
FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m-connectivity.


Original
HU 1034 K
Iteration 1
HU 1034 K
Iteration 3
HU 1034 K
Iteration 5
HU 1034 K
Iteration 7
HU 1034 K

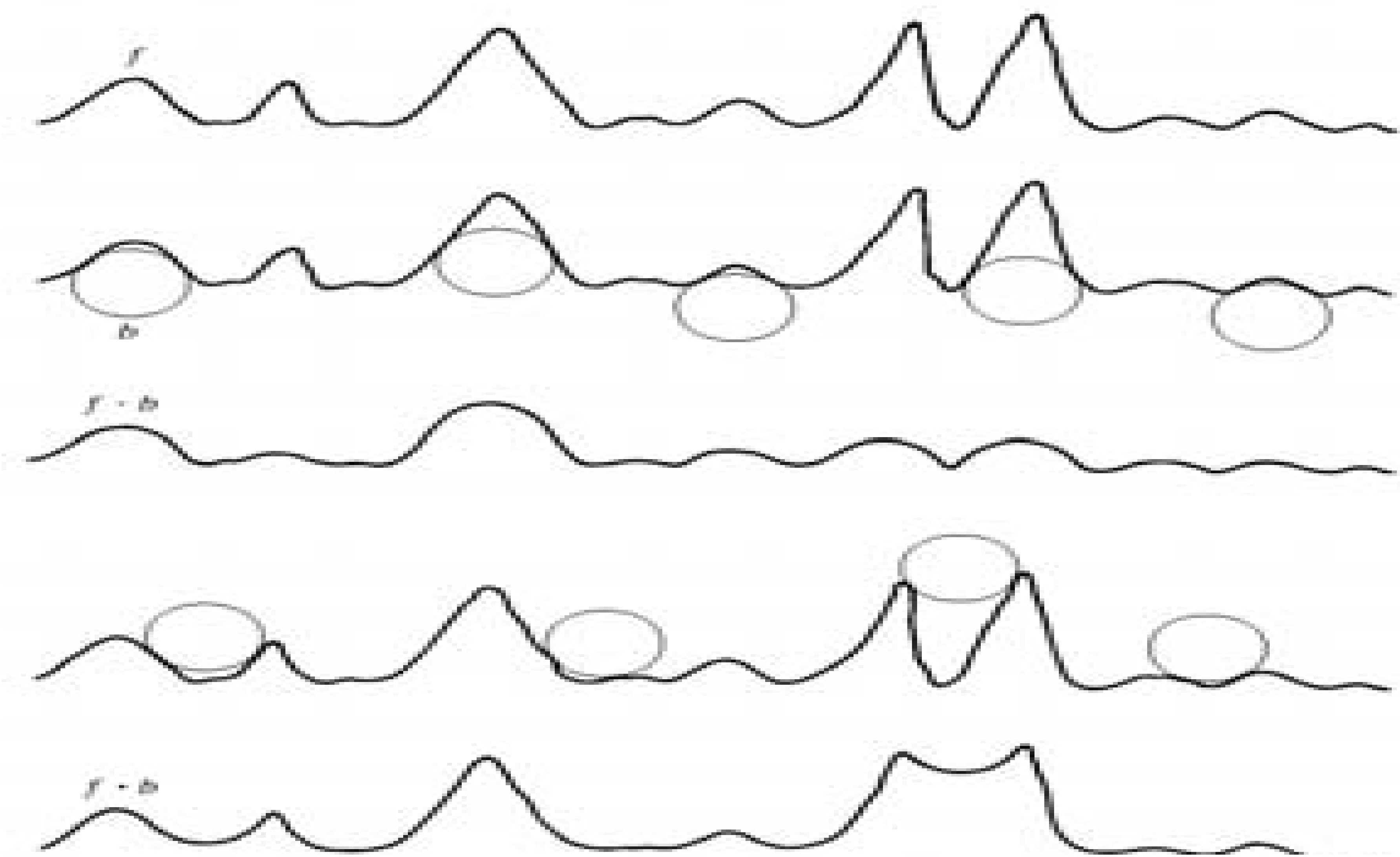
Morfologie – dilatace a eroze Šedotónová

DILATACE – v každé poloze se sečtou po prvcích hodnoty strukt. elementu a odpovídající části obrazu a nalezne se MAX

EROZE– v každé poloze se odečtou po prvcích hodnoty strukt. elementu od odpovídající části obrazu a nalezne se MIN

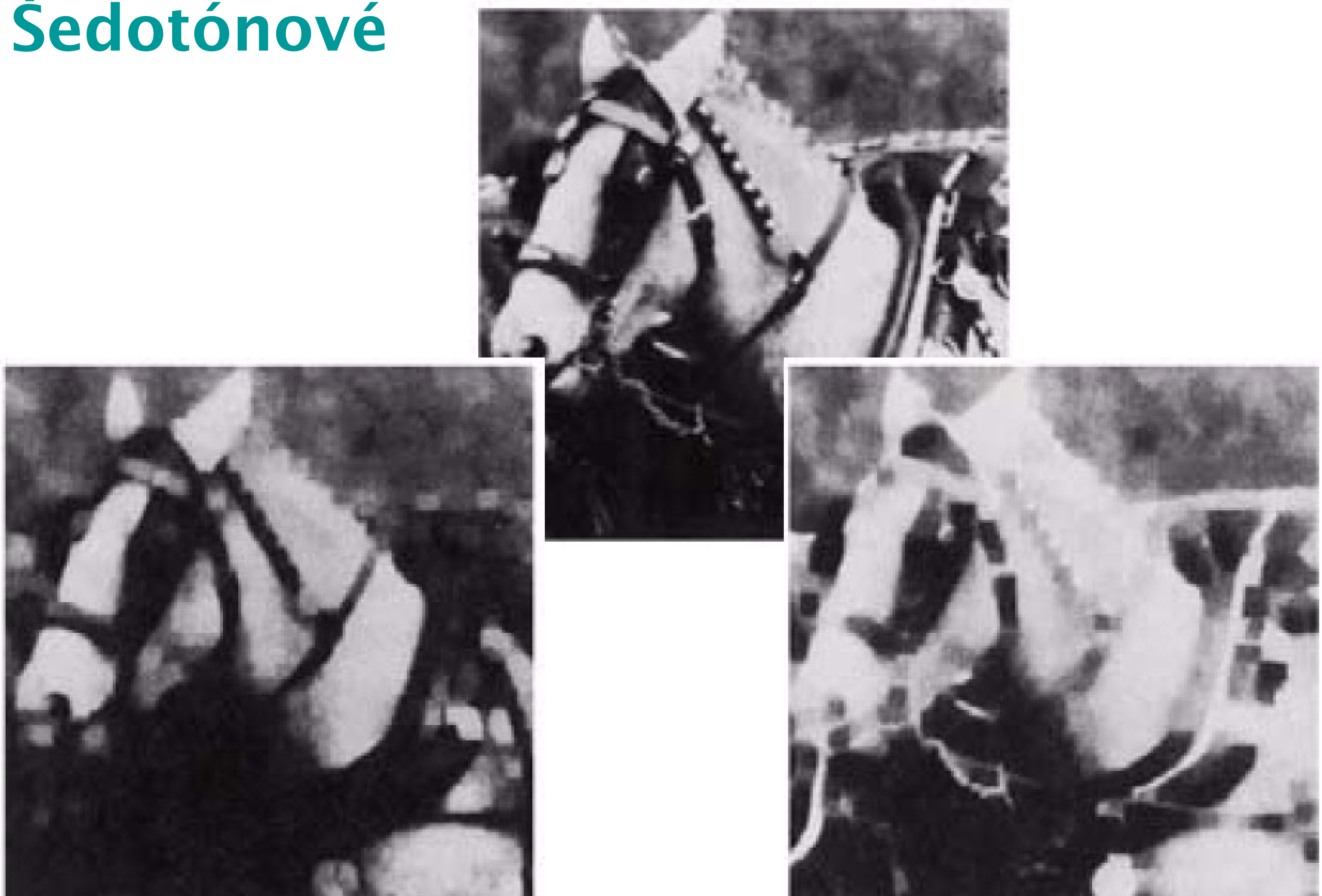


Morfologie – otevření a uzavření Šedotónové



Morfologie – otevření a uzavření

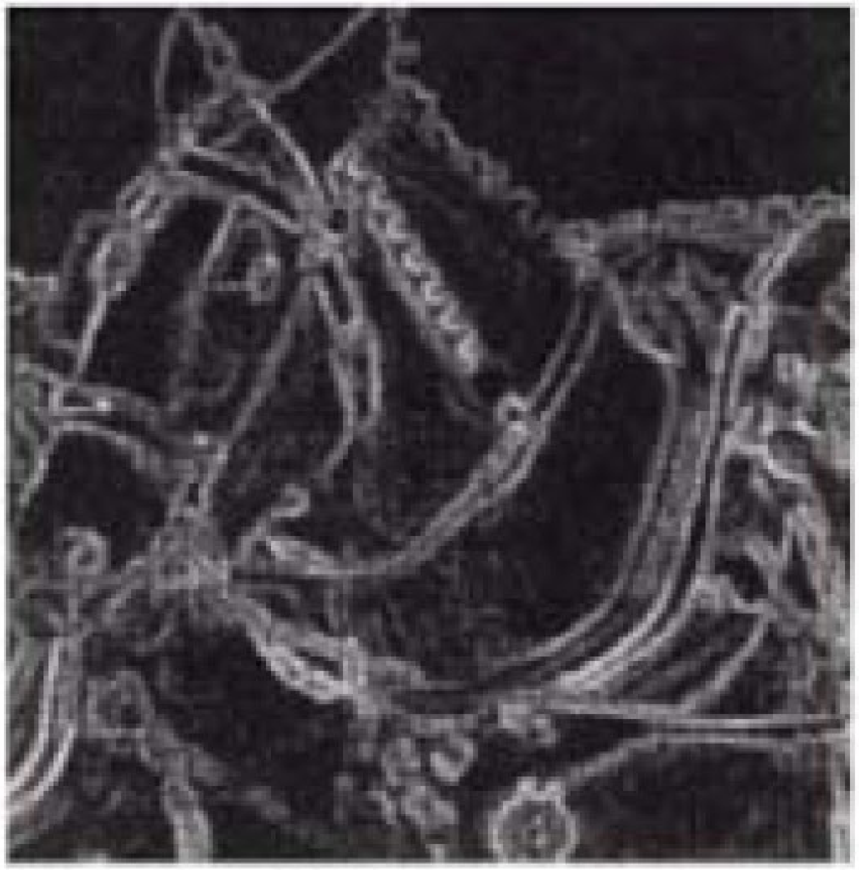
Šedotónové



Morfologie – vyhlazování Šedotónové

OTEVŘENÍ pak UZAVŘENÍ

Morfologie – gradient
Šedotónové



$$(A \oplus B) - (A \ominus B)$$

Morfologie

Erosion

IΘB

Dilatation

I⊕B

Closing I•B

= (I⊕B)ΘB

Opening IoB

= (IΘB)⊕B

Images courtesy of OpenCV tutorial at CVPR-2001