Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 11 – Image Segmentation 2

Winter Semester 2014/15

Dr. Benjamin Seppke
Prof. Siegfried Stiehl

Uniformity Assumption

Many segmentation procedures are based on a uniformity assumption:

- meaningful objects correspond to regions which satisfy a uniformity predicate
  - → region finding
- object boundaries correspond to discontinuities of a uniformity predicate
  - → edge finding

Typical uniformity predicates:

- greyvalues within a narrow interval (e.g. in B/W images)
- similar colour
- small greyvalue gradient
- uniform statistical properties (e.g. local distribution, texture)
- smoothness in 3D

Region Growing

Regions which satisfy a uniformity criterion may be grown from seed regions based on two criteria:

1. Merge region with new area if merged region satisfies uniformity criterion.

   E.g. greyvalue variance remains limited
2. Merge region with new area if boundary area satisfies a merging criterion.

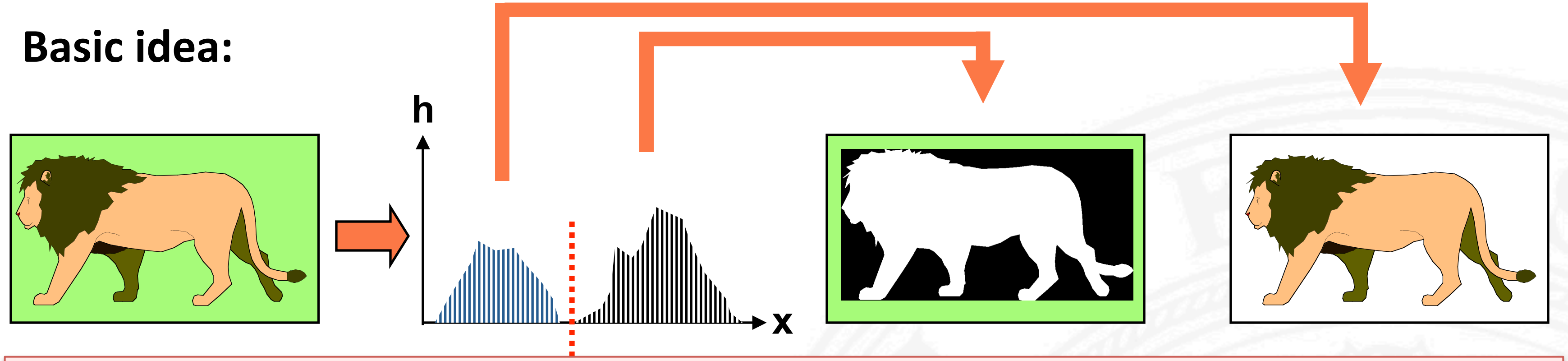
   E.g. boundary area has weak edges

Problem with (1): Large regions may be merged with small patches even if the patches are distinctly different.

Problem with (2): Distinct large regions may be merged if they are connected by a weak boundary.

Segmentation into Regions using Histograms

Basic idea:



Recursive histogram decomposition:

- compute 1D histograms of pixel features (e.g. R, G, B histograms)
- use "clearest" histogram for decomposition into regions
- apply procedure recursively to individual regions

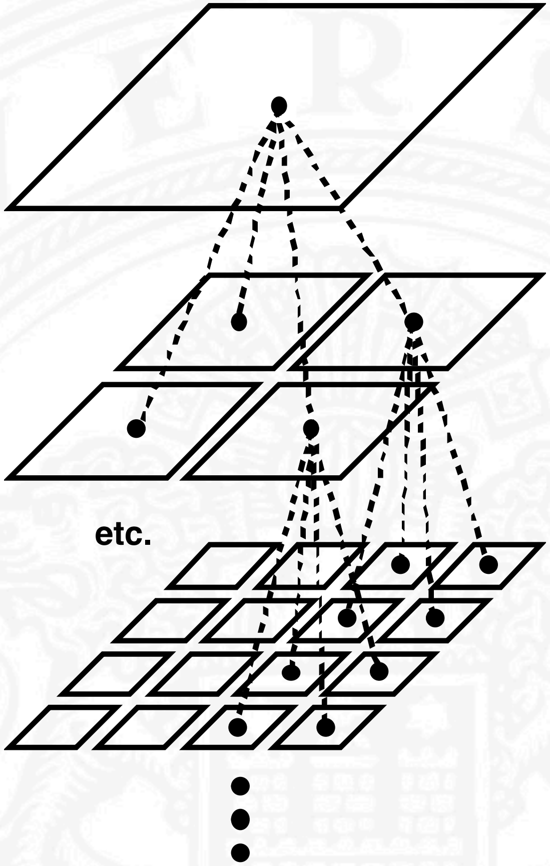
Problems:

- histograms do not reflect neighbourhood relationships
- histograms may not show multimodality clearly
- bad early decisions cannot be corrected

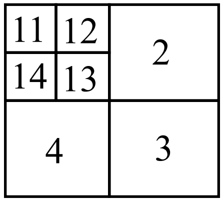
Region Segmentation by Split-and-merge

Region boundaries are determined along quadtree region boundaries.

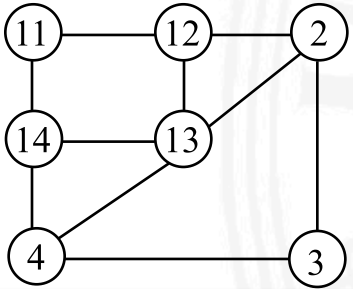
- Begin with an arbitrary region decomposition in a quadtree plane
- Split each region which violates a uniformity predicate into its 4 quadtree sons
- Merge (recursively) all regions which jointly satisfy a uniformity criterion



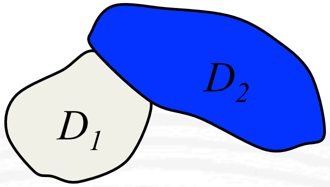
Supporting data structure:



Region adjacency graph



Maximum-likelihood Edge Finding



Hypothesis test about the likelihood of a boundary between two regions D_1 and D_2

H_0: Pixels from D_1 and D_2 stem from the same statistical source $N(\mu_0, \sigma_0)$.

H_{12}: Pixels from D_1 and D_2 stem from different statistical sources $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$.

Maximum-likelihood decision chooses hypothesis H_x for which P(g_{ij} are observed | H_x is true) is maximal.

Step 1: Maximum-likelihood estimation of $\mu_0, \sigma_0, \mu_1, \sigma_1, \mu_2, \sigma_2, D_0 = D_1 \cup D_2$

$$\hat{\mu}_k = \frac{1}{|D_k|} \sum_{g_{ij} \in D_k} g_{ij} \qquad \hat{\sigma}_k^2 = \frac{1}{|D_k|} \sum_{g_{ij} \in D_k} (g_{ij} - \hat{\mu}_k)^2 \qquad k = 0, 1, 2$$

Step 2: Determine likelihood quotient

$$\frac{\prod_{g \in D_0} P(g \mid H_0)}{\prod_{g \in D_1} P(g \mid H_{12}) \prod_{g \in D_2} P(g \mid H_{12})} > 1$$

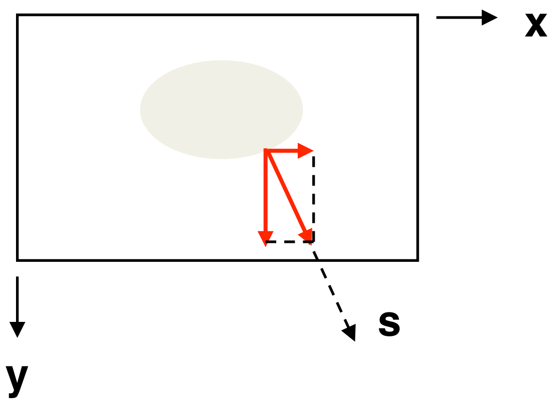
⇒

Decision rule: $\dfrac{\hat{\sigma}_1^{|D_1|} \; \hat{\sigma}_2^{|D_2|}}{\hat{\sigma}_0^{|D_0|}} > S$

S **to be determined empirically**

Greyvalue Discontinuities

Edges may be localized via the 1. and 2. derivative of the greyvalue function.



Gradient

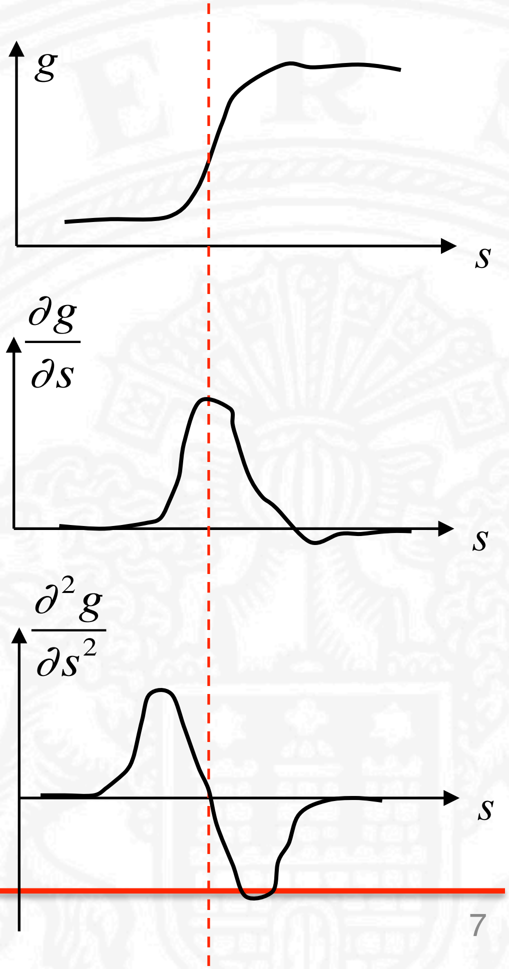
= vector in the direction of steepest increase

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

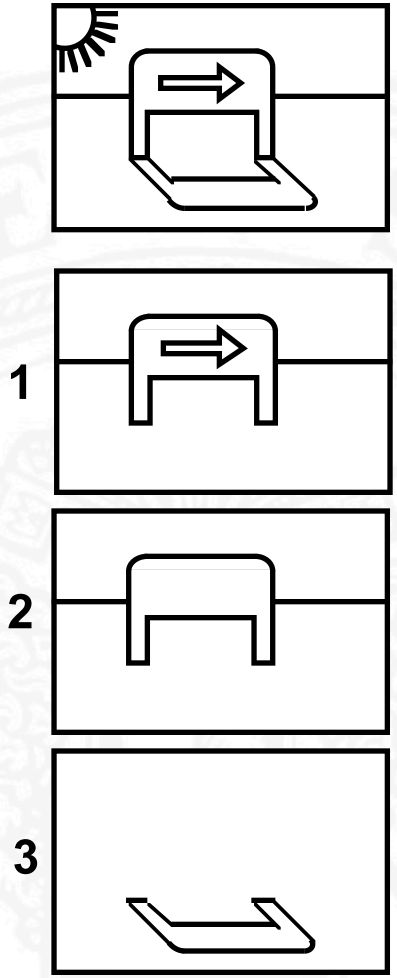
edges may be located at ...

... high gradient magnitudes ...

... zero crossings of the second derivative



Are Edges Object Boundaries?



Four reasons for edges in images:

1. Discontinuities of physical object surface properties

   e.g. colour, material, smoothness ("reflectivity")

2. Discontinuities of object surface orientation towards observer

   e.g. strong curvature, 3D-edges, specularities

3. Discontinuities of illumination

   e.g. shadows, secondary illumination

4. Discretization effects

   e.g. binarisation

Edges in Real-World Images

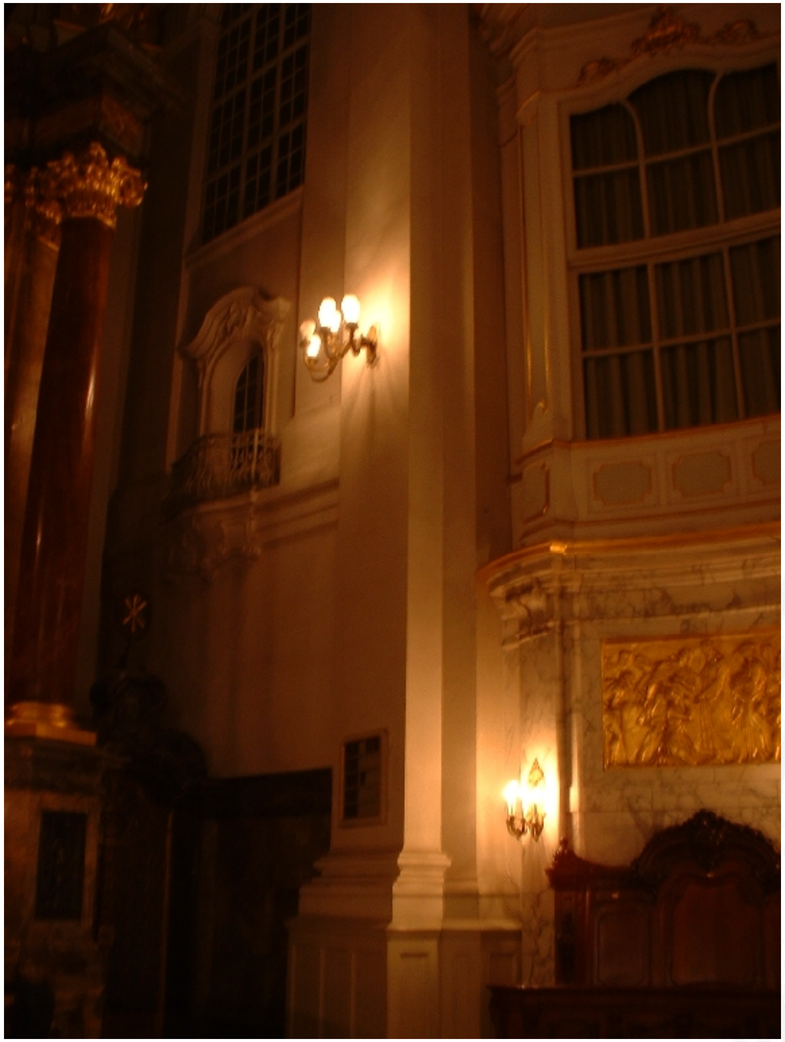


Image of Michaelis Church in Hamburg
(thanks to Wolfgang Förstner)

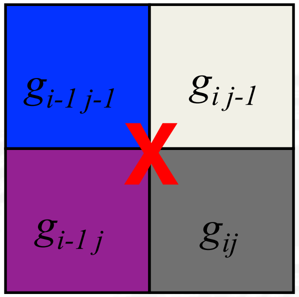
Consider vertical edge with lamps left and right:

In the lower part, the region left of the edge is darker than the region right of the edge, in the upper part vice versa.

→ In between, the edge must have no contrast at all!

Robert´s Cross Operator

Computes the gradient based on crosswise greyvalue differences



Gradient magnitude:

$$\left|\nabla g_{ij}\right| = \sqrt{(g_{i\ \mathrm{j\text{-}1}} - g_{i-1\ \mathrm{j}})^2 + (g_{i\ \mathrm{j}} - g_{i-1\ \mathrm{j\text{-}1}})^2}$$

$$\left.\begin{aligned} &\approx \left|g_{i\ \mathrm{j\text{-}1}} - g_{i-1\ \mathrm{j}}\right| + \left|g_{i\ \mathrm{j}} - g_{i-1\ \mathrm{j\text{-}1}}\right| \\ &\approx \max\left\{\left|g_{i\ \mathrm{j\text{-}1}} - g_{i-1\ \mathrm{j}}\right| , \left|g_{i\ \mathrm{j}} - g_{i-1\ \mathrm{j\text{-}1}}\right|\right\} \end{aligned}\right\} \text{ approximations}$$

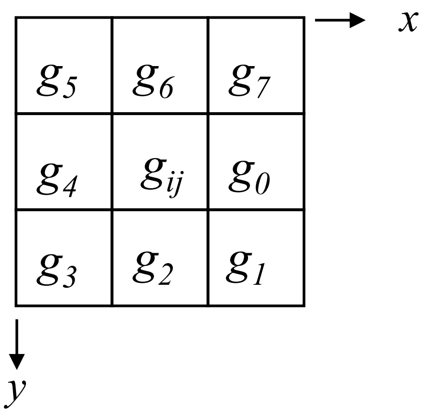
Gradient direction:

$$\tan\gamma = \frac{g_{i\ \mathrm{j}} - g_{i-1\ \mathrm{j\text{-}1}}}{g_{i\ \mathrm{j\text{-}1}} - g_{i-1\ \mathrm{j}}}$$

direction angle γ in coordinate system rotated by 45^0

Sobel Operator

Popular operator contained in most image processing software packages



g_5	g_6	g_7
g_4	g_{ij}	g_0
g_3	g_2	g_1

- **Computes gradient components Δx and Δy based on pixels taken from a 3x3 neighbourhood.**
- **Performs simultaneous smoothing**

Partial derivatives:

$$\Delta g_x = \left(g_1 + 2g_0 + g_7\right) - \left(g_3 + 2g_4 + g_5\right)$$

$$\Delta g_y = \left(g_1 + 2g_2 + g_3\right) - \left(g_7 + 2g_6 + g_5\right)$$

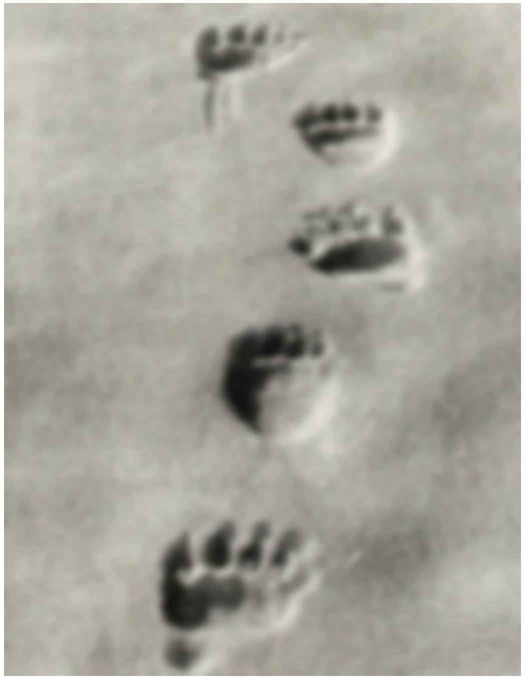
Gradient magnitude:

$$\left|\nabla g_{ij}\right| = \sqrt{\Delta g_x^2 + \Delta g_y^2}$$

Gradient direction:

$$\tan\ \gamma = \frac{\Delta \mathrm{g_y}}{\Delta \mathrm{g_x}}$$

Example for Sobel Operator

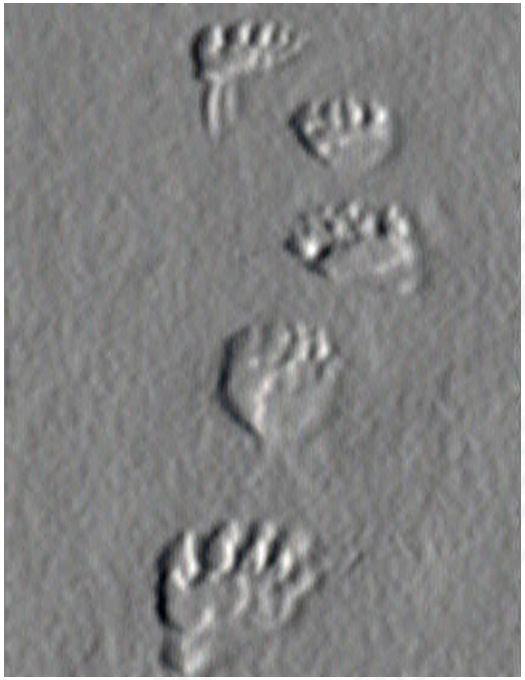


$g(x, y)$

greyvalue image

0 = black

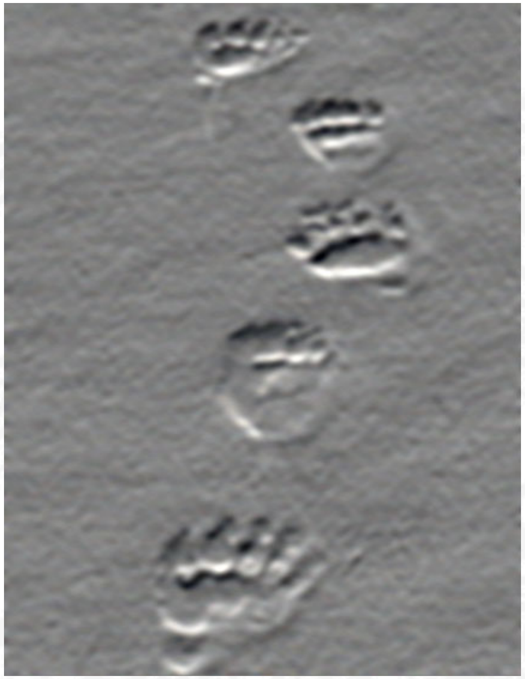
255 = white



Δg_x

x-component of greyvalue gradient

0 = greyvalue 128



Δg_y

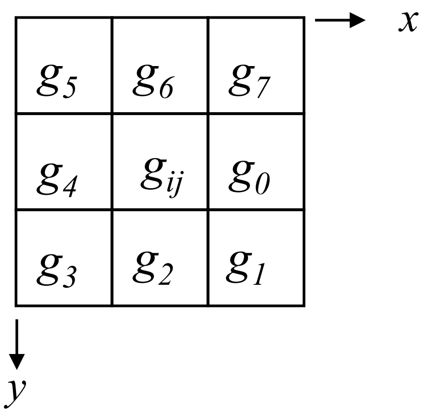
y-component of greyvalue gradient

0 = greyvalue 128

Kirsch Operator

Another popular operator

- **Computes gradient magnitude in 8 directions, selects maximum**
- **Performs simultaneous smoothing**

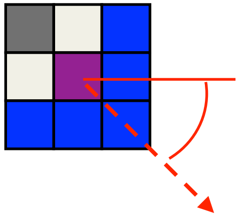


g_5	g_6	g_7
g_4	g_{ij}	g_0
g_3	g_2	g_1

Gradient magn.: $|\nabla g_{ij}| = \max\limits_{k=0..7 \bmod 8} \left\{ 3\left(g_k + g_{k+1} + g_{k+2} + g_{k+3} + g_{k+4}\right) - 5\left(g_{k+5} + g_{k+6} + g_{k+7}\right) \right\}$

Gradient direction: $\gamma = \left(90° + k_{\max} \cdot 45°\right) \bmod 360°$

Example:



$k_{\max} = 7$

$\gamma = \left(90° + 7 \cdot 45°\right) \bmod 360° = 45°$

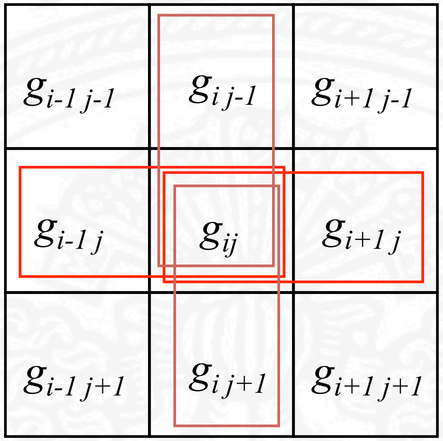
Laplacian Operator

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

Orientation-independent measure for the strength of the second derivative of a greyvalue function

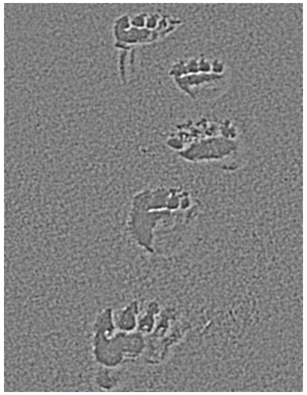
Discrete approximation by differences of differences of greyvalues:

$$\begin{aligned}\nabla^2 g_{i,j} = {} & \left(g_{i+1,j} - g_{i,j}\right) - \left(g_{i,j} - g_{i-1,j}\right) \\ + {} & \left(g_{i,j+1} - g_{i,j}\right) - \left(g_{i,j} - g_{i,j-1}\right) \\ = {} & g_{i+1,j} + g_{i-1,j} + g_{i,j+1} + g_{i,j-1} - 4g_{i,j}\end{aligned}$$



$g_{i-1\,j-1}$	$g_{i\,j-1}$	$g_{i+1\,j-1}$
$g_{i-1\,j}$	g_{ij}	$g_{i+1\,j}$
$g_{i-1\,j+1}$	$g_{i\,j+1}$	$g_{i+1\,j+1}$

"difference between the greyvalue of a point and the average of its surrounding"



Using the Laplacian operator on raw images will typically give unacceptable results since the 2. derivative amplifies noise. (A single isolated point generates the maximal response.)

Marr-Hildreth Operator

Locates edges at zero crossings of second derivative of smoothed image

Laplacian of Gaussian (LoG): $\nabla^2 \left(f(x,y,\sigma) * g(x,y) \right)$

with Gaussian filter: $f(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$

Interchanging the order of differentiation and convolution in the LoG gives

$$\nabla^2 \left(f(x,y,\sigma) \right) * g(x,y) = h(x,y) * g(x,y)$$

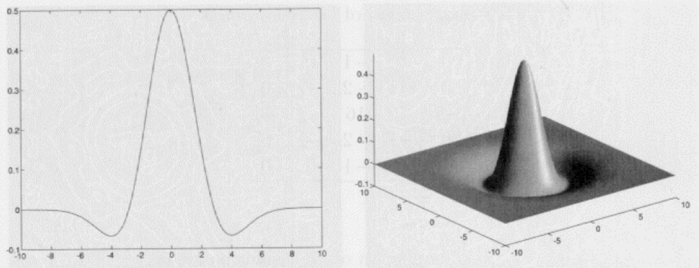
$$h(x,y) = c\left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

c normalizes the sum of mask elements to zero

discrete 5 x 5 approximation:

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

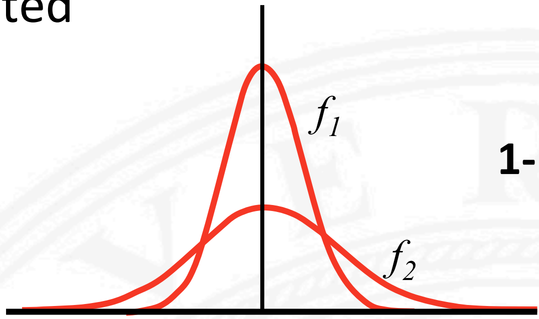
Nickname: Mexican Hat Operator



Difference of Gaussians (DoG)

The Marr-Hildreth Operator can be approximated by the difference of 2 Gaussians:

$$h(x, y) = f_1(x, y) - f_2(x, y)$$

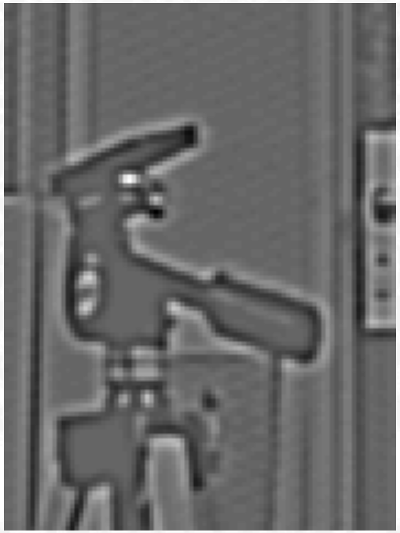


1-D DoG

The best approximation of the Laplacian is for $\sigma_2 \approx 1.6\, \sigma_1$

original image

result of DoG filtering with $\sigma_1 = 1,\ \sigma_2 = 1.6$



Canny Edge Detector I

Optimal edge detector for step edges corrupted by white noise.

Optimality criteria:

- **Detection of all important edges and no spurious responses**
- **Minimal distance between location of edge and actual edge**
- **One response per edge only**

1. Derivation for 1D results in edge detection filter which can be effectively approximated (< 20% error) by the 1st derivative of a Gaussian smoothing filter.
2. Generalization to 2D requires estimation of edge orientation:

$$\vec{n} = \frac{\nabla(f * g)}{|\nabla(f * g)|}$$

$\vec{n}$ normal perpendicular to edge
f Gaussian smoothing filter
g greyvalue image

Edge is located at local maximum of g convolved with f in direction $\vec{n}$:

$$\frac{\partial^2}{\partial \vec{n}^2} f * g = 0$$ "non-maximal suppression"

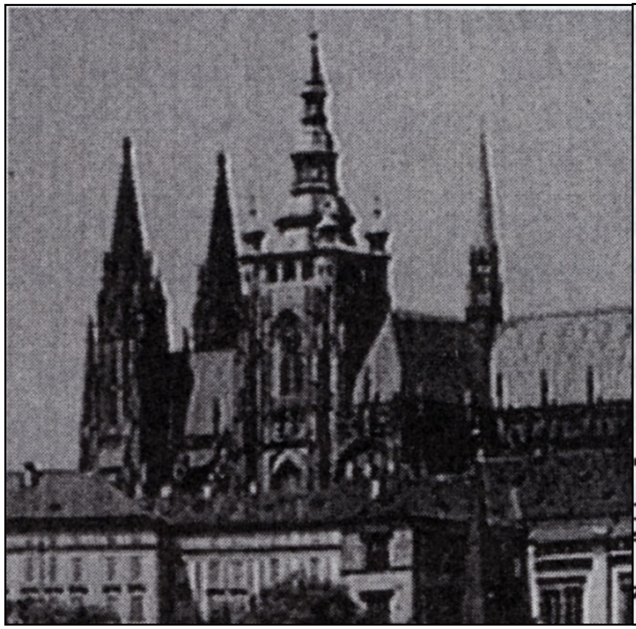
Canny Edge Detector II

Algorithm includes

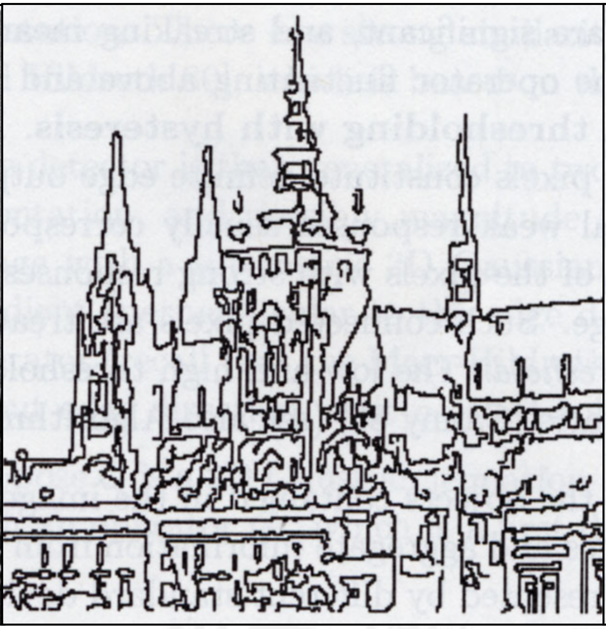
- choice of scale σ
- hysteresis thresholding to avoid streaking (breaking up edges)
- "feature synthesis" by selecting large-scale edges dependent on lower-scale support

1. Convolve image g with Gaussian filter f of scale σ
2. Estimate local edge normal direction γ for each point in the image
3. Find edge locations using non-maximal suppression
4. Compute magnitude of edges
5. Threshold edges with hysteresis to eliminate spurious edges
6. Repeat steps (1) through (5) for increasing values of σ
7. Aggregate edges at multiple scales using feature synthesis

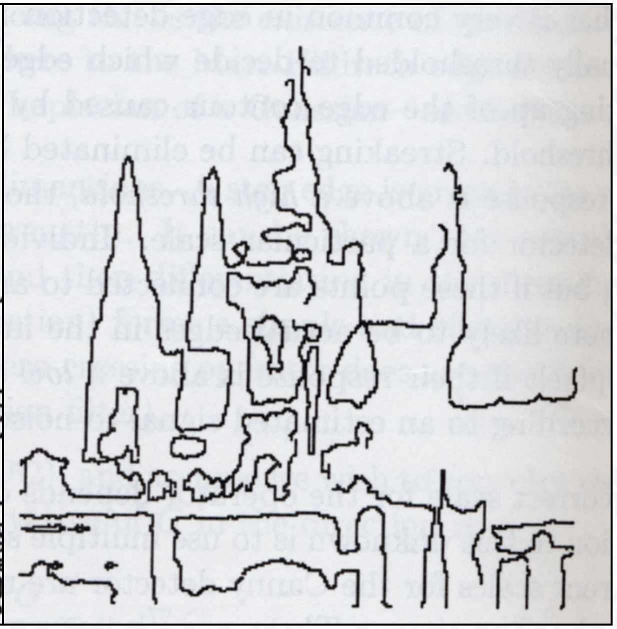
Examples for Canny Edge Detector



original



Canny operator $\sigma = 1.0$

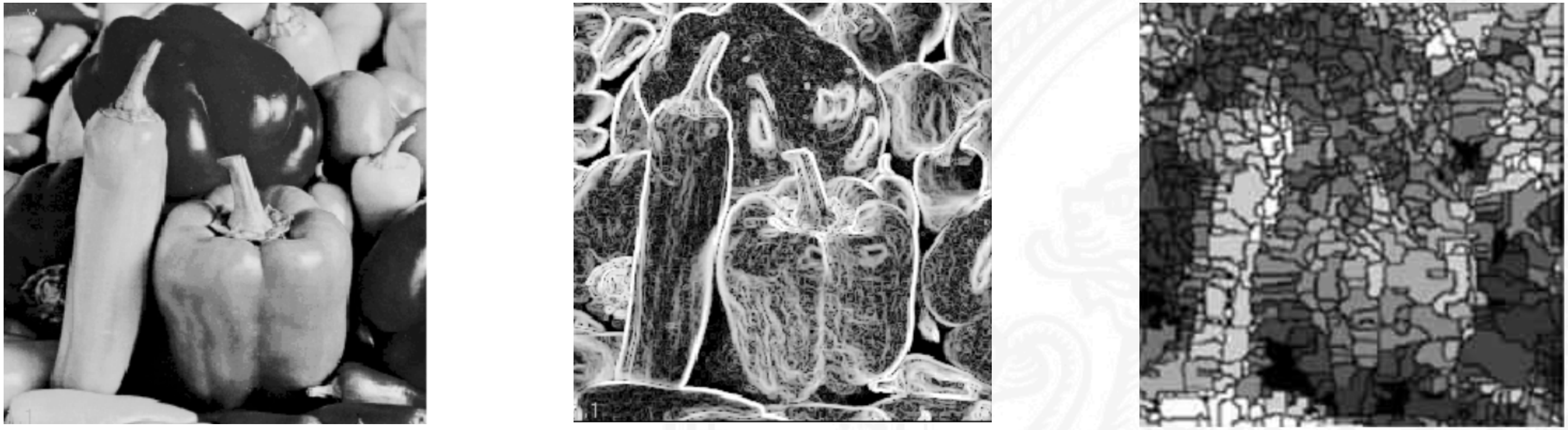


Canny operator $\sigma = 2.8$
(without feature synthesis)

Watershed Segmentation

Basic idea:

- Determine gradient magnitude "image" and visualize it as a 3D topographic map with small gradient values as low areas and large gradient values as hills.
- Determine watershed lines and record them as region boundaries



Typical result is an over-segmentation, can be reduced by region merging.

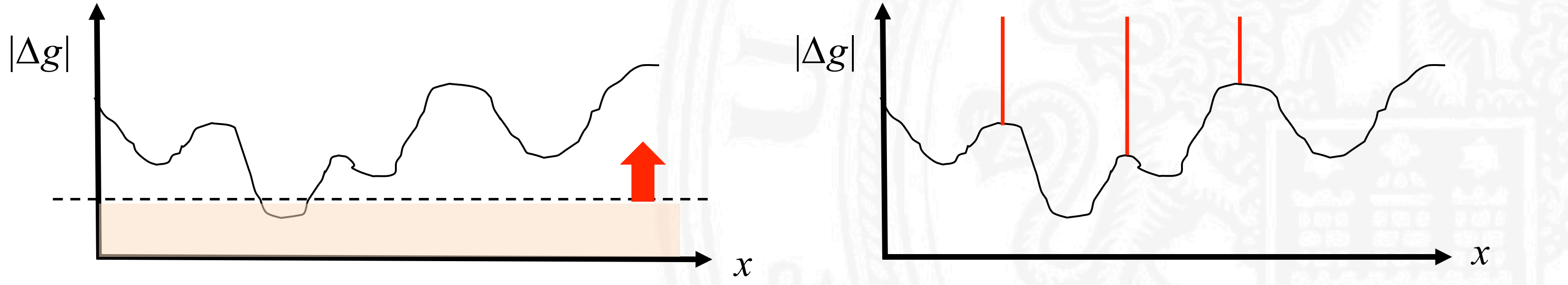
Principle of Watershed Computation

A watershed is the line between drainage basins where water is equally likely to flow down into either basin.

Idea for computing watersheds [Vincent and Soille 91]:

- Determine local minima as basin seed points
- Assume holes drilled at seed points and a water level rising from below, filling basins
- Establish watershed lines (= region boundaries) at locations where separate basins merge

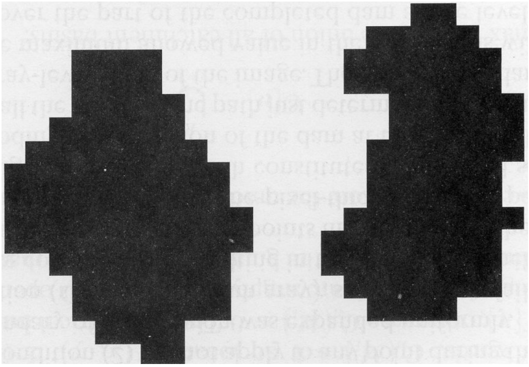
1D-Illustration



Determining Watershed Lines

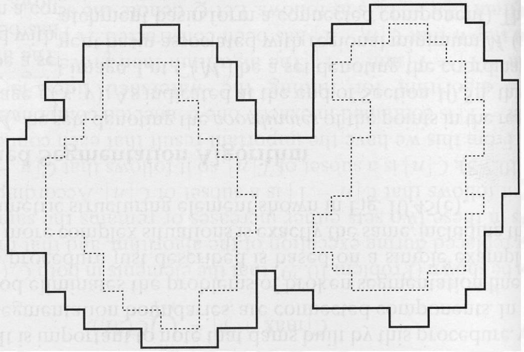
Assume that two basins are separate at step n-1 and merge at step n:

$I^{(n-1)}$



$C_i^{(n-1)}$ $C_j^{(n-1)}$

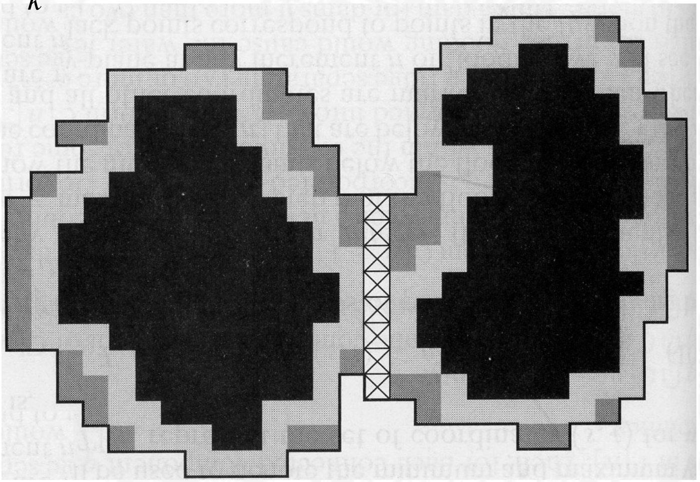
$I^{(n)}$



$C_k^{(n)}$

S $\boxed{\begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix}}$

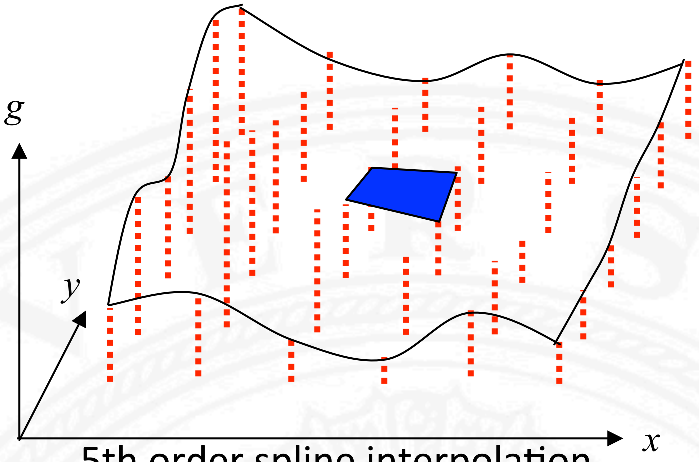
- Determine watershed pixels by dilation of the binary component image $I^{(n-1)}$ using structuring element S until regions overlap.
- Restrict dilation so that center of S is only in $C_k^{(n)}$.



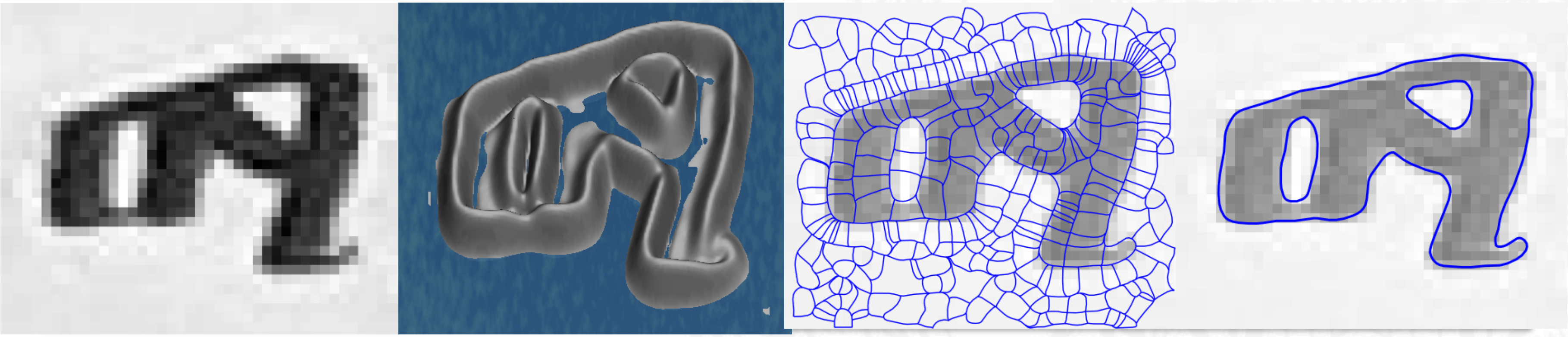
Subpixel Watershed Segmentation

(Meine and Koethe 05)

1. Generate continuous image by spline interpolation between pixels of original image
2. Determine gradient magnitude image by differentiating the (analytical) continuous image
3. Trace maxima in gradient image (watersheds)
4. Remove weak edges

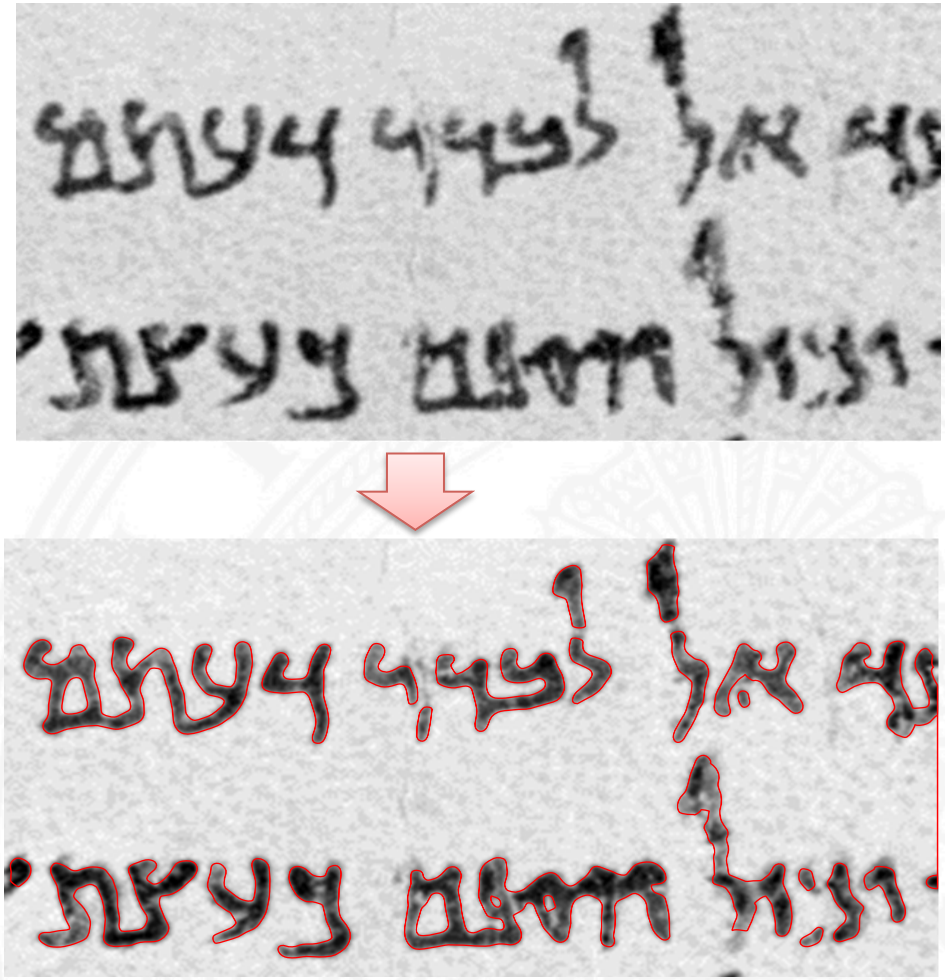


5th order spline interpolation



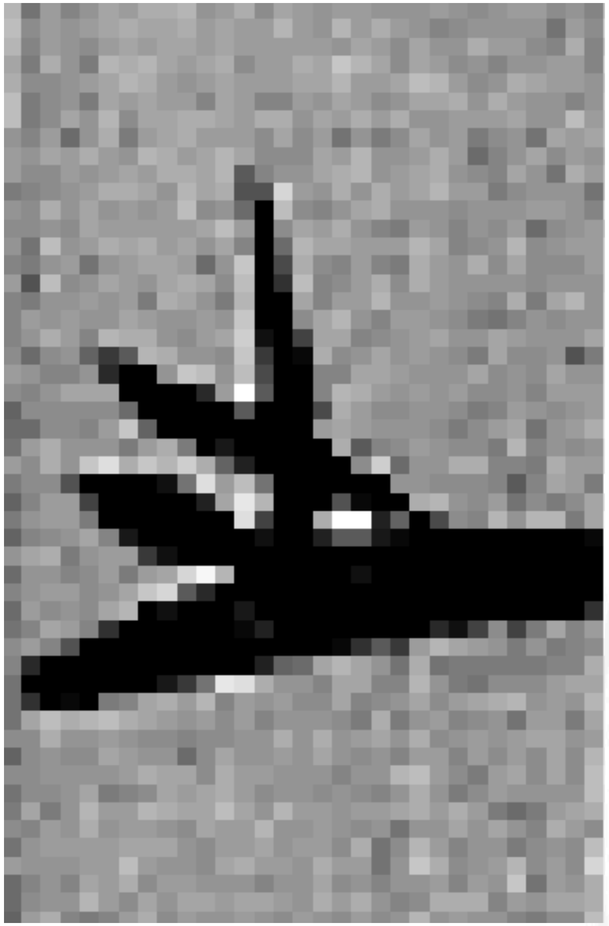
Applications in Manuscript Analysis I

segment of dead-sea scrolls

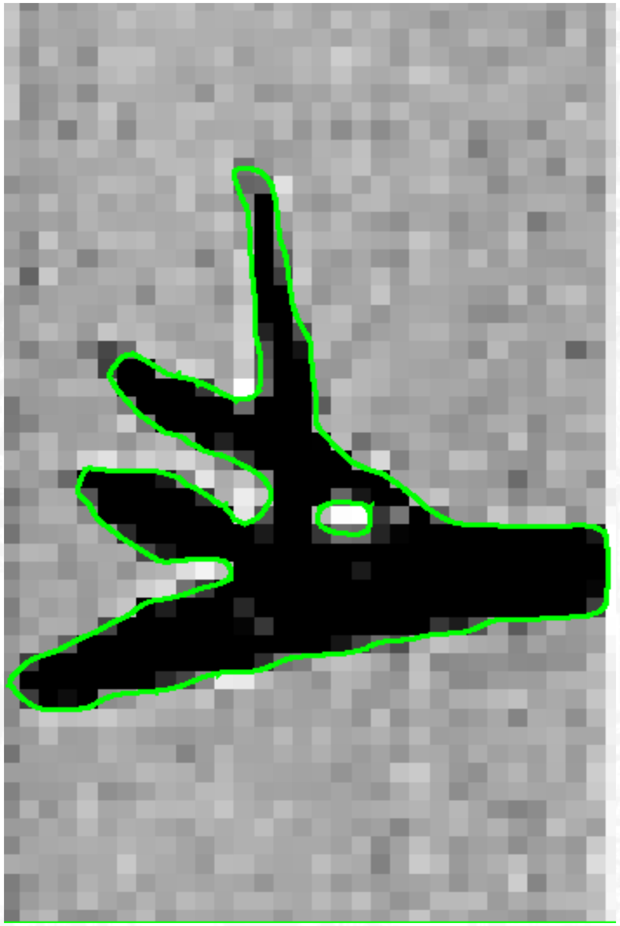


subpixel watershed segmentation

Applications in Manuscript Analysis II



Original



Subpixel Watershed Contours