Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

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

Image Processing 1 (IP1)
Bildverarbeitung 1

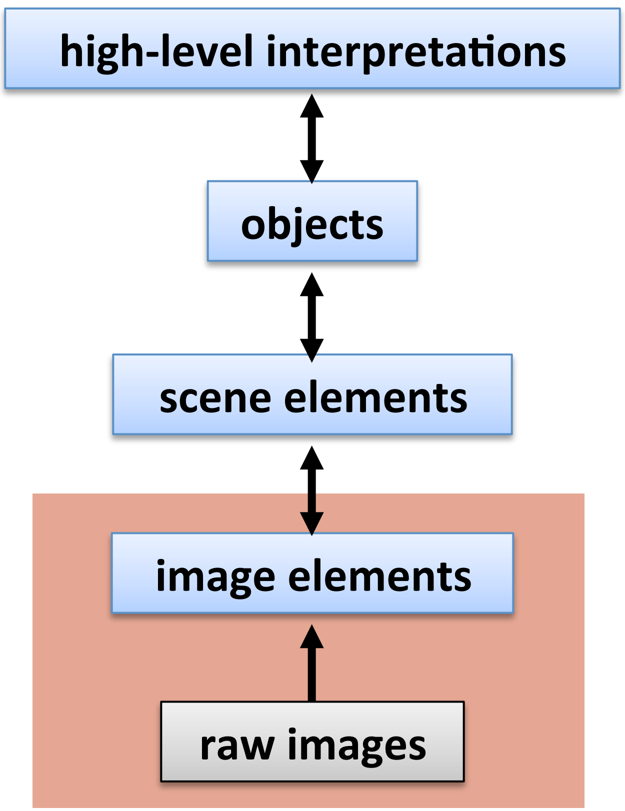
Lecture 10 – Image Segmentation 1

Winter Semester 2014/15

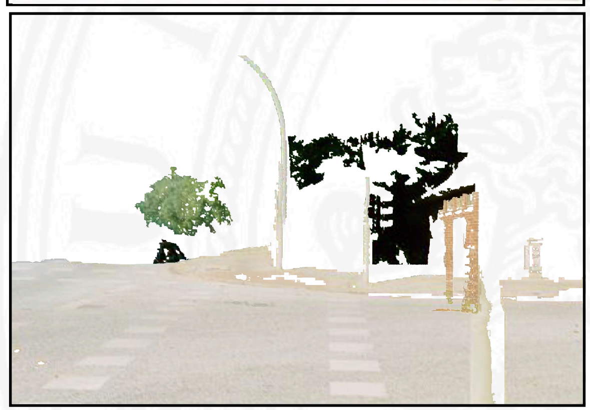
Dr. Benjamin Seppke
Prof. Siegfried Stiehl

Segmentation

Segmenting the image into image elements which may correspond to meaningful scene elements

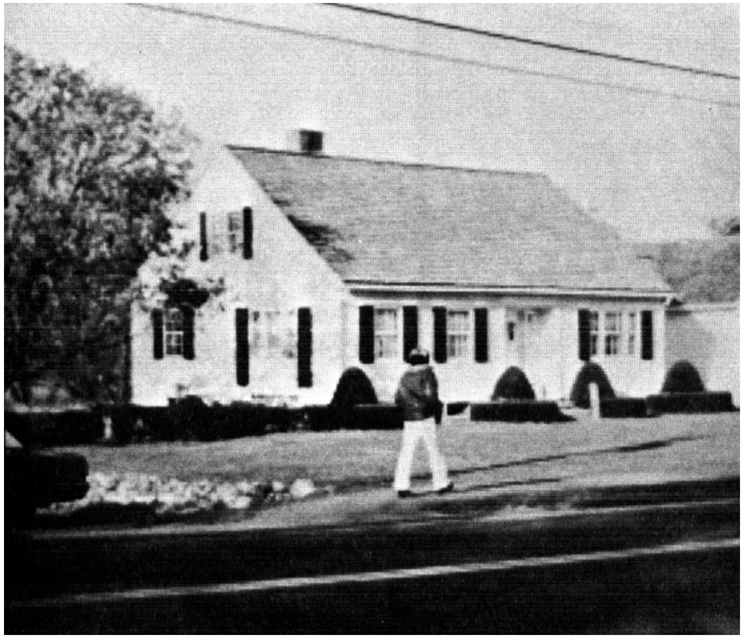


Example:
Partitioning an image into regions which may correspond to objects

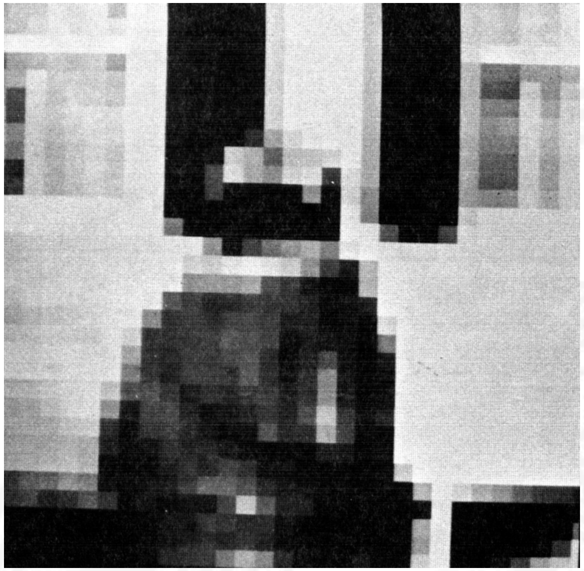


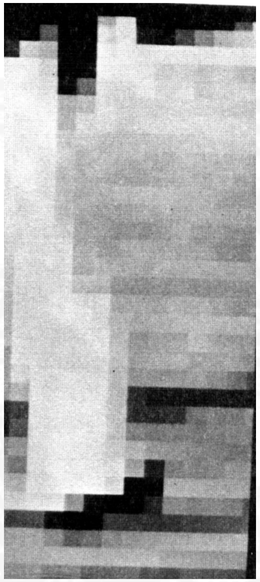
Typical results of first segmentation steps

Problems with Segmentation



landhouse scene





upper part and leg of person

Greyvalues of foreground may be indistinguishable from greyvalues of background.

In general, context knowledge is necessary for successful segmentation!

Primary Goal of Segmentation

"Segmenting an image into image elements which may correspond to meaningful scene elements"

What sort of image elements may correspond to meaningful scene elements?

Answer depends on type and complexity of images: Less constrained scenes must be segmented more conservatively.

Segmentation into ...

... entire objects	e.g. for	printed character recognition industrial object recognition medical cell analysis
... edge lines	e.g. for	aerial image analysis indoor scenes
... edge elements, vertices, groupings	e.g. for	natural scenes

Secondary Goals of Segmentation

Multiple resolutions for subsequent processes

- coarse resolution description for e.g.
  - analysis of image layout (horizon, foreground, background)
  - control of attention
  - planning a detailed analysis
- fine resolution description e.g. for
  - details
  - stereo analysis
  - motion analysis

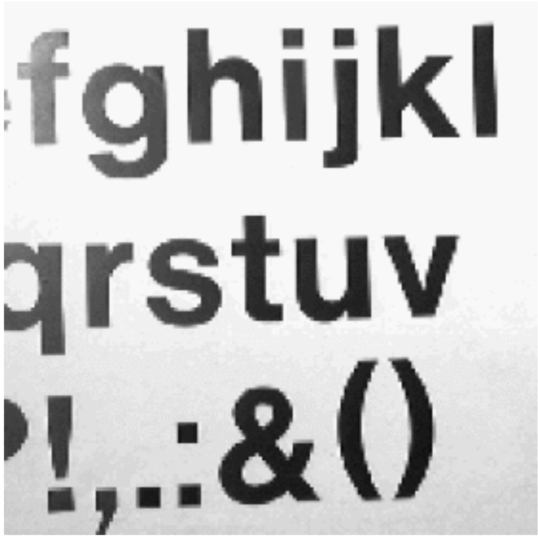
Data reduction

- Because of their large data volume, raw images are inconvenient as basic data structures for image analysis, e.g.:

TV colour image	3 x 512 x 576 ≈ 7 MB
10 sec TV colour images	10 x 25 x 7 ≈ 1750 MB

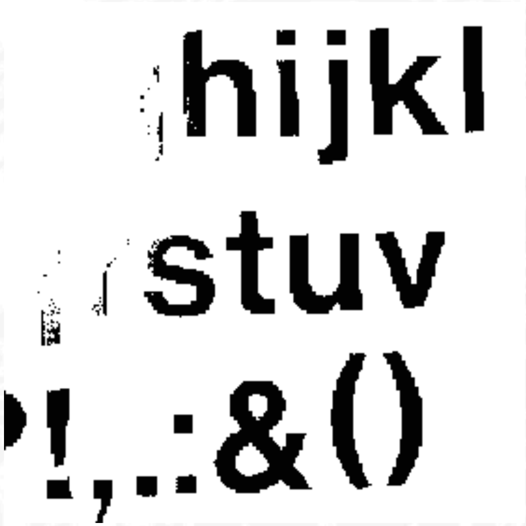
Thresholding

Thresholding has been introduced as a discretization technique. The same techniques can be applied for segmentation.



greyvalue image

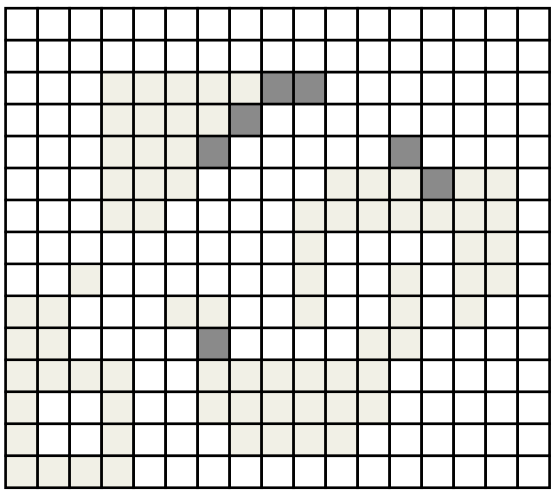
threshold too low



threshold too high

Representing Regions

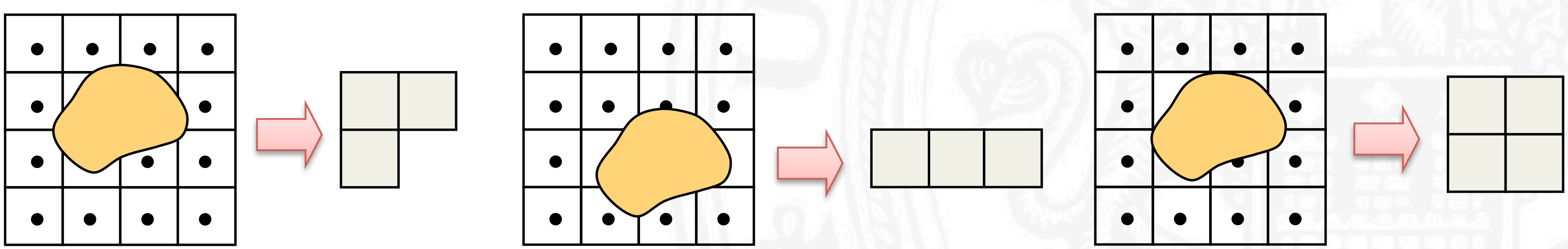
A region is a maximal 4- (or 8-) connected set of pixels.



Methods for digital region representation:

- grid occupancy
  - labelling
  - run-length coding
  - quadtree coding
  - cell sets
- boundary description
  - chain code
  - straight-line segments, polygons
  - higher-order polynomials

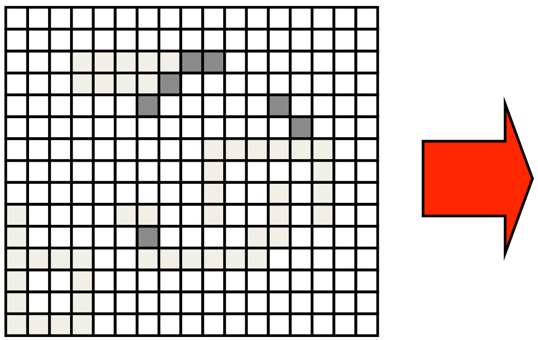
Note that discretizations of an analog region are not shift or rotation invariant:



Component Labelling

> **In this example: component descriptions using run-length coding**

Determining connected regions in B/W images



Component 1
(2 3 9)(3 3 7)(4 6 6)

Component 2
(4 12 12)

Component 3
(5 13 13)(6 9 14)(7 9 9 14 14)(8 9 9 14 14)(9 9 9 14 14)

Component 4
(9 0 0)(10 0 0)(11 0 3)(12 0 0 3 3)(13 0 0 3 3)(14 0 0 3 3)

Component 5
(9 5 6 12 12)(10 6 6 11 12)(11 6 11)

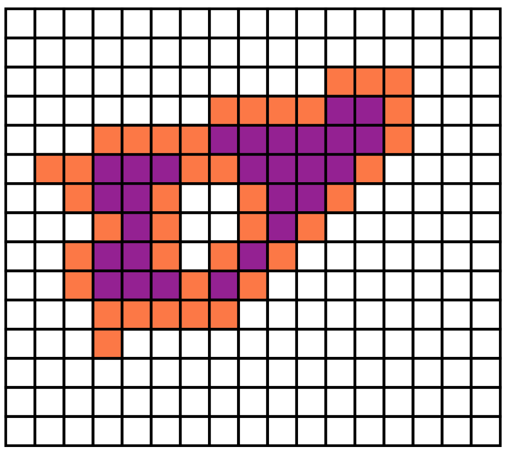
Component labelling of B/W images with 4-neighbourhood

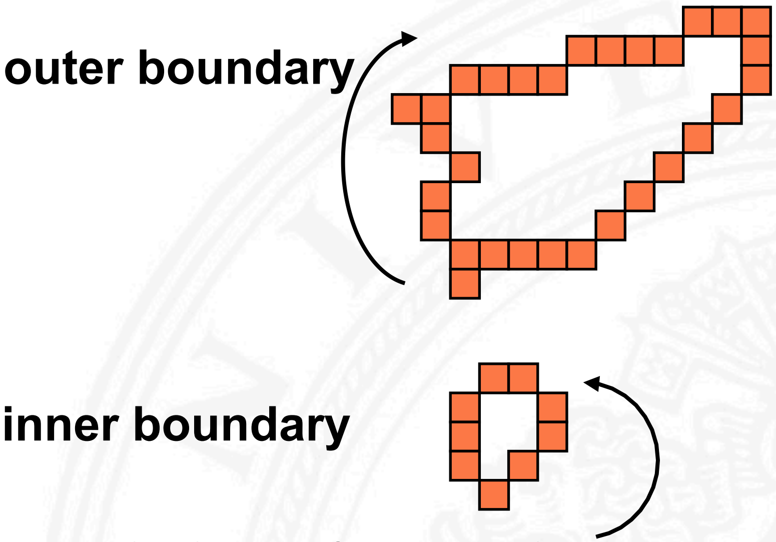
Scan image left to right, top to bottom:
- if pixel is white then continue
- if pixel is black then
  - if left neighbour is white and upper neighbour is white then assign new label
  - if left neighbour is black and upper neighbour is white then assign left label
  - if left neighbour is white and upper neighbour is black then assign upper label
  - if left neighbour is black and upper neighbour is black then
    - assign left label, merge left label and upper label

Boundaries

For a 4- (8-) connected region R the boundary is defined as the set of pixels of R which are 8- (4-) connected to the complement R^c of R.

Example for 8-connectivity:





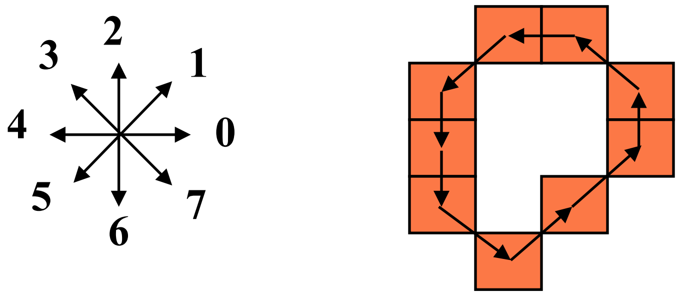
Boundary pixels are usually ordered clockwise for outer boundaries and counter-clockwise for inner boundaries.

- <u>Disadvantage</u> of this boundary definition:
- R and R^c have different boundaries - but nothing is in between.

Chain Code

Chain code represents boundaries by "chaining" direction arrows between successive boundary elements.

Chain code for <u>8-connectivity</u>:



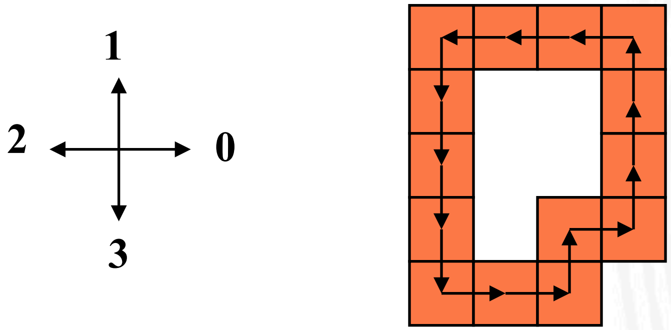
Arbitrary choice of starting point, chain code can be represented e.g. by

{456671123}

Normalization by circular shift until the smallest integer is obtained:

{112345667}

Chain code for <u>4-connectivity</u>:



Arbitrary starting point:

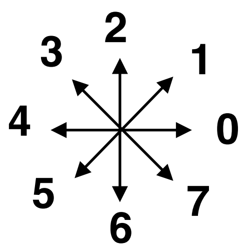
{22233330010111}

Normalized:

{00101112223333}

Chain Code Derivatives

Chain code is highly susceptible to discretization noise. Hence derived properties are usually also noisy.

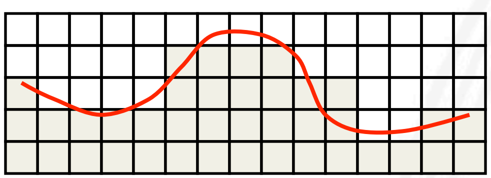


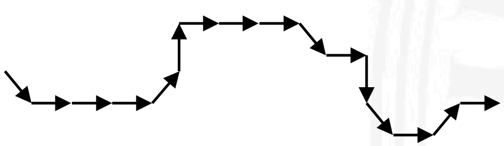
Slope:

chain code	0	1	2	3	4	5	6	7
tan θ	0	1	±∞	-1	0	1	±∞	-1
θ	0	45	90	135	±180	-135	-90	-45

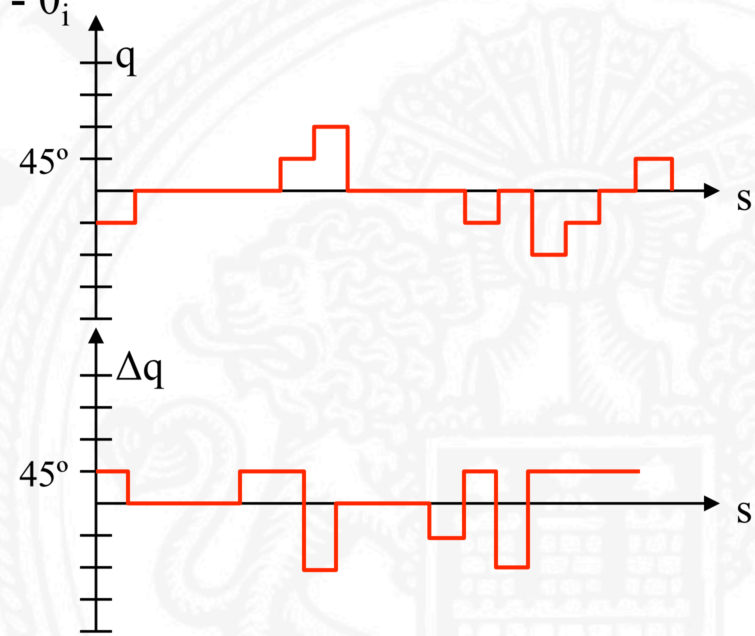
Curvature: $\Delta\theta = \theta_{i+1} - \theta_i$

Example:





{7000120007067010}



k-Slope and k-Curvature

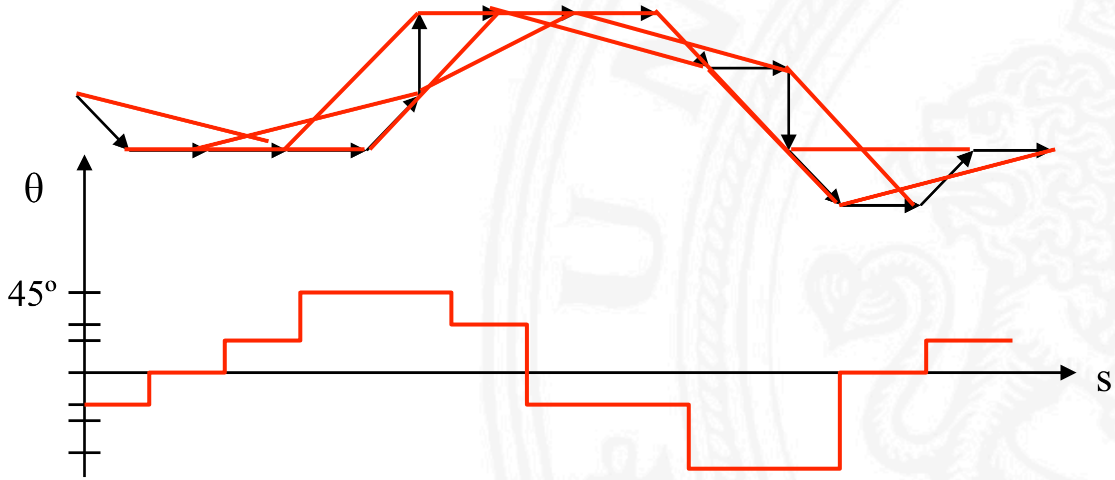
Smoothed chain code slope and curvature:

L	chain code
$\{p_1 \dots p_N\}$	starting points of chain code elements

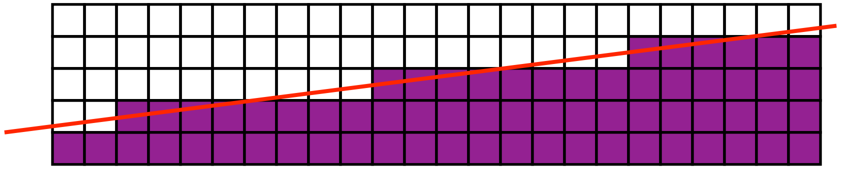
- <u>right k-slope</u> of L at i, $k \geq 1$, is slope from p_i to p_{i+k}
- <u>left k-slope</u> of L at i, $k \geq 1$, is slope from p_i to p_{i+k}
- <u>k-curvature</u> at i is difference between right and left k-slope

<u>Example</u>:

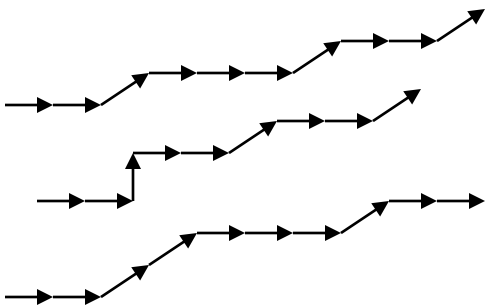
$k = 3$



Digital Straight Lines



What are the properties of a chain code which represents a straight line boundary?



may represent a straight line

may not represent a straight line

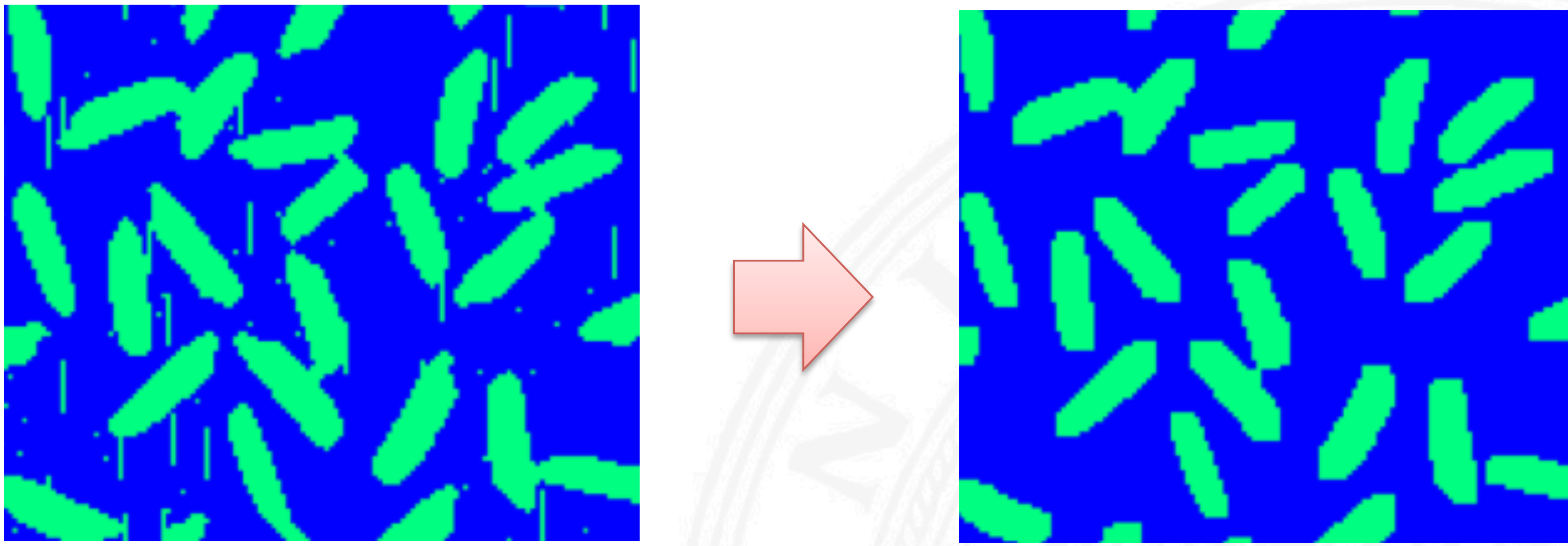
may not represent a straight line

Necessary and sufficient straight line properties of chain code:

1. **Only 2 element types**
2. **Numerical difference of element types (mod 8) at most 1**
3. **One of the element types occurs only in runs of length 1 and is distributed "as regularly as possible".**

"as regularly as possible": Assume 2 types a and b, b single. Runs of a must have lengths l_0 and l_0+1. Consider l_0-runs and l_0+1-runs as 2 chain code types and apply straight line criteria recursively.

Removal of Small Disturbances



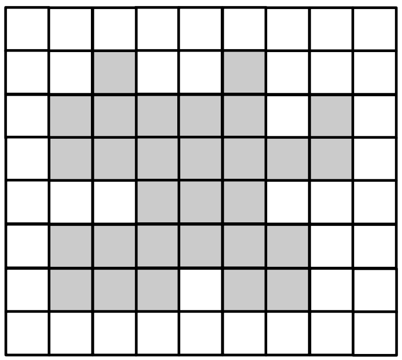
- **Salt-and-pepper noise creates false objects or holes**
- **Objects are merged because of touch or noise**

Morphological Operations

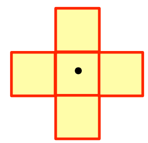
- „morphology“ ≈ laws of structure, relevant for many disciplines
- Qualitative characterizations of morphologigal operations on images:
  - Erosion removes boundary strip from regions
  - Dilation expands regions by boundary strip
  - Opening erosion followed by dilation, removes small protrusions from a region
  - Closing dilation followed by erosion, removes small intrusions from a region
- Morphological operations in Image Processing are defined by logical operations on binary images:
  - black = 0 = F
  - white = 1 = T

Erosion and Dilation

image G



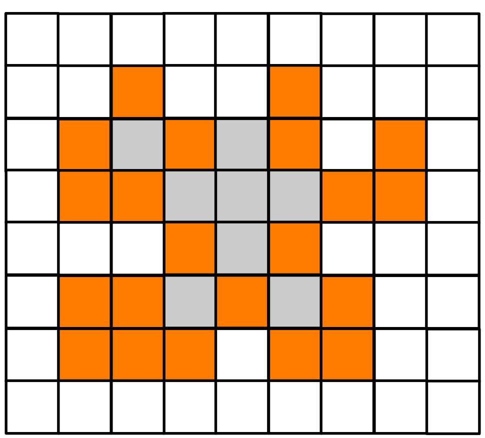
example of structuring element H



Local neighborhood operation with $g' = f(G_H, H)$ where G_H are pixels of G covered by H.

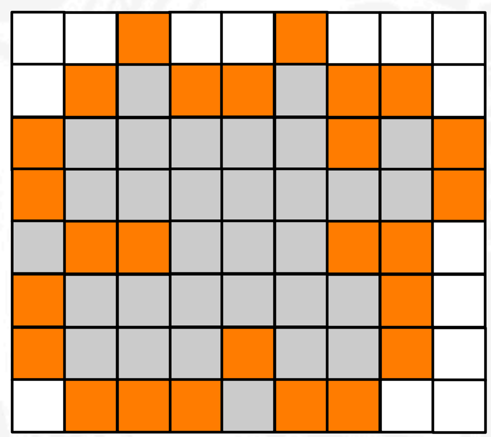
Erosion: $g' = AND(G_H, H)$

Reference location is 1 if all pixels covered by H are 1.



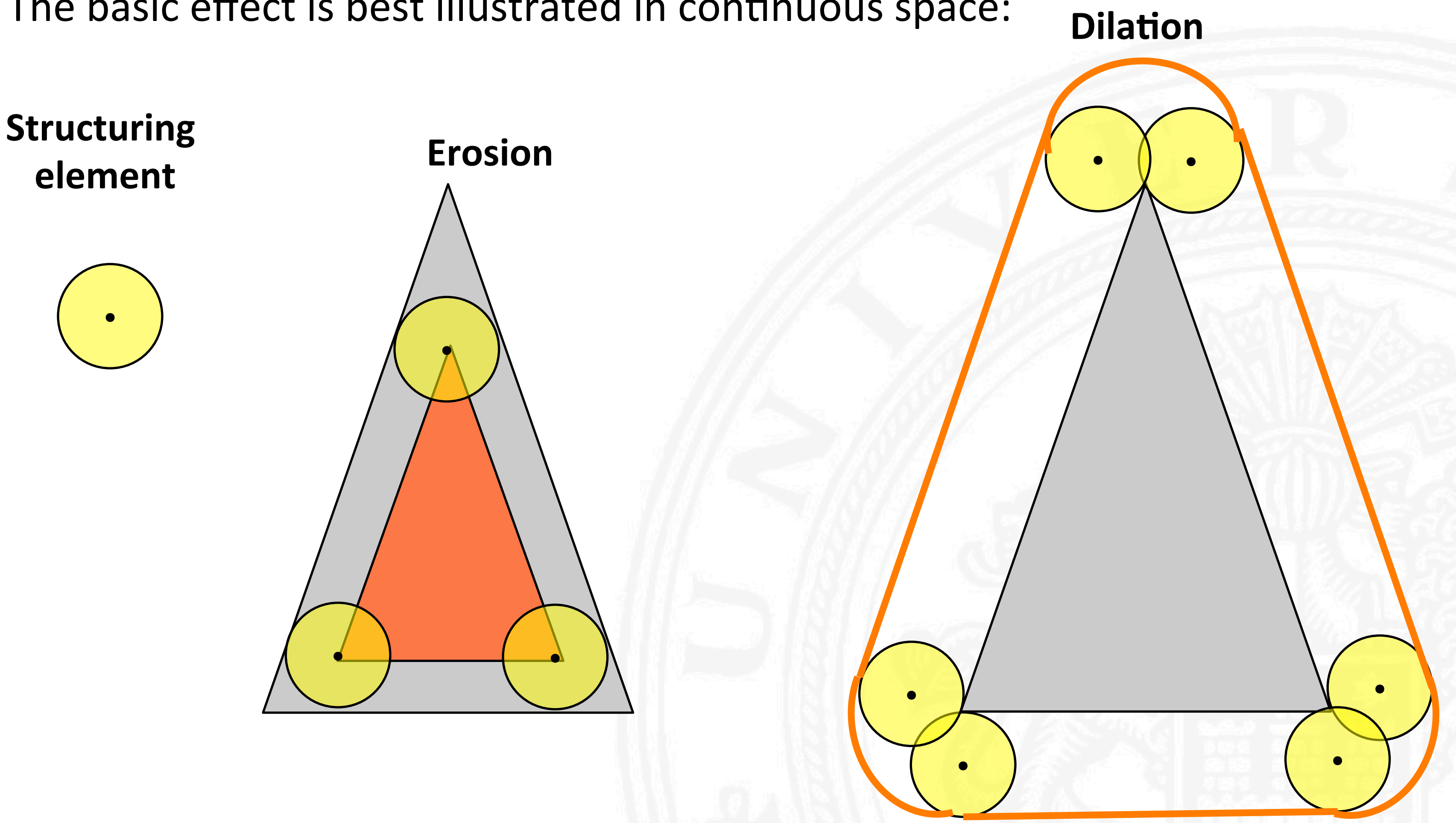
Dilation: $g' = OR(G_H, H)$

Reference location is 1 if at least one pixel covered by H is 1.



Erosion and Dilation Examples

The basic effect is best illustrated in continuous space:



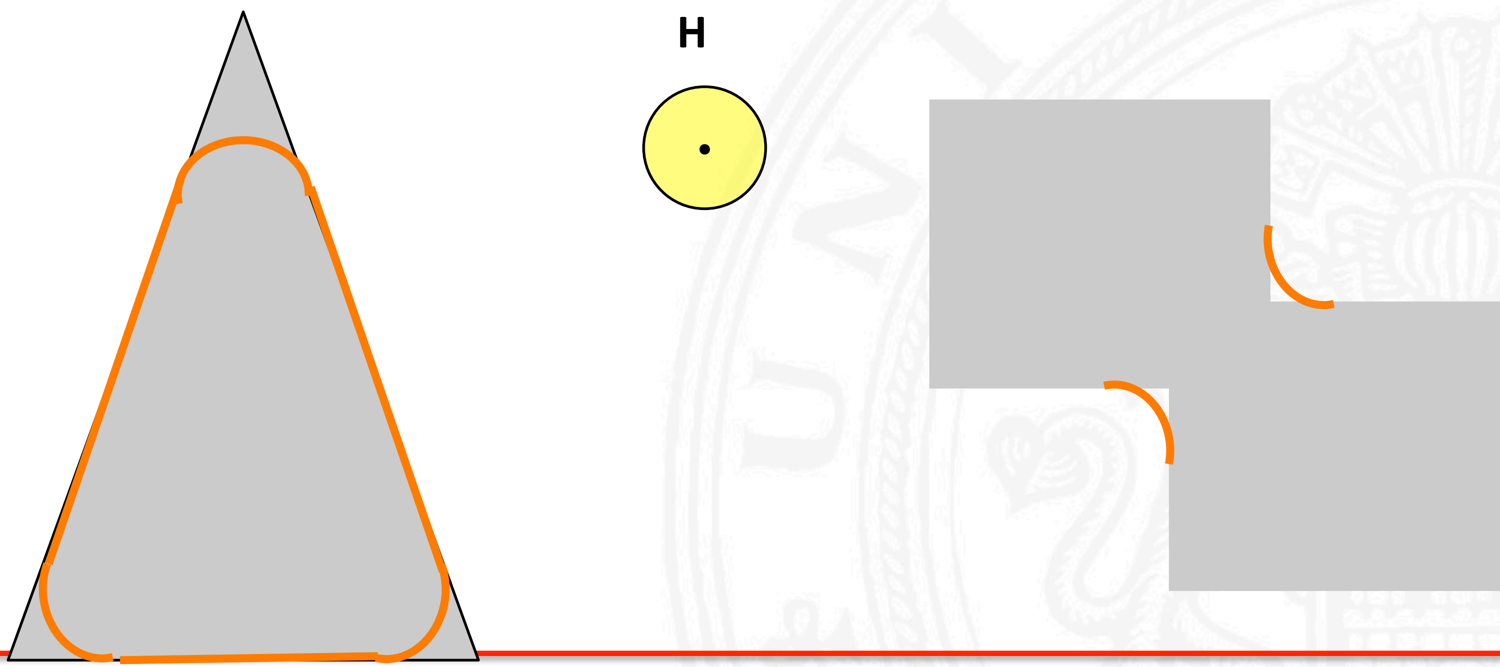
Opening and Closing

Opening is erosion of G by H, resulting in G', followed by dilation of G' by H.

Opening removes small protruding structures.

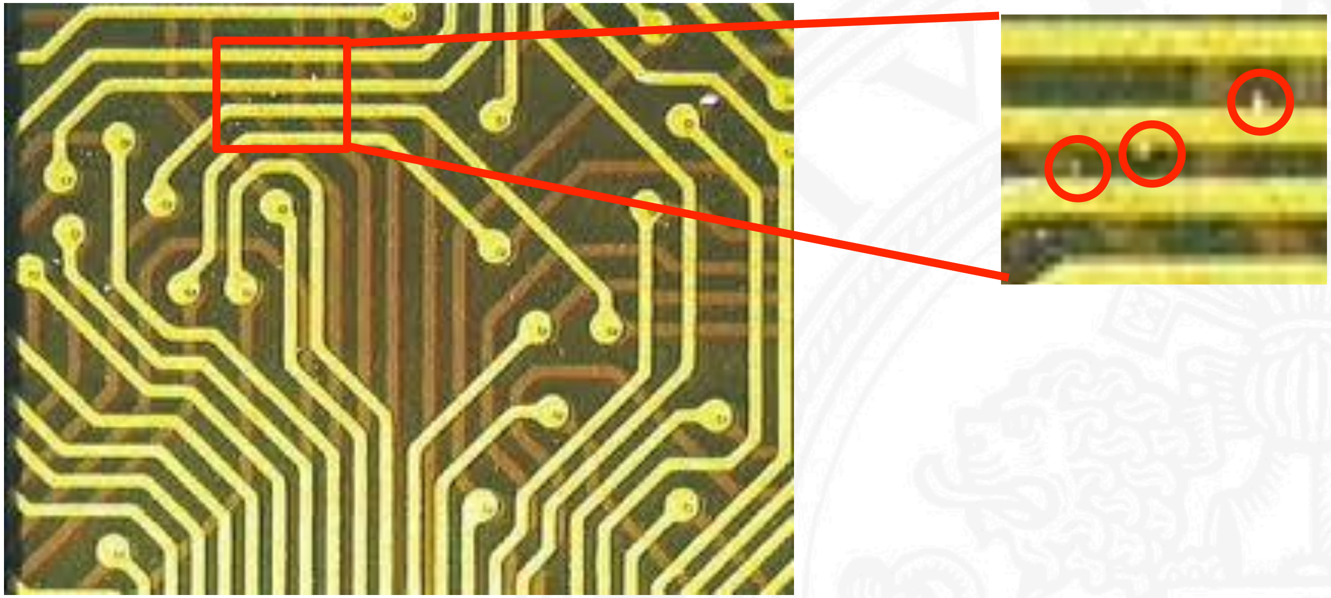
Closing is dilation of G by H, resulting in G', followed by erosion of G' by H.

Closing adds small protruding structures.



Example for Morphological Operations

- Quality control of printed circuits:
- Find small unwanted protrusions



- What morphological operations can locate protrusions?

Erosion and Dilation for Greyvalue Images

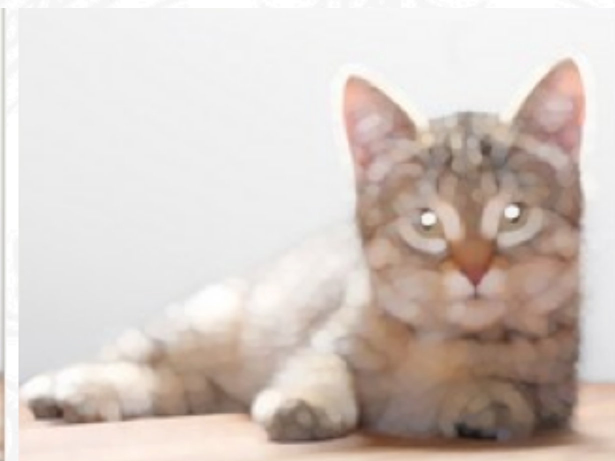
Erosion: $g' = \min(G_H, H)$ New value g' at reference location is minimum of all greyvalues of G covered by H.

Dilation: $g' = \max(G_H, H)$ New value g' at reference location is maximum of all greyvalues of G covered by H.

original

eroded

dilated



Note that morphological operations are often defined similar to convolution:
Move mirror image of H across G (no difference if H is symmetrical).