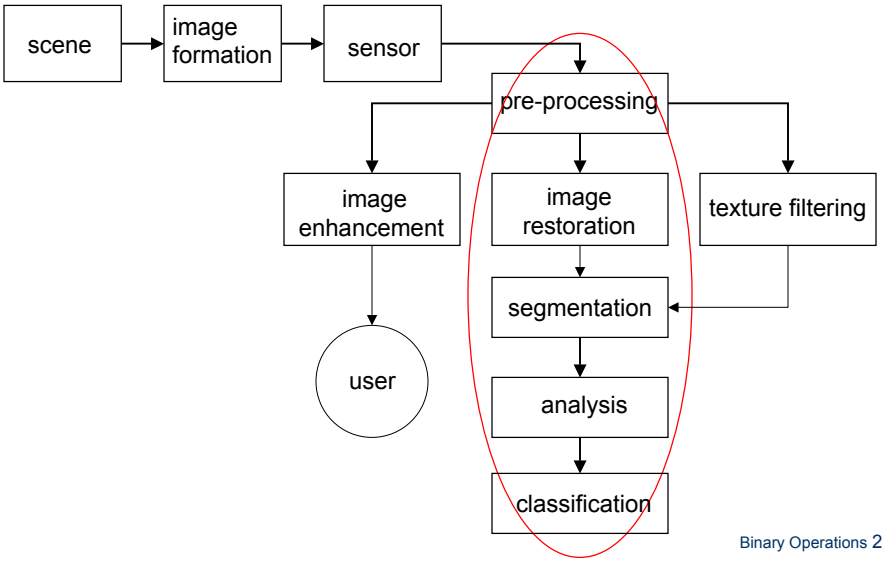
Binary Image Processing

Lucas J van Vliet
www.ph.tn.tudelft.nl/~lucas

TU Delft
TNW: Faculty of Applied Sciences
IST: Imaging Science and technology
PH: Pattern Recognition Group

Image Analysis Paradigm



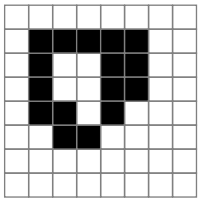
Binary Image Operations

TU Delft
Pattern Recognition Group

- Binary images have 1 bit per pixel:
  1 = object pixel
  0 = non-object pixel (background pixel)

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

Data



Image

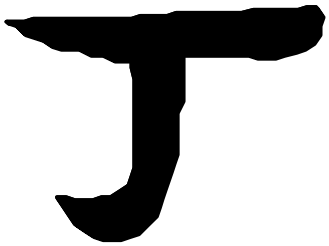
Binary Image Operations

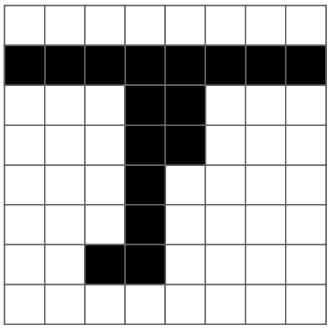
TU Delft
Pattern Recognition Group

- Binary images are obtained from segmentation procedures such as:
  - Thresholding (binarization at grey-level *t*)
  - Edge detection and filling
  - Region analysis
- Binary Image Operations are used to
  - Correct or improve imperfect segmentation
  - Analysis of connectivity of components
  - Object selection using geometric features

Objects and Components

TU Delft
Pattern Recognition Group





Object =
real world thing

Component =
blob of connected pixels

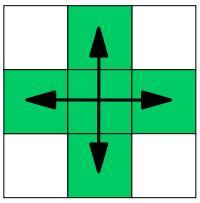
Correspondence by segmentation and binary image processing

The paradox of the square grid

TU Delft
Pattern Recognition Group

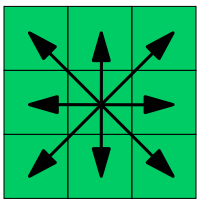
Two basic neighborhoods (connectivities) on a square grid, denoted by N_4 and N_8, symmetric around the origin.

N_4

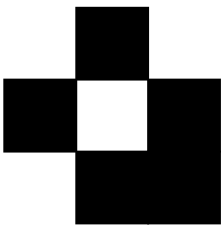


4-connectivity

N_8



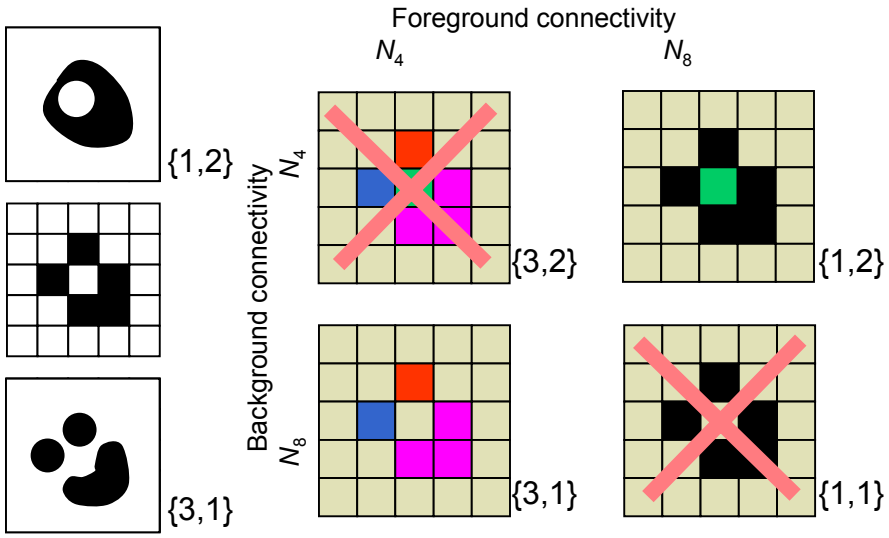
8-connectivity



How many objects?
And how many background components?

The paradox of the square grid

TU Delft
Pattern Recognition Group



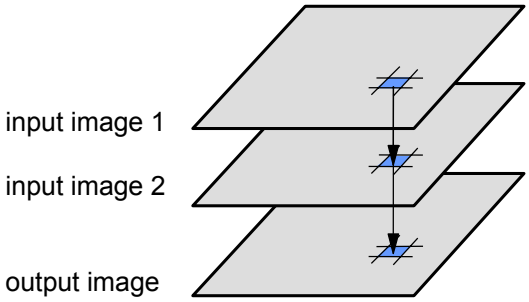
Types of operations

TU Delft
Pattern Recognition Group

Operation type	Monadic (one input image)	Dyadic (two input images)
Pixel operations	Identity, Inversion	Boolean arithmetic
Neighborhood operations	Cellular logic operations, Binary morphology	Hit-or-miss transforms
Object operations	Skeleton, Exo-skeleton	Propagation, Anchor skeleton

Pixel operations

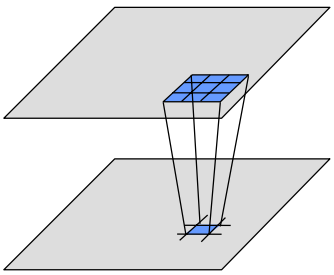
TU Delft
Pattern Recognition Group

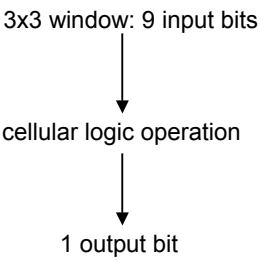


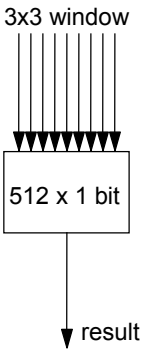
	Boolean operations	Set operations
Monadic (1 input image)	• Identity • Invert	• = • Complement: **C**
Dyadic (2 input images)	• AND • OR • XOR	• Intersection • Union

Neighborhood operations

TU Delft
Pattern Recognition Group







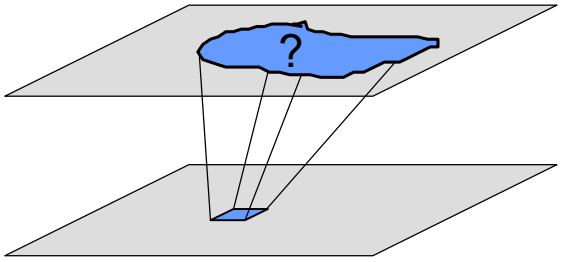
Each operation is equivalent to a table with $2^9 = 512$ entries

Examples: Erosion, Dilation, Contour extraction, Remove isolated pixels

Object oriented operations

TU Delft
Pattern Recognition Group

It is impossible to state beforehand which pixels from input image contribute to output: No fixed neighborhood size.



Examples:
- skeleton
- exo-skeleton
- propagation
- anchor-skeleton

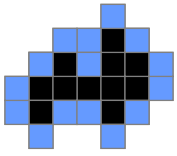
Dilation (expansion): δ

TU Delft
Pattern Recognition Group

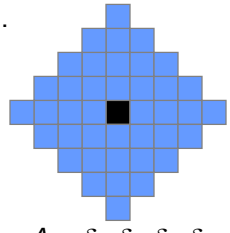
Expands each object pixel with its N_4 (N_8) neighbors.

4-connected dilation

δ_{N_4}



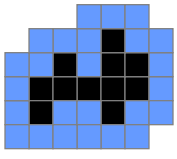
$\delta_{N_4} A$



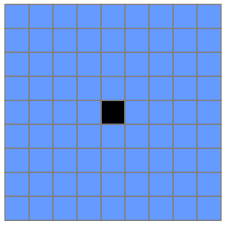
$\delta_{4N_4} A = \delta_{N_4}\delta_{N_4}\delta_{N_4}\delta_{N_4} A$

8-connected dilation

δ_{N_8}



$\delta_{N_8} A$



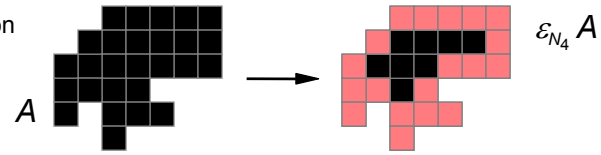
$\delta_{4N_8} A = \delta_{N_8}\delta_{N_8}\delta_{N_8}\delta_{N_8} A$

Erosion (shrinking): ε

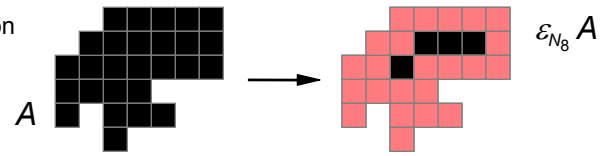
TU Delft
Pattern Recognition Group

Removes each object pixel having a neighbor in the background.
Yields all pixels whose N_4 (N_8) neighbors are all object pixels.

4-connected erosion



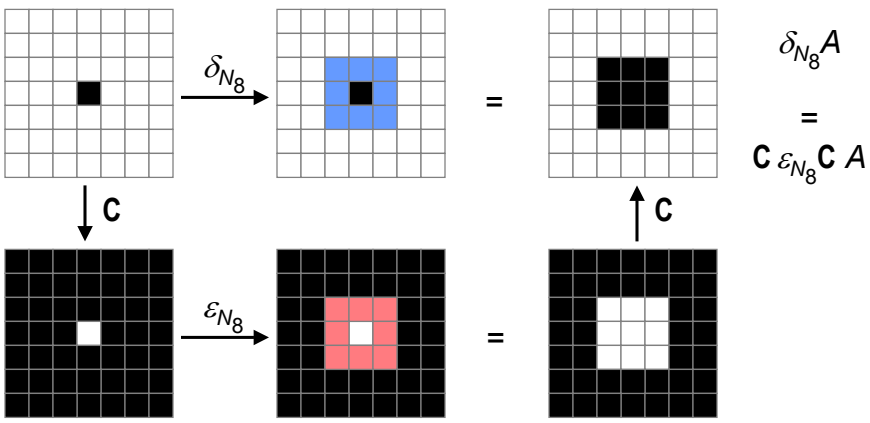
8-connected erosion



Duality: erosion versus dilation

TU Delft
Pattern Recognition Group

An 8-connected dilation of the object(s) corresponds to
an 8-connected erosion of the background, and vice-versa.



$$\delta_{N_8} A = \mathbf{C}\, \varepsilon_{N_8} \mathbf{C}\, A$$

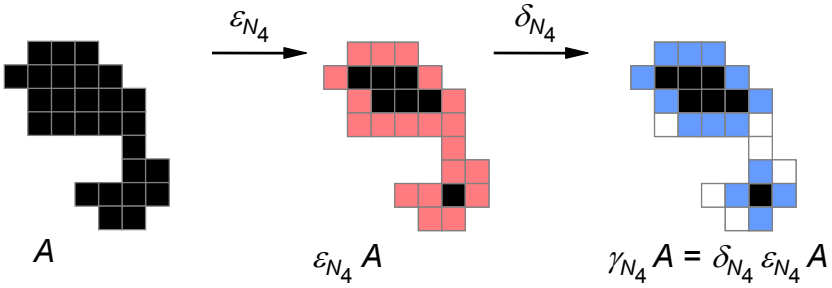
The same holds for the 4-connected case.

Opening

TU Delft
Pattern Recognition Group

An opening consists of a number of erosions followed by the same number of dilations.
N denotes: n x N_4 and m x N_8, with $n,m \in \mathbb{N}$

$$\gamma_N A = \delta_N \varepsilon_N A$$



Applications:

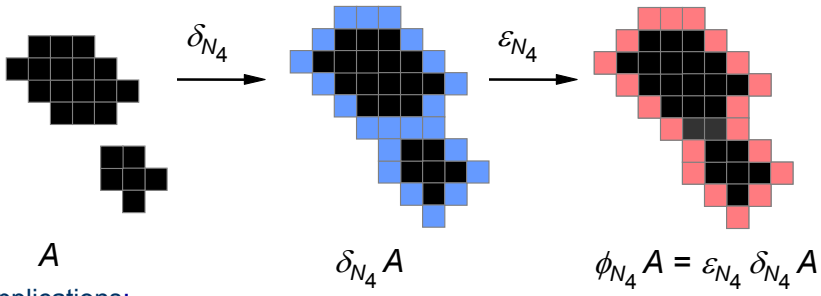
- separation of connected objects
- removing small objects (size discrimination)

Closing

TU Delft
Pattern Recognition Group

A closing consists of a number of dilations followed by the same number of erosions.
N denotes: n x N_4 and m x N_8, with $n,m \in \mathbb{N}$

$$\phi_N A = \varepsilon_N \delta_N A$$

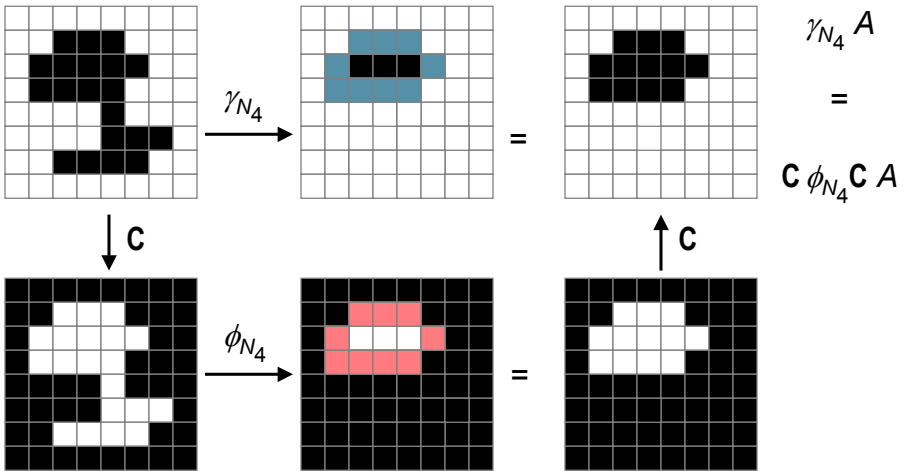


Applications:

- connection of disconnected object parts
- closing of holes
- contour smoothing

Duality: opening versus closing

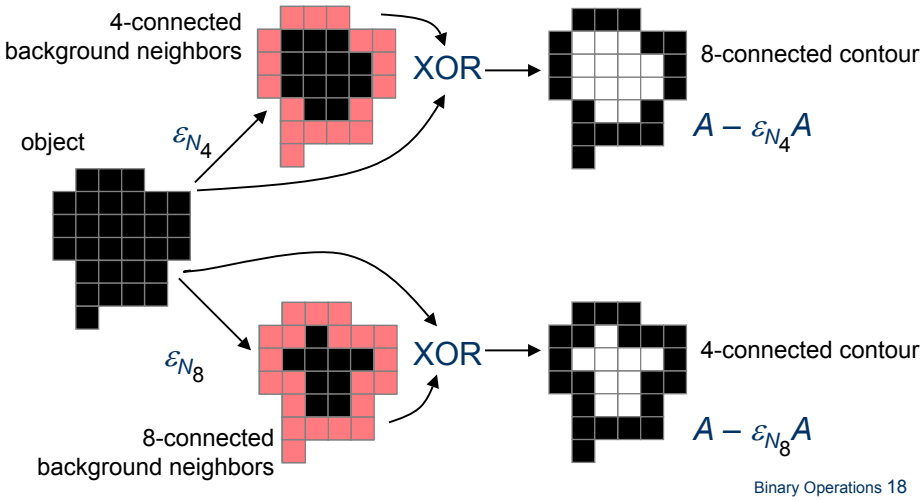
TU Delft
Pattern Recognition Group



Contour finding

TU Delft
Pattern Recognition Group

The contour is the set of object pixels that are connected to the background



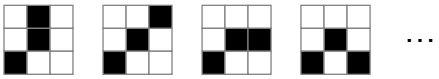
Conditional erosion: skeleton

TU Delft
Pattern Recognition Group

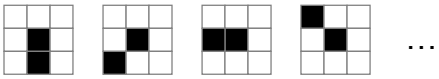
A conditional erosion is an erosion under one or more of the following conditions:

A. Do not remove a single pixel

B. Do not break the connectivity



C. Do not remove an end-pixel

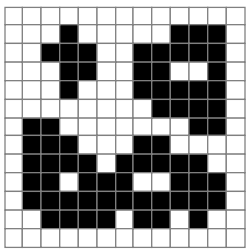


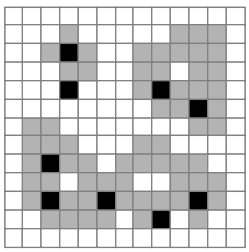
{A}: Object to point(s) reduction
{B}: Find holes
{A, B}: Skeleton without end-pixel condition
{A, B, C}: Skeleton with end-pixel condition

Conditional erosion *(cont.)*

TU Delft
Pattern Recognition Group

Do not remove a single pixel...





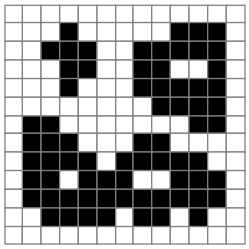
…object to point reduction

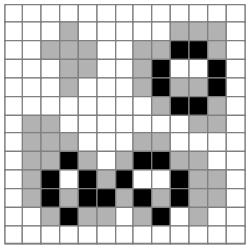
Note: not always one point per object !

Conditional erosion *(cont.)*

TU Delft
Pattern Recognition Group

Do not break connectivity...



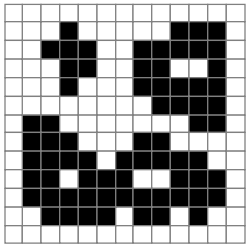


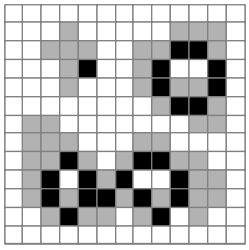
…find holes

Conditional erosion *(cont)*

TU Delft
Pattern Recognition Group

Do not remove a single pixel &
do not break connectivity...



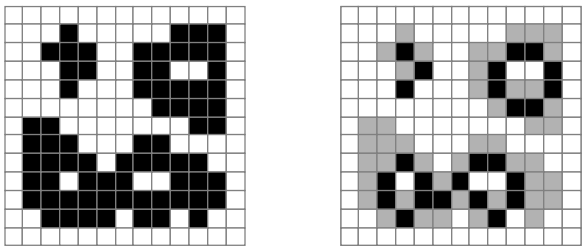


…skeleton without end-pixel condition

Conditional erosion *(cont.)*

TU Delft
Pattern Recognition Group

Do not remove a single pixel &
Do not break connectivity &
Do not remove an end-pixel...



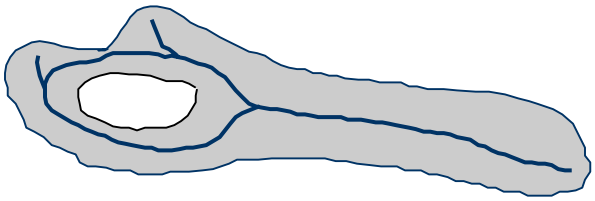
…skeleton with end-pixel condition

Skeleton: definition

TU Delft
Pattern Recognition Group

The skeleton of an object is a line
connecting points midway between the boundaries

A skeleton has the same connectedness
("topology") as the original object.



Digital skeleton

TU Delft
Pattern Recognition Group

- Wishes:

  The skeleton...

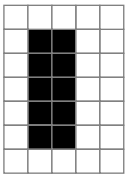
  1. has the same connectedness as the original object,
     so it can be used as an representation of the object.
  2. is one pixel thick,
     so crossings and end points can be found in a 3x3 neighborhood.
  3. preserves isolated pixels and endpoints,
     so long objects have long skeletons.
  4. is in the middle of the object,
     so it can be used as an object 'axis'.

Digital skeleton *(continued)*

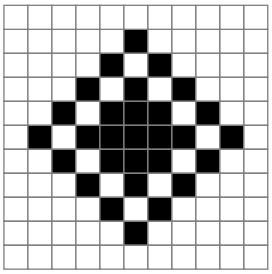
TU Delft
Pattern Recognition Group

- Wishes may conflict

  Conflict between:
  - 'one pixel thick' and
  - 'in the middle of the object'



  Conflict between:
  - 'same connectedness' and
  - 'one pixel thick'



Digital skeleton *(continued)*

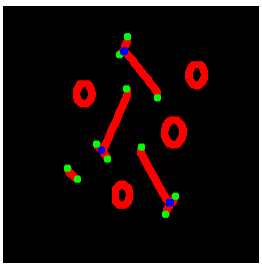
TU Delft
Pattern Recognition Group

- Many different skeleton algorithms exist, and also a variety of versions of the same algorithm
- They differ in:
  - handling conflicting wishes
  - noise sensitivity
  - 'cleanness' of the result
  - sensitivity to details of the object contours
- Well-known algorithms:
  - the Hilditch algorithm for sequential machines and
  - the algorithm by Arcelli and Levialdi for parallel machines

Special filters: skeleton points

TU Delft
Pattern Recognition Group

- A skeleton may consist of
  - End-pixels
  - Link-pixels
  - Branch-pixels



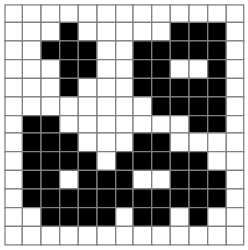
- Identification of special points may lead to object classification:
  - Number of end-pixels
  - Number of holes (topology, Euler number)
  - Number of link-pixels (size)

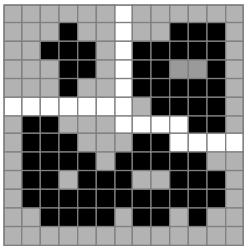
Conditional dilation: exo-skeleton

TU Delft

Pattern Recognition Group

The exo-skeleton is a dilation under the condition:
'Do not connect disconnected components'



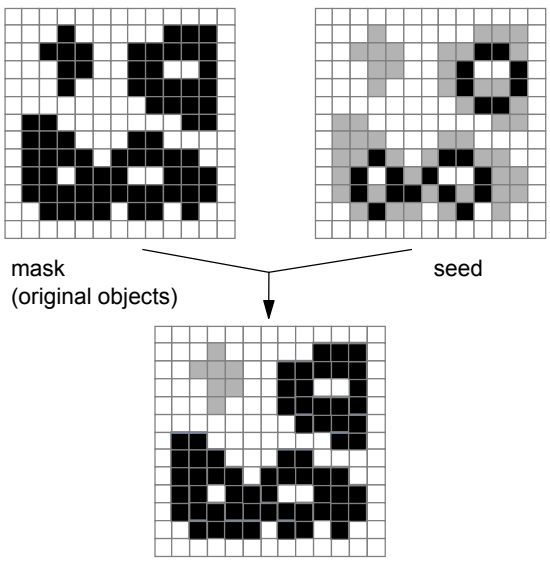


Similar result by inverting the original image and skeletonize it
However: Most skeleton programs can only give an 8-connected result, i.e., areas remain 8-connected

Propagation

TU Delft

Pattern Recognition Group



Propagation is a dyadic conditional dilation:
Dilate a (seed) image recursively within a given (object) mask.

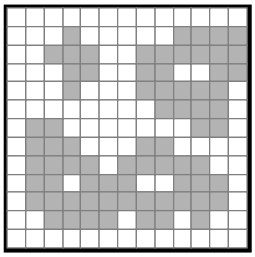
$$S_{i+1} = \delta_{N_8} S_i \cap M$$

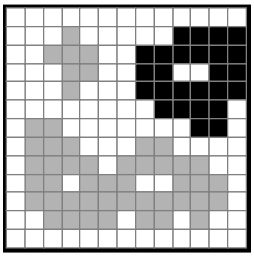
S_0 = seed image
M = mask image

Propagation: applications

TU Delft
Pattern Recognition Group

Find objects connected to the image boundary





$S_{i+1} = \delta_{N_8} S_i \cap M$

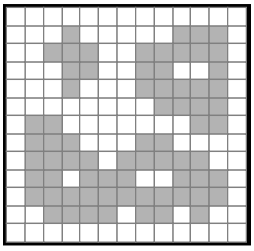
S_0 = image border (as seed)
M = objects (as mask)

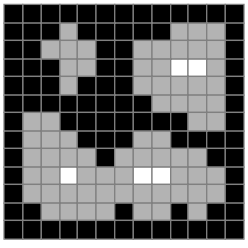
Result: object(s) connected to the image boundary

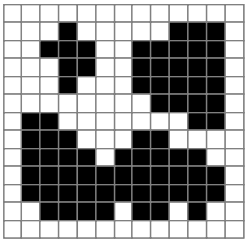
Propagation: applications

TU Delft
Pattern Recognition Group

Close holes







Seed: image boundary
Mask: background

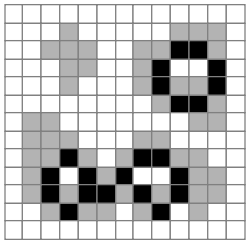
Result:
background connected to the image boundary

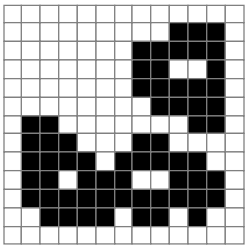
Inverse of propagation result:
objects with holes closed

Propagation: applications

TU Delft
Pattern Recognition Group

Find specific objects





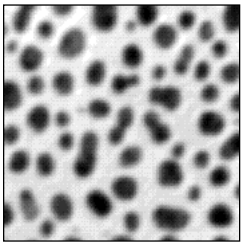
Seed: pixels of selected objects
Mask: all objects

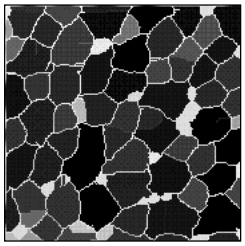
Result: selected objects

Watershed

TU Delft
Pattern Recognition Group

Characterise grey-scale topology



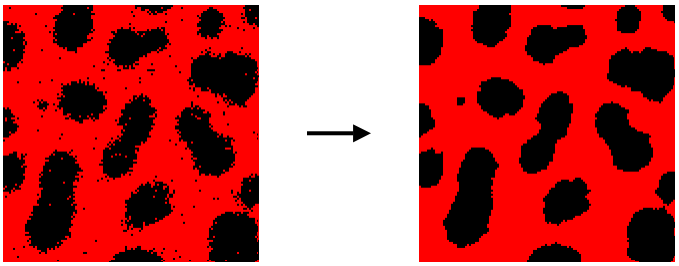


Note:
- very noise sensitive
- pre-processing recommended
- splitting and merging nearly always necessary

Special filters: Majority Vote

TU Delft
Pattern Recognition Group

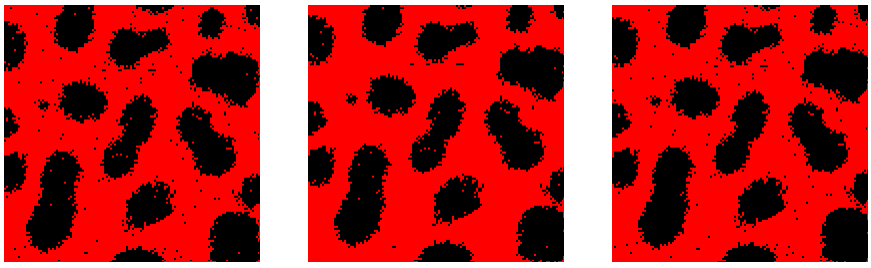
- Majority vote is the binary counterpart of a median filter.
- Replace the current pixel by the majority of its 8-connected neighbors
- Majority voting is binary “smoothing” filter:
  - Contours become smoother
  - Small objects or holes are removed. How small?



Special filters: pepper & salt

TU Delft
Pattern Recognition Group

- Pepper and salt removal refers to the suppression of single pixels (either ‘one’ or ‘zero’).
- Removal of isolated object or background pixels



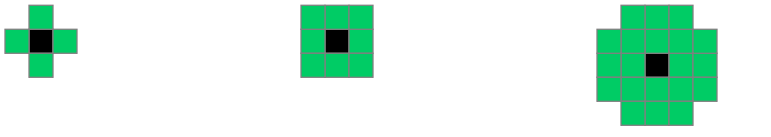
Input A | Pepper removed | Salt removed

Dilation revisited

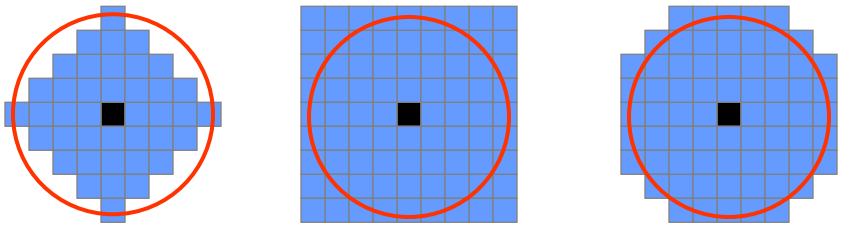
TU Delft
Pattern Recognition Group

Expands each object pixel with its neighbors.

4-connected dilation | 8-connected dilation | $2N_{4\text{-}8}$ connected dilation



δ_{4N_4} | δ_{4N_8} | $\delta_{4N_{4\text{-}8}}$



Distance transforms (1)

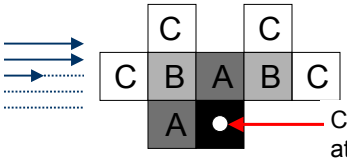
TU Delft
Pattern Recognition Group

Assign to each object pixel the distance to the nearest background pixel.

Algorithm:

1. Initialize: dt(**x**) = 0 iff **x** ∈ background), else infinity
2. Repeat step 3 and 4 until stabilization
3. Forward scan (for all **x**)
   $dt(\mathbf{x}) = \min\{dt(\mathbf{x}-\mathbf{a}_{1,2}) + A,\ dt(\mathbf{x}-\mathbf{b}_{1,2}) + B,\ dt(\mathbf{x}-\mathbf{c}_{1,2,3,4}) + C\}$
4. Backward scan (for all **x**)
   $dt(\mathbf{x}) = \min\{dt(\mathbf{x}+\mathbf{a}_{1,2}) + A,\ dt(\mathbf{x}+\mathbf{b}_{1,2}) + B,\ dt(\mathbf{x}+\mathbf{c}_{1,2,3,4}) + C\}$

Forward scan

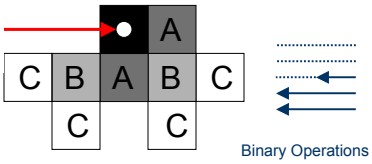


$-\mathbf{a}_{1,2}$ address neighbors A
($-\mathbf{b}_i \rightarrow$ B and $-\mathbf{c}_i \rightarrow$ C)
(A,B,C) are distances: A ≤ B ≤ C

Current pixel at position **x**

Backward scan

$+\mathbf{a}_{1,2}$ address neighbors A (etc)
($+\mathbf{b}_i \rightarrow$ B and $+\mathbf{c}_i \rightarrow$ C)
(A,B,C) are distances: A ≤ B ≤ C



Distance transforms (2)

TU Delft
Pattern Recognition Group

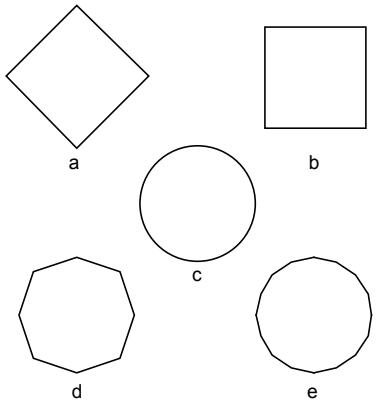
Each set of neighbor distances (A,B,C) yields a metric.

Which metric approximates the Euclidean metric?

"Circles" for different metrics:

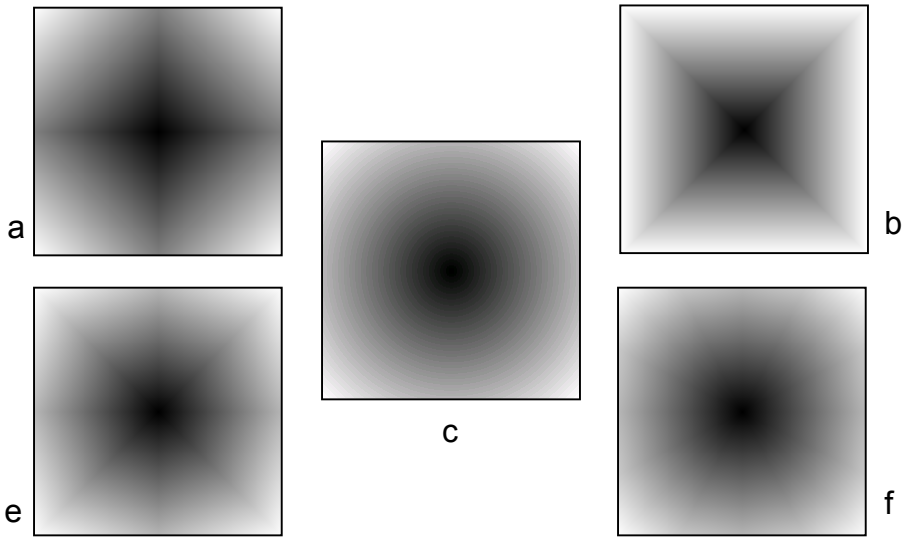
Neighbor distances (A,B,C)

a) City block (4-conn.) (1,2,1+2=3)
b) Chess board (8-conn.) (1,1,1+1=2)
c) Euclidean
d) Octagonal (5,7,5+7=12)
e) Hexadecagonal (5,7,11)



Distance transforms (3)

TU Delft
Pattern Recognition Group

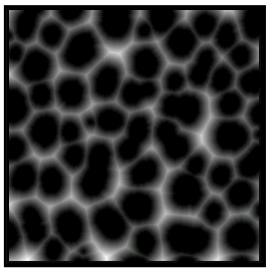


Isotropic morphology

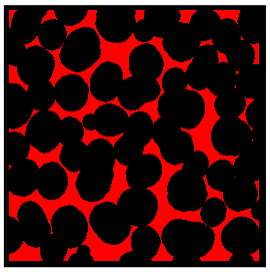
TU Delft
Pattern Recognition Group

- Isotropic morphology requires an isotropic structuring element.
- Thresholding the Euclidean distance transform at value ***t*** is identical to an erosion with a disc of radius ***t***.

Input image A



Distance transform
d = dt(A)



ε_{7R} A
= threshold(d, t = 7)

- The ridges in the 'distance image' yield the skeleton.

Threshold transforms (1)

TU Delft
Pattern Recognition Group

- Morphological filtering of image ***A*** can be realized by a uniform filter with neighborhood (=kernel) ***N***, followed by thresholding at level ***t***.

$$\mathrm{M}_N(A;t) = \text{threshold}(A * N;t) = \begin{cases} 1 & \text{if } A * N \geq t \\ 0 & \text{if } A * N < t \end{cases}$$

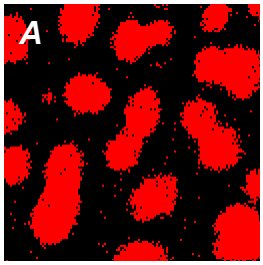
with $A * N$ is convolution of image A by filter N

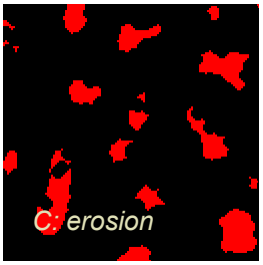
- Kernel size: #N = sum(B)
- Dilation: $\delta_N A$ = $\mathrm{M}_N(A;$ ***t*** = 1) (Sensitive to a single object pixel)
- Erosion: $\varepsilon_N A$ = $\mathrm{M}_N(A;$ ***t*** = #N) (All '#N' pixels should fit in the object)

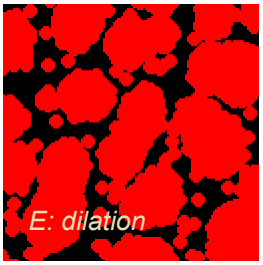
- Fuzzy dilation can be obtained by: $\mathrm{M}_N(A;$ ***t*** ≈ 0.1 #N)
- Fuzzy erosion can be obtained by: $\mathrm{M}_N(A;$ ***t*** ≈ 0.9 #N)
  - Less sensitive to noise pixels

Threshold transforms (2)

TU Delft
Pattern Recognition Group







A = input image
N = disc of *radius* 7
sum(*B*) = 37

$C = M_N(A; t = 37)$
$D = M_N(A; t = 34)$
$E = M_N(A; t = 1)$
$F = M_N(A; t = 4)$