INF 4300 – Digital Image Analysis

MORPHOLOGICAL IMAGE PROCESSING

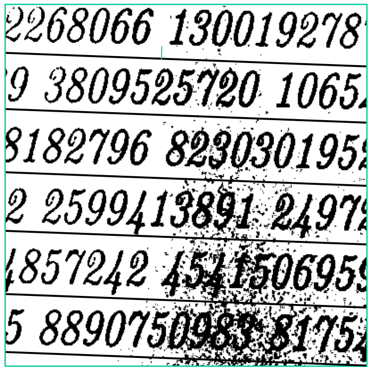
Fritz Albregtsen 04.11.2015

Today

- Gonzalez and Woods, Chapter 9
  - Except sections 9.5.7 (skeletons), 9.5.8 (pruning), 9.5.9 (reconstruction) and 9.6.4 (gray scale reconstruction).
- Repetition:
  - binary dilatation, erosion, opening, closing
- Binary region processing:
  - connected components, convex hull, thinning/thickening.
- Grey-level morphology:
  - erosion, dilation, opening, closing,
  - smoothing, gradient, top-hat, bottom-hat, granulometry.

Example

- Text segmentation and recognition.
- Binary morphological operations are useful after segmentation in order to improve segmentation of the objects.



Some symbols have been fragmented.

Some symbols are connected with background noise.

Symbols can be connected with neighboring symbols.

Need to remove lines or frames.

Simple set theory – read yourself

- Let A be a set in $\mathbb{Z}^2$ (integers in 2D). If the point $a=(a_1,a_2)$ is an element in A we denote: $a \in A$
- If a is not an element in A we denote: $a \notin A$
- An empty set is denoted $\varnothing$.
- If all elements in A are also part of B, A is called a subset of B and denoted: $A \subseteq B$
- The union of two sets A and B consists of all elements in either A or B, and is denoted: $A \cup B$
- The intersection (="snitt") of A and B consists of all elements that are part of both A and B and is denoted: $A \cap B$
- The complement of a set A is the set of elements not in A:

$$A^C = \{w \mid w \notin A\}$$

- The difference of two sets A and B is:

$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^C$$

Set theory on binary images

- The complement of a binary image

$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)=0\\ 0 & \text{if } f(x,y)=1\end{cases}$$

- The intersection of two images f and g is

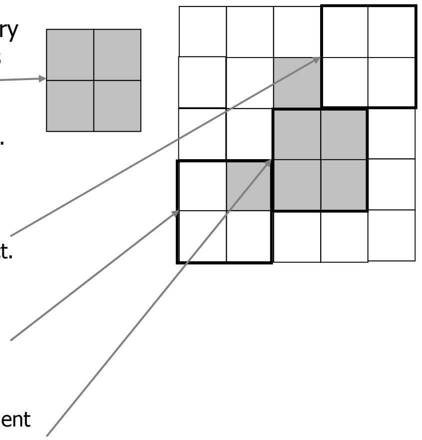
$$h=f\cap g=h(x,y)=\begin{cases}1 & \text{if } f(x,y)=1 \text{ and } g(x,y)=1\\ 0 & \text{otherwise}\end{cases}$$

- The union of two images f and g is

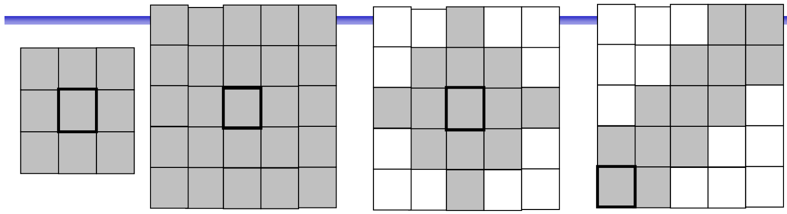
$$h=f\cup g=h(x,y)=\begin{cases}1 & \text{if } f(x,y)=1 \text{ or } g(x,y)=1\\ 0 & \text{otherwise}\end{cases}$$

Repetition - Hit vs. Fit

- A structuring element for a binary image is a small matrix of pixels
- Let the structuring element overlay the binary image containing an object at different pixel positions.
- The following cases arise:
  - Positions where the element does not overlap with the object.
  - Positions where the element partly overlaps the object
    - the element hits the object
  - Positions where the whole element fits inside the object
    - the element fits the object



Repetition – structuring elements



- structuring elements can have different sizes and shapes
- A structuring element has an origin/center
  - The origin is a pixel
  - The origin *can* be outside the element.
  - The origin is often marked on the structuring element using □
  - Otherwise, we assume the center pixel is the origin.
- The structuring element can be flat or non-flat (have different values)
  - We will only work with a flat structuring element

Repetition
Erosion of a binary image

- To compute the erosion of pixel (x,y) in image f with the structuring element S: place the structuring element such that its origo is at (x,y). Compute

$$g(x,y)=\begin{cases}1 & \text{if S fits f}\\ 0 & \text{otherwise}\end{cases}$$

- Erosion of the image f with structuring element S is denoted

ε (f|S) = f ⊖ S

- Erosion of a set A with the stucturing element B is defined as the position of all pixels x in A such that B is included in A when origo of B is at x.

$$A\ominus B=\{x \mid B_x\subseteq A\}$$

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

eroded by

1	1	1
1	$\boxed{1}$	1
1	1	1

gives

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

0	1	0
1	$\boxed{1}$	1
0	1	0

gives

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

Edge detection using erosion

- Erosion removes pixels on the border of an object.
- We can find the border by subtracting an eroded image from the original image: $g = f - (f \ominus s)$
- The structuring element decides if the detected edge pixels will be 4-neighbors or 8-neighbors

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

Eroded by

$$\begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$$

gives

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

difference => edge pixels

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

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

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

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

8-edgels – 4-edgels

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

Edge detection

$$f - (f \ \theta \ S)$$

Example use: find border pixels in a region

Dilation of a binary image

- Place S such that origo lies in pixel (x,y) and use the rule

$$g(x,y) = \begin{cases} 1 & \text{if } S \text{ hits } f \\ 0 & \text{otherwise} \end{cases}$$

- The image f dilated by the structuring element S is denoted:

$$f \oplus S$$

- Dilation of a set A with a structuring element B is defined as the position of all pixels x such that B overlaps with at least one pixel in A when the origin is placed at x.

$$A \oplus B = \{x \mid B_x \cap A \neq \emptyset\}$$

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

Dilated by

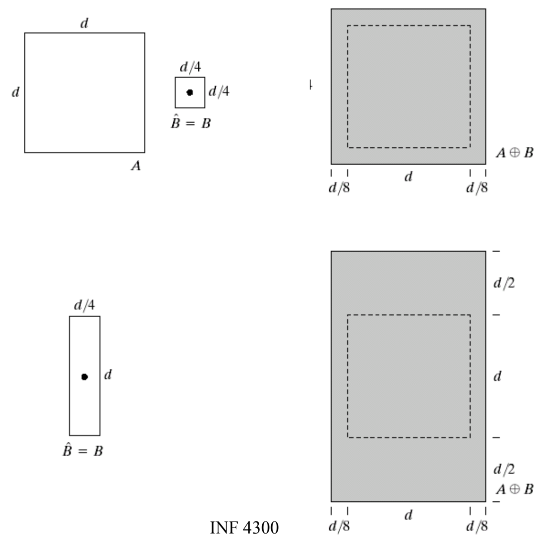
$$\begin{matrix} 1 & 1 & 1 \\ 1 & \boxed{1} & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$$

gives

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

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

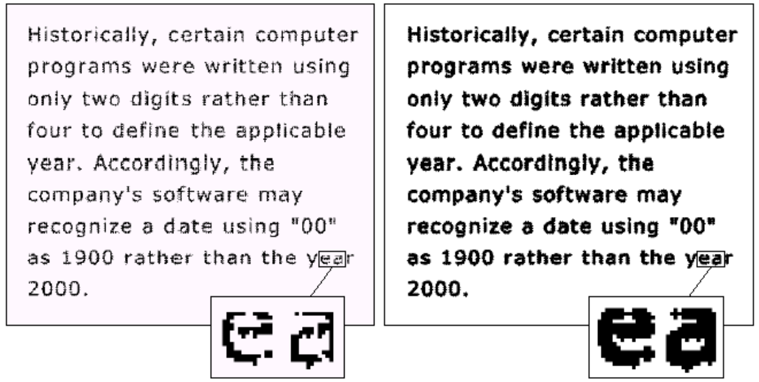
Dilation



Effect of dilation

- Expand the object borders
  - Both inside and outside borders of the object
- Dilation fills holes in the object
- Dilation smooths out the object contour
- Depends on the structuring element
- Bigger structuring element gives greater effect

Example of use of dilation – fill gaps



0	1	0
1	1	1
0	1	0

Opening

- Erosion of an image removes all structures that the structuring element cannot fit inside, and shrinks all other structures.
- Dilating the result of the erosion with the same structuring element, the structures that survived the erosion (were shrunken, not deleted) will be restored.
- This is called morphological opening:

$$f \circ S = (f \,\theta\, \mathrm{S}) \oplus S$$

- The name tells that the operation can create an opening between two structures that are connected only in a thin bridge, without shrinking the structures (as erosion would do).

Opening

- The operation can create an opening between two structures that are connected only in a thin bridge, without shrinking the structures (as erosion would do).

$$f \circ S = (f \,\theta\, \mathrm{S}) \oplus S$$

original

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

1	1	1
1	1	1
1	1	1

erosion

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

dilated erosion

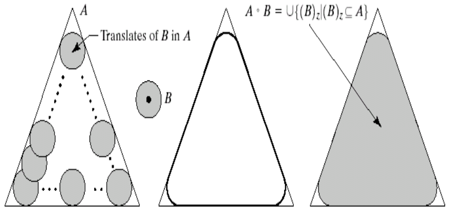
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	1	1	1	0
0	1	1	1	0	0	0	1	1	1	0
0	1	1	1	0	0	0	1	1	1	0
0	1	1	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

difference

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

Visualizing opening

- Imagine that the structuring element traverses the edge of the object.
  - First on the inside of the object. The object shrinks.
  - Then the structuring element traverses the outside of the resulting object from the previous passage.
  - The object grows, but small branches removed in the last step will not be restored.



Closing

- A dilation of an object grows the object and can fill gaps.
- If we erode the result with the rotated structuring element, the objects will keep their structure and form, but small holes filled by dilation will not appear.
- Objects merged by the dilation will not be separated again.
- Closing is defined as $f \bullet S = \left(f \oplus \hat{S}\right) \theta\, \hat{S}$
- This operation can close gaps between two structures without growing the size of the structures like dilation would.

Closing

- This operation can close gaps between two structures without growing the size of the structures like dilation would.

$$f \bullet S = \left(f \oplus \hat{S}\right) \theta\, \hat{S}$$

original

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

0	1	0
1	1	1
0	1	0

dilation

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

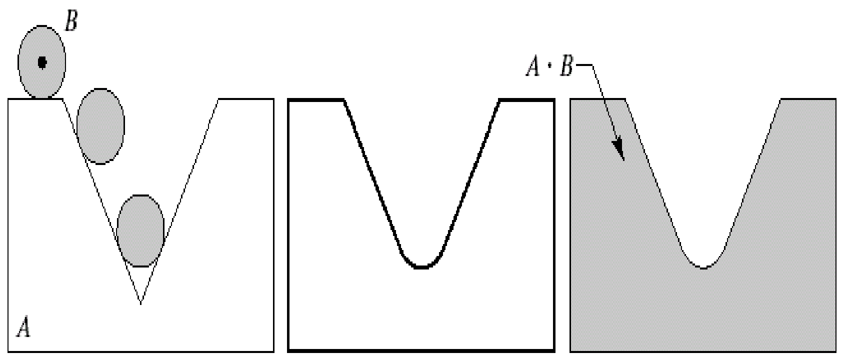
eroded dilation = closing

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

Difference from original

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

Closing

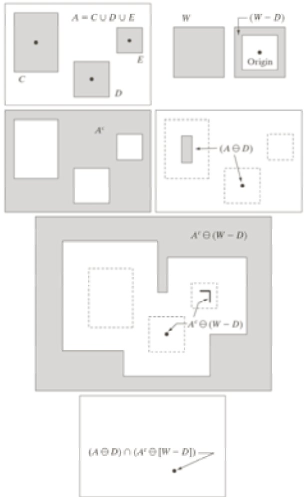


"Hit or miss"- transformation

- Transformation used to detect a given pattern in the image – "template matching"
- Objective: find location of the shape D in set A.
- D can fit inside many objects, so we need to look at the local background W-D.
- First, compute the erosion of A by D, AθD (all pixels where D can fit inside A)
- To fit also the background: Compute A^C, the complement of A. The set of locations where D exactly fits is the intersection of AθD and the erosion of A^C by W-D, A^C θ(W-D).
- Hit-or-miss is expressed as A ⊛D:

$$(A\theta D)\cap\left[A_C\theta(W-D)\right]$$

Main use: Detection of a given pattern or removal of single pixels

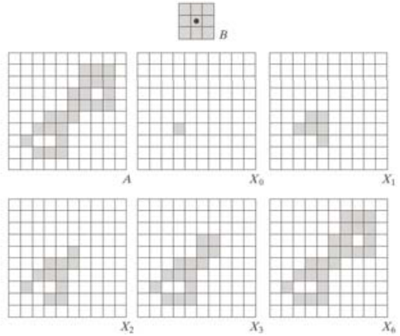


Extracting connected components

- Detects a connected object Y, in image A, given a point p in Y
  - Start with X_0, a point in Y
  - Dilate X_0 with either a square or plus
  - Let X_1 be only those pixels in the dilation that are part of the original region.
  - Continue dilating X_1 to give X_k until $X_k = X_k$-1

$$X_k = X_{k-1} \oplus B \cap A$$

$$X_0 = p,\ k=1,2,3,$$



Computing convex hull using morphology

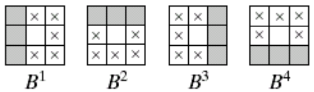
- Convex hull C of a set of points A may be estimated using the Hit-or-Miss transformation
- Consider the four structuring elements B^1-B^4.
- Apply hit-or-miss with A using B^1 iteratively until no more changes occur. Let D^1 be the result.
- Then do the same with B^2,..,B^4 to compute D^2…D^4 in the same manner.
- Then compute the convex hull by the union of all the D^is.

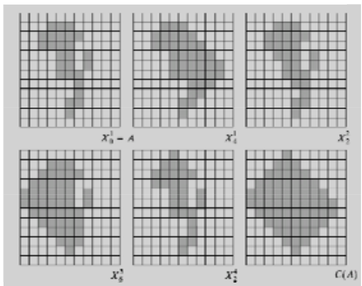
$$X_k^i = (X_{k-1}^i \circledast B^i) \cup A;\ i = 1, 2, 3, 4;$$
$$k = 1, 2, 3, \ldots;\ X_0^i = A; \text{ and}$$
$$D^i = X_{\text{conv}}^i.$$

Then C (A) = $\bigcup D_i$

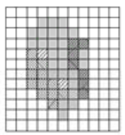
- Gives too big area to guarantee convexity
- Can be corrected by taking the intersection to the maximum dimension in x and y direction

C_res = C (A) ∩ ROI(A)



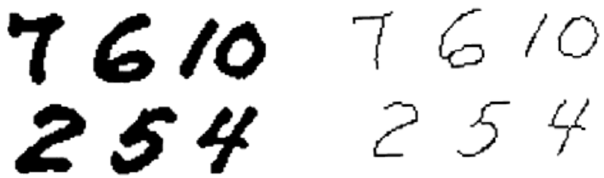


C_res



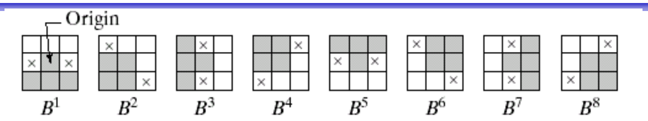
x in B means don't care

Region thinning and skeletons



- Let the region object be described using an intrinsic coordinate system, where every pixel is described by its distance from the nearest boundary pixel.
- The skeleton is defined as the set of pixels whose distance from the nearest boundary is locally maximum.
- Many different methods for computing the skeleton exist.
- Thinning is a procedure to compute the skeleton.
- Shape features can later be extracted from the skeleton.
- Skeletonization may imply loss of information.

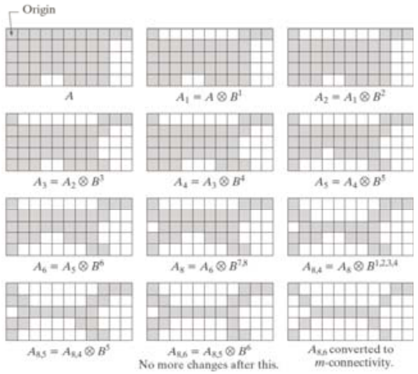
Thinning



- Thinning of set A with structuring element B (a set of n structuring elements, here n=8)
- First, thin A by one pass of B^1, then thin the result by one pass of B^2, and so on, until A is thinned with one pass of B^n.
- Then repeat the entire process until no further changes occur.

- $A \otimes B = A - (A \circledast B)$

- $A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$



Thickening

- Thickening is the dual operator of thinning.

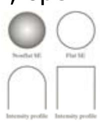
- It can be computed as a separate operation, but thickening the object is normally computed by thinning the background and then complementing the result:

  from $C=A^C$, thin C, then form C^C.

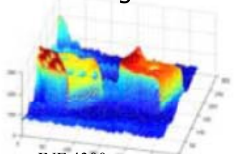
  (Example: Fig 9.22 in G&W)

Gray level morphology

- We apply a simplified definition of morphological operations on gray level images
  - Grey-level erosion, dilation, opening, closing
- Image f(x,y)
- Structuring element b(x,y)
  - May be nonflat or flat



- Assume symmetric, flat structuring element, origo at center (this is sufficient for normal use).
- Erosion and dilation then correspond to local minimum and maximum over the area defined by the structuring element



Gray level erosion /dilation

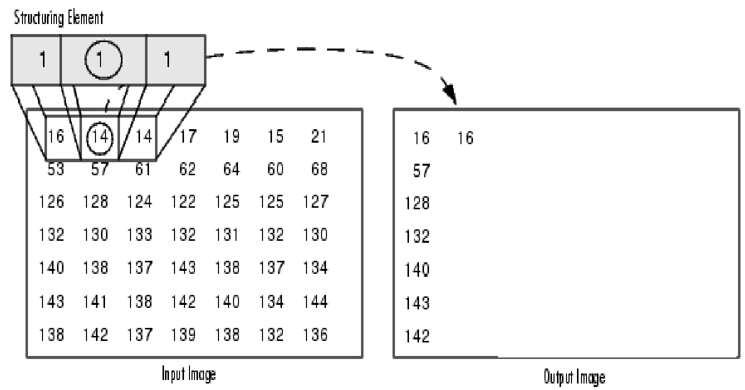
- **Erosion**:
  - Place the structuring element with origo at pixel (x,y)
  - Chose the local **minimum** gray level in the region defined by the structuring element
  - Assign this value to the output pixel (x,y)
  - Results in darker images and removal of light details:

$$[f \,\theta\, b](x,y) = \min_{(s,t)\in B}\{f(x+s, y+t)\}$$

- **Dilation**:
  - Chose the local **maximum** gray level in the region defined by the (reflected) structuring element
  - Let pixel (x,y) in the outimage have this value.
  - Gives brighter images where dark details are removed

$$[f \oplus b](x,y) = \max_{(s,t)\in B}\{f(x-s, y-t)\}$$

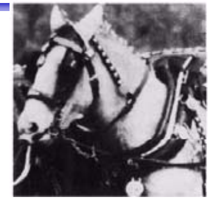
Gray level morphology- some details



Morphological Dilation of a Grayscale Image

Gray level opening and closing

- Corresponding definition as for binary opening and closing
- Result in a filter effect on the intensity
- Opening: Bright details are smoothed
- Closing: Dark details are smoothed

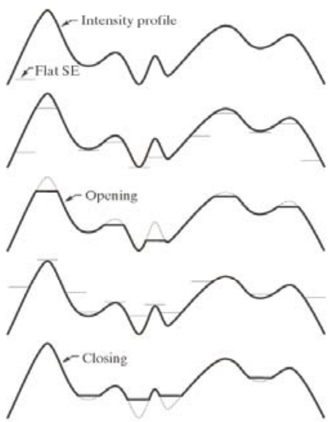


$$f \circ S = (f \,\theta\, S) \oplus S$$
$$= \max(\min(f))$$

$$f \bullet S = (f \oplus S) \,\theta\, S$$
$$= \min(\max(f))$$

Interpretation of grey-level opening and closing

- Intensity values are interpreted as height curves over the (x,y)-plane.
- Opening of f by b:
  Push the structuring element up from below towards the curve f.
  The value assigned is the highest level b can reach.
  smooths bright values down.
- Closing:
  Push the structuring element down from above towards the curve f.
  smooths dark values upwards

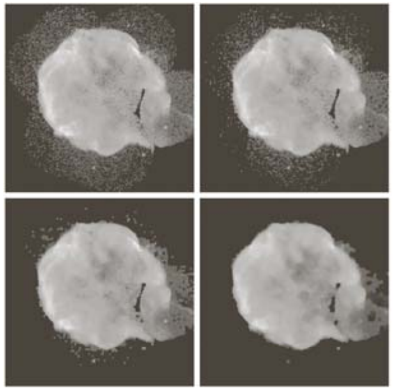


Morphological filtering

- Grey-level opening and closing with flat structure elements can be used to filter out noise.
- This is particularly useful for e.g. dark or bright noise.
- Remark: bright or dark is relative to surroundings, eg. local extremas are filtered.
- To remove bright noise, do first opening, then closing.
  min(max(max(min(f(x,y)))))
  This can be repeated with different SE's.
- The size of the structuring element should reflect the size of the noise objects that we wish to remove.
- NB! "2D Min and Max are separable!

Example – morphological filtering

Original



Opening and closing by disk of size 1

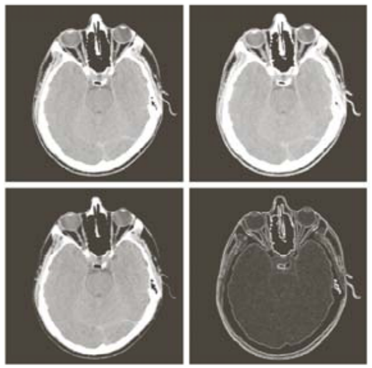
Opening and closing by disk of size 5

Opening and closing by disk of size 3

Morphological gradient

- Gray level dilation will (under some conditions) give an image with equal or brighter values – as it is a local max-operator.
- Erosion will under the same conditions produce an image with equal or lower values - as it is a local min-operator.
- This can be used for edge detection
- Morphological gradient = $(f \oplus S) - (f \ominus S)$

Morphological gradient



Dilation

Gradient
= dilation-erosion

Erosion

Top-hat transformation

- Purpose: detect (or remove) structures of a certain size.
- Top-hat: detects light objects on a dark background
  - also called white top-hat.
- Top-hat:

$$f - (f \circ b)$$

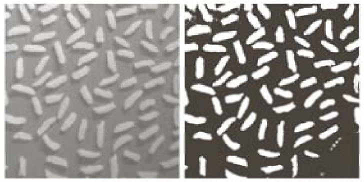
- Bottom-hat: detects dark objects on a bright background
  - also called black top-hat.
- Bottom-hat:

$$(f \bullet b) - f$$

- Very useful when correcting for uneven illumination or objects on a varying background ☺

Example – top-hat

Original, un-even background



Global thresholding using Otsu's method.

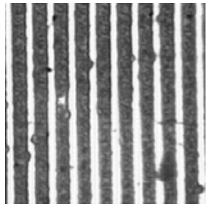
Objects in the lower right Corner disappear. Misclassification of background in upper right corner.

Opening with a 40x40 structuring element removes objects and gives an estimate of the background

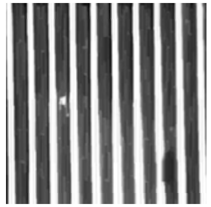
Top-hat transform (original – opening)

Top-hat, thresholded with global threshold

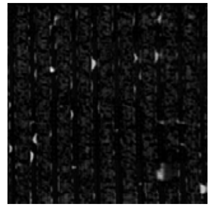
Fault detection using 'Bottom-hat'



Original

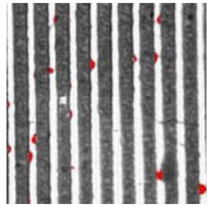


Closing' with vertical structuring element



Bottom-Hat= Original – closing

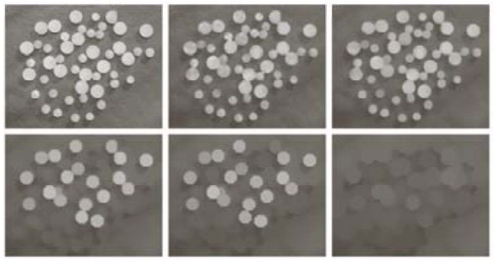
Threholded previous image



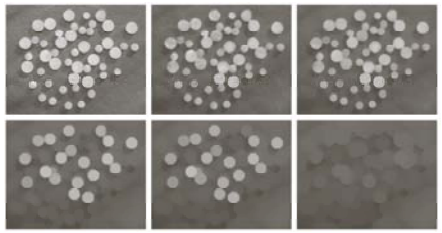
Error detection

Example application: granulometry

- Granulometry: determine the size distribution of particles in an image.
- Assumption: objects with regular shape on a background.
- Principle: perform a series of openings with increasing radius r of structuring element
- Compute the sum of all pixel values after the opening.
- Compute the difference in this sum between radius r and r-1, and plot this as a function of radius.



Example - granulometry



a b c
d e f

FIGURE 9.41 (a) 531 × 675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

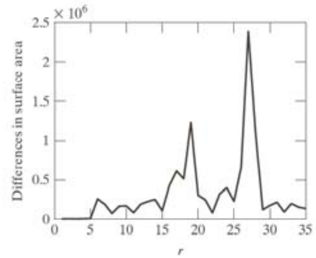


FIGURE 9.42 Differences in surface area as a function of SE disk radius, r. The two peaks are indicative of two dominant particle sizes in the image.

Learning goals - morphology

- Understand in detail binary morphological operations and selected applications:
  - Basic operators (erosion, dilation, opening, closing)
    - Understand the mathematical definition, perform them ”by hand” on new objects
  - Applications of morphology:
    - edge detection, connected components, convex hull etc.
    - Verify the examples in the book
- Grey-level morphology:
  - Understand how grey-level erosion and dilation (and opening and closing) works.
  - Understand the effect these operations have on images.
  - Understand top-hat, bottom-hat and what they are used for.