Mathematical Morphology

Outline:

- **Basic principles**
- **Binary images**
  - Erosion and Dilation
  - Opening and Closing
  - Thinning
- **Grey level images**
  - Basic operations
  - Some applications
- **Mathematical Morphology in Xite**

Mathematical Morphology

Introduction

Initial considerations

Morphological image processing defines a set of operations in which the spatial structure of objects within an image is modified. Erosion, dilation, opening and closing are four basic operations. All these operations can be used using "hit or miss" transformations, where a small mask (the structuring element) is scanned over a binary image. If the pattern of the mask matches the state of the pixel under the mask (hit), then the corresponding output pixel is set to "1"; if there is a mismatch (miss), the output pixel is set to "0".

When used correctly mathematical morphology operations preserve the essential features of shape of an object, while removing the irrelevant details.

Applications

Mathematical morphology finds application where shape and speed is an issue. Analysis of microscopic images (in biology, material science and document analysis), visual inspection and character recognition are domains of application.

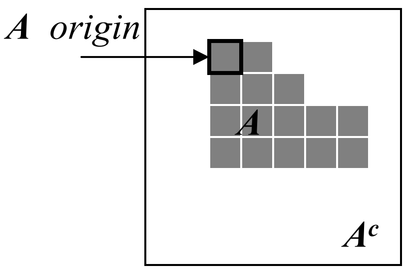
Mathematical Morphology

Basic principles. Binary images

Basic definitions

The images in the mathematical morphology context are sets of points (***A*** and ***B*** in the figure), described in the discrete Z^2 space.

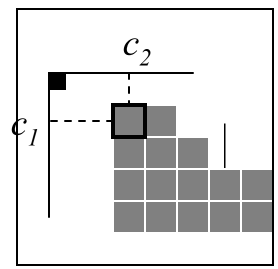
Image



$\boldsymbol{A} = \{a \mid a \in \boldsymbol{A}\}$

$\boldsymbol{A}^c = \{a \mid a \notin \boldsymbol{A}\}$

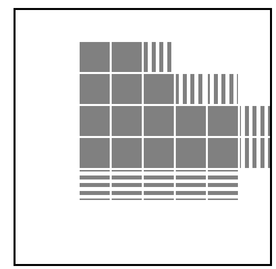
Translation



$(\boldsymbol{A})_x = \{x \mid x = a + c, \text{ for } a \in \boldsymbol{A}\}$

$c = (c_1, c_2)$

Dilation - A ⊕ B



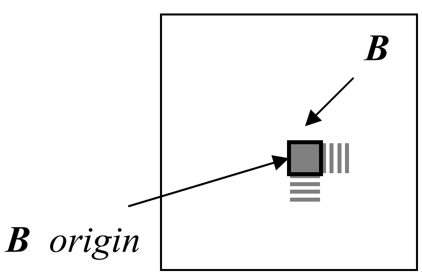
$\boldsymbol{A} \oplus \boldsymbol{B} = \{x \mid x = a + b, \; a \in \boldsymbol{A}, \; b \in \boldsymbol{B}\}$

Dilation

The dilation of ***A*** by ***B***, is the set of all *x* displacements, such that ***B*** and ***A***, overlap by at least one element. A dilation is a Minkowski addition

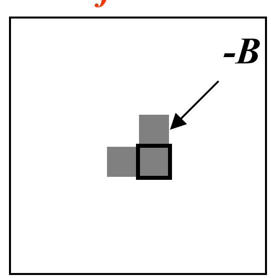
$\boldsymbol{A} \oplus \boldsymbol{B}$

Structuring element



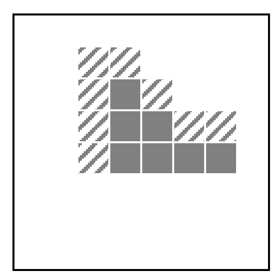
$\boldsymbol{B} = \{b \mid b \in \boldsymbol{B}\}$

Reflection



$\boldsymbol{B}^r = \{x \mid x = -b, \; b \in \boldsymbol{B}\}$

Erosion - A Θ B^r



$\boldsymbol{A} \ominus \boldsymbol{B}^r = \{x \mid x - b \in \boldsymbol{A}, \; b \in \boldsymbol{B}\}$

Erosion

The erosion of ***A*** by ***B***, is the set of all *x* displacements, such that -***B*** translated by *x* is contained in ***A***. It is the Minkowski subtraction $\boldsymbol{A} \ominus \boldsymbol{B}^r$. -***B*** is the 180º rotation of ***B***, around the origin.

If ***B*** is symetric, an erosion is equal to the Minkowski subtraction.

Dilation and erosion properties

Dilation properties

Theorem: $A \oplus B = \bigcup_{b \in B} (A)_b$

Commutative: $A \oplus B = B \oplus A$

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

Translation invariance: $(A)_x \oplus B = (A \oplus B)_x$

Increasing: $if\ A_1 \subset A_2 \ \Rightarrow\ A_1 \oplus B \subset A_2 \oplus B$

Erosion properties

Theorem: $A \Theta B = \bigcap_{b \in B} (A)_{-b}$

Non-Commutative: $A \Theta B \neq B \Theta A$

Translation invariance: $(A)_x \Theta B = (A \Theta B)_x$

Increasing: $if\ A_1 \subset A_2 \ \Rightarrow\ A_1 \Theta B \subset A_2 \Theta B$

Decreasing: $if\ B_1 \subset B_2 \ \Rightarrow\ A \Theta B_1 \supset A \Theta B_2$

Duality

$$(A \Theta B)^c = A^c \oplus B^r$$

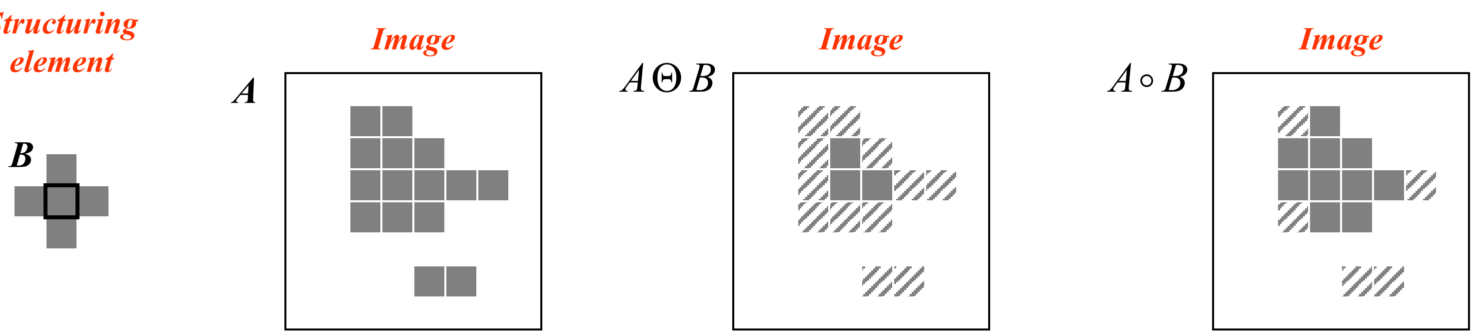
$$A \oplus B = (A^c \Theta B^r)^c$$

This duality properties implies that the dilation of an object is equivalent to the erosion of the background

Opening and closing

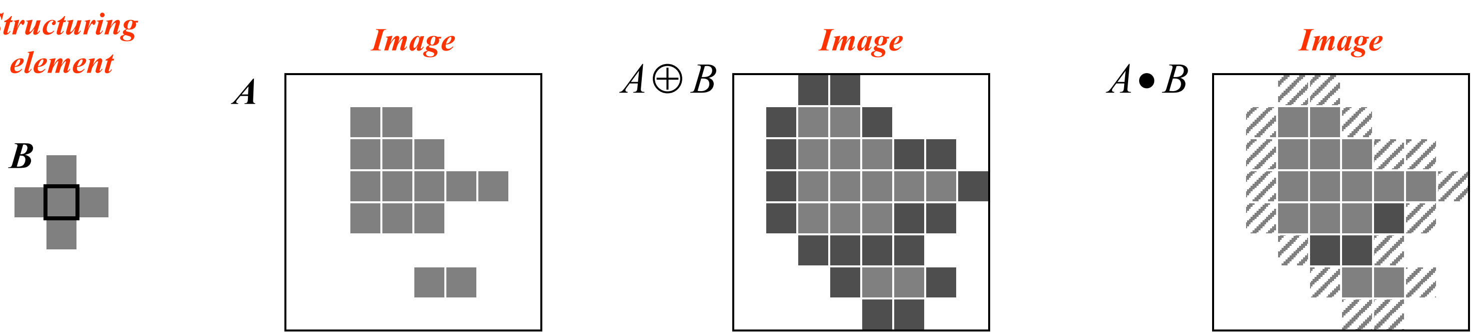
Opening

$$A \circ B = (A \Theta B) \oplus B$$



Closing

$$A \bullet B = (A \oplus B) \Theta B$$



Opening and closing properties

Duality

$$(A \bullet B)^c = A^c \circ B \qquad (A \circ B)^c = A^c \bullet B$$

Translation

$$(A)_x \circ B = (A \circ B)_x \qquad (A)_x \bullet B = (A \bullet B)_x$$

For an opening or closing with a structuring element B, with images A_1 and A_2 ($A_1 \subseteq A_2$), we have the following properties:

Opening

Antiextensivity $A \circ B \subseteq A$

Increasing monotonicity $A_1 \circ B \subseteq A_2 \circ B$

Idempotence $[(A \circ B) \circ B] = A \circ B$

Some consequences of these properties:

- The result of an opening is a subset of the input;
- Monotonicity is preserved
- Apply more than one opening doesn't change the result.
- Opening tends to remove small objects and very thin parts of the object.
- Opening tends to separate objects linked by thin bridges.

Closing

Extensivity $A \subseteq A \bullet B$

Increasing monotonicity $A_1 \bullet B \subseteq A_2 \bullet B$

Idempotence $[(A \bullet B) \bullet B] = A \bullet B$

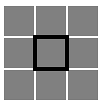
Some consequences of these properties:

- The input is a subset of a closing;
- Monotonicity is preserved
- Apply more than one closing doesn't change the result.
- Closing tends to fill small whole in the object
- Closing tends to link close objects.

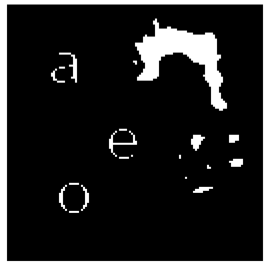
Results

Structuring element

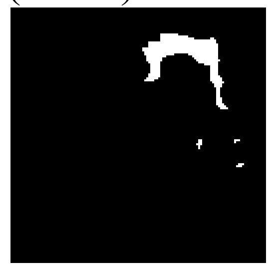
B



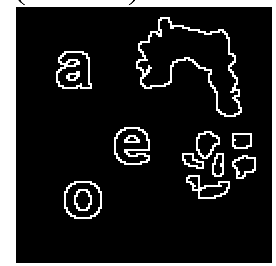
$A \Theta B$



$(A \Theta B) \Theta B$



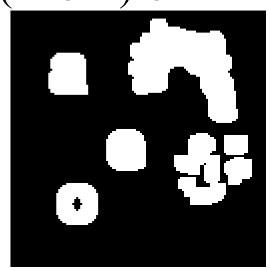
$(A \oplus B) - A$



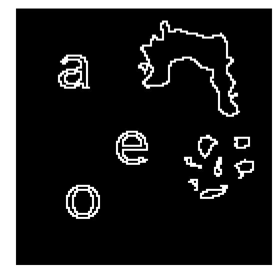
A *Image*

$A \oplus B$

$(A \oplus B) \oplus B$



$A - A \Theta B$



$A \circ B$

$A \bullet B$

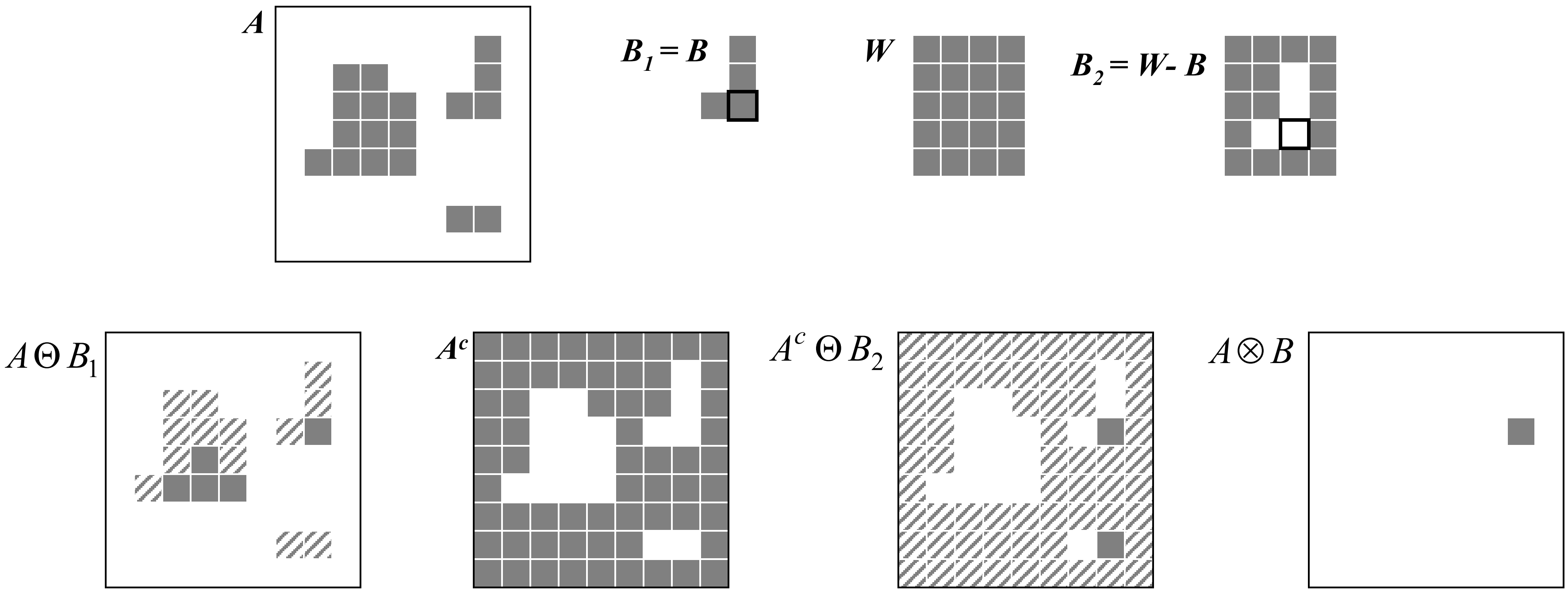
Difference of two sets

$$A - B = \{x \mid x \in A, x \notin B\} = A \cap B^c$$

Hit and miss transform

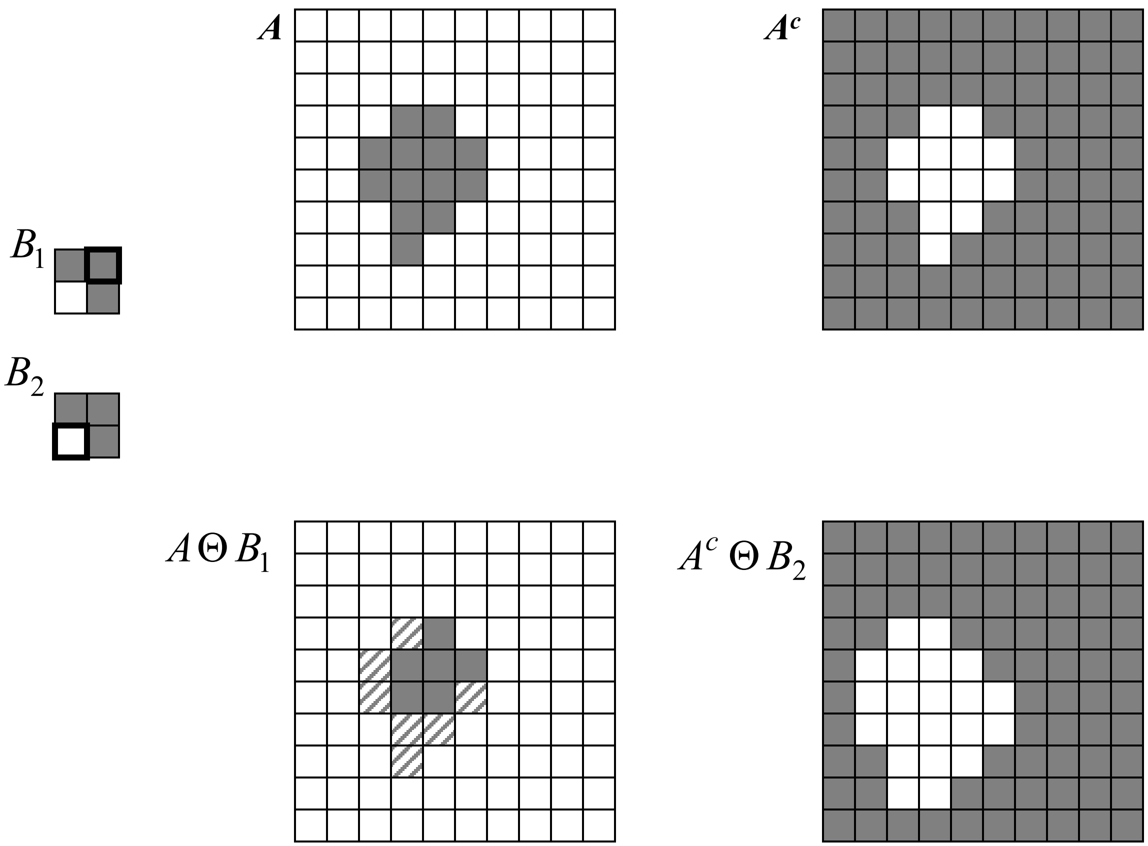
$$A \otimes B = (A \Theta B) \cap [A^c \Theta (W - B)]$$

$$A \otimes B = (A \Theta B_1) \cap [A^c \Theta B_2] \qquad if \;\; B_1 = B \;\; and \;\; B_2 = W - B$$

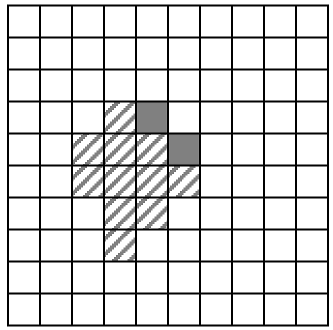


Hit and miss transform

Finding corners



$$A \otimes B = (A \Theta B_1) \cap [A^c \Theta B_2]$$

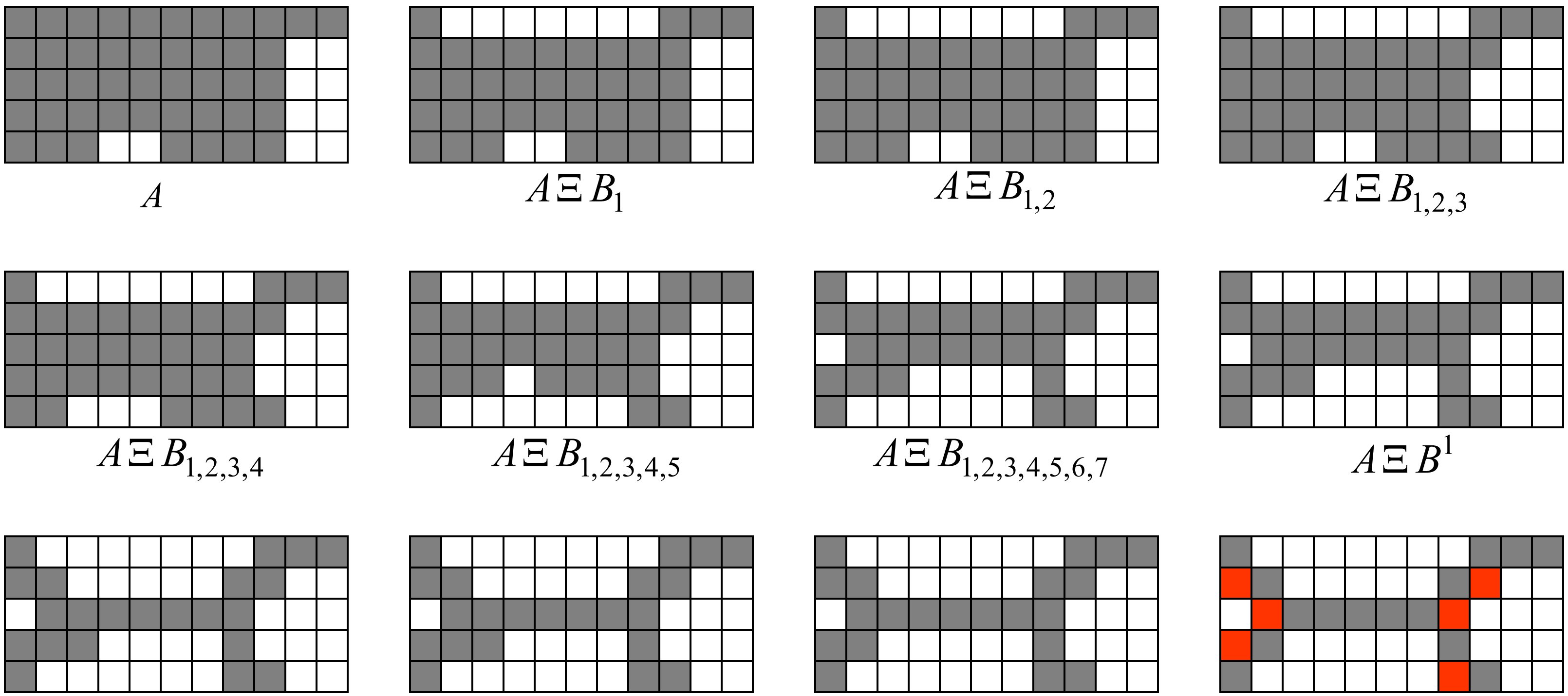


Thinning

$$A \Xi B = A - (A \otimes B) \quad or \quad A \Xi \{B\} = ((\cdots((A \Xi B^1) \Xi B^2)\cdots) \Xi B^n)$$

Don't care

B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8

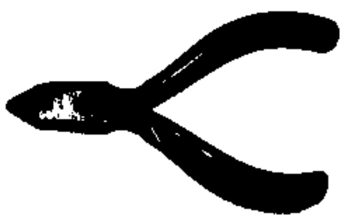


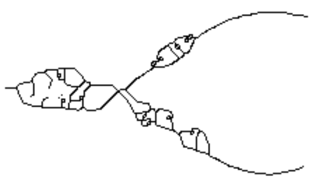
Thinning

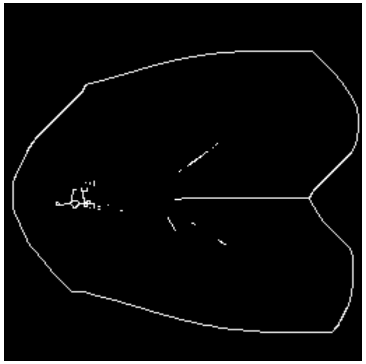
Input

Thinning result

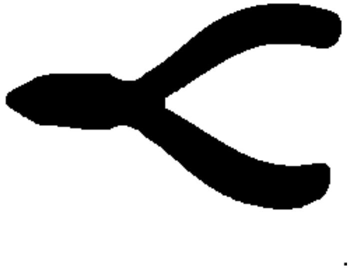
Exo-skeleton

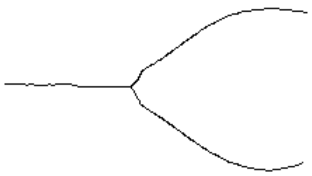


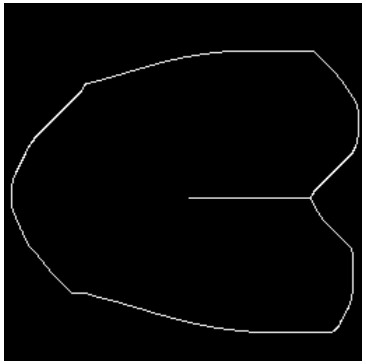


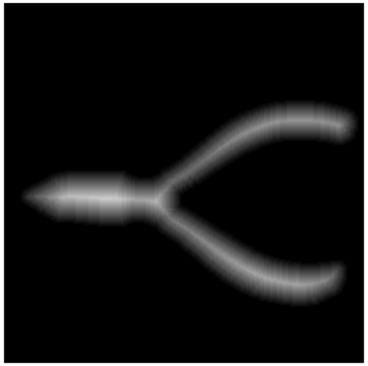


Distance Transform









Mathematical Morphology

Grey level images. Dilation

Dilation $f \oplus b = \max\{f(s-x,\ t-y)+b(x,y)\,|\,(s-x,\ t-y)\in D_f;(x,y)\in D_b\}$

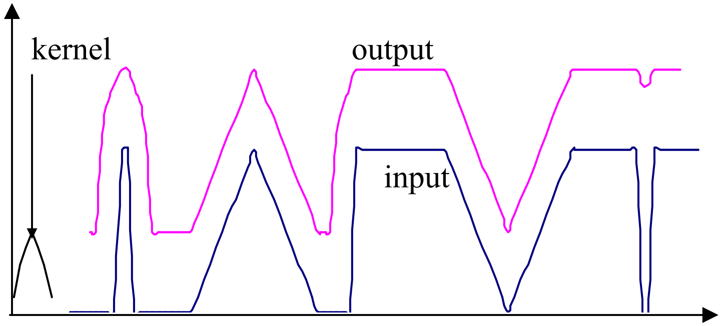
Exemplos:

s	4	5	6	7	8	9	10	11
$f(s)$	19	21	22	10	19	22	20	21

x	-1	0	1
$b(x)$	0	0	0

$$f \oplus b = \max\{f(s-x,\ t-y)\,|\,(s-x,\ t-y)\in D_f;(x,y)\in D_b\}$$

s	4	5	6	7	8	9	10	11
$f \oplus b$		19	22	22	22	22	22	



Mathematical Morphology

Grey level images. Erosion

Erosion $f\,\Theta\,b = \min\{f(s-x,\ t-y) - b(x,y) \mid (s-x,\ t-y) \in D_f; (x,y) \in D_b\}$

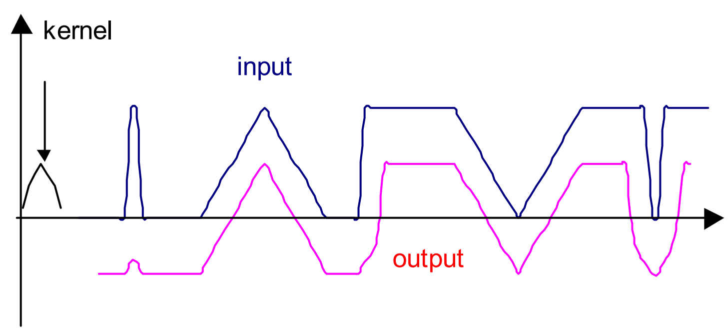
Exemplos:

s	4	5	6	7	8	9	10	11
$f(s)$	19	21	22	40	19	22	20	21

x	-1	0	1
$b(x)$	0	0	0

$$f\,\Theta\,b = \min\{f(s-x,\ t-y)$$
$$\mid (s-x,\ t-y) \in D_f; (x,y) \in D_b\}$$

s	4	5	6	7	8	9	10	11
$f\Theta b$		19	21	19	19	19	20	



Mathematical Morphology

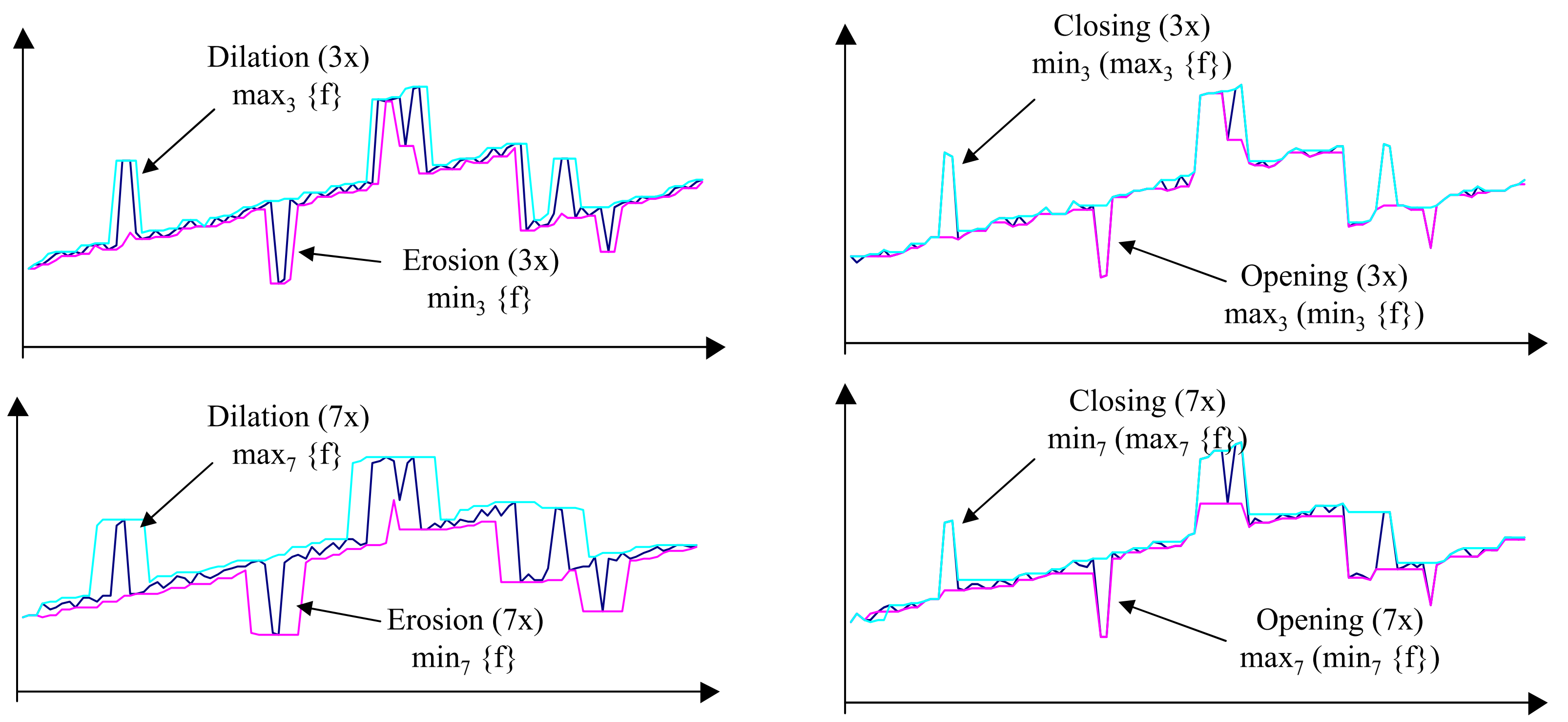
Grey level images. Opening and Closing. Examples

Opening $f \circ b = (f \,\Theta\, b) \oplus b$

Closing $f \bullet b = (f \oplus b) \,\Theta\, b$

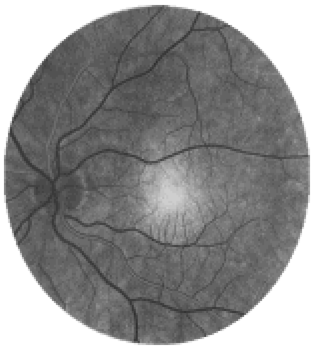
The properties of binary morphological operations can also be applied to gray level operations.

Examples

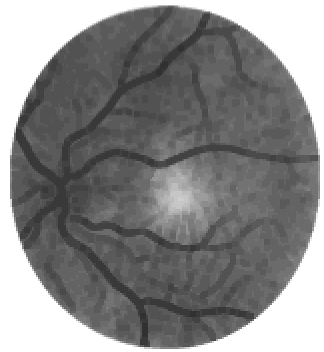


Mathematical Morphology

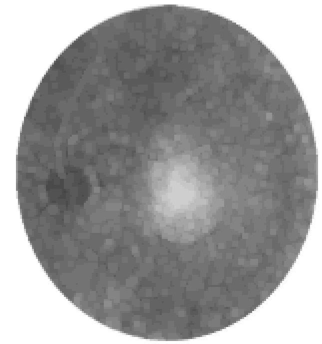
Grey level images. Examples



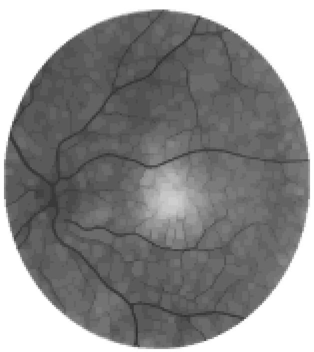
Input



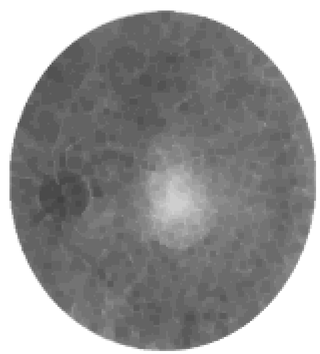
Erosion



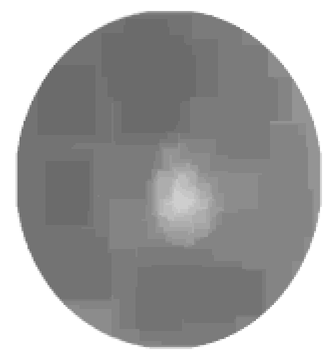
Dilation



Opening



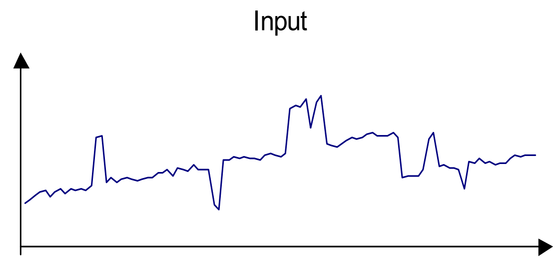
Closing



Dilate-Erode (10 x)

Mathematical Morphology

Grey level images. Examples



Morphological smoothing

$$s = (f \circ b) \bullet b$$

It is an opening followed by a closing operation. Removes both bright and dark artifacts of noise.

Morphological gradient

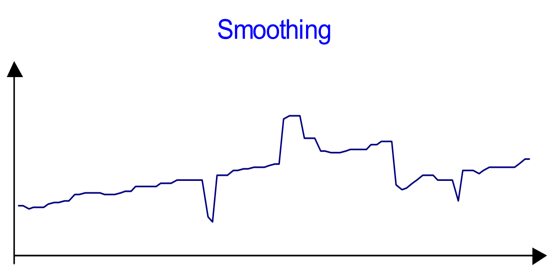
$$g = (f \oplus b) - (f \,\Theta\, b)$$

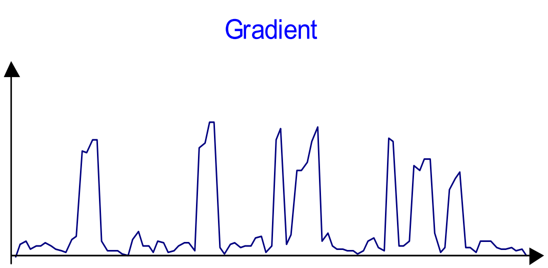
It is the difference between a dilation and an erosion. Highlights sharp grey level transitions

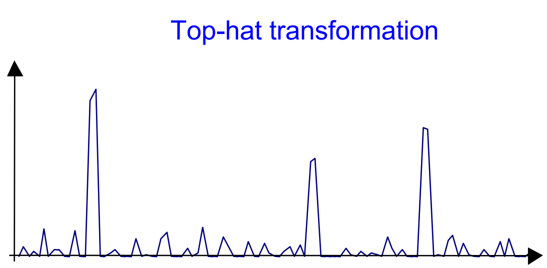
Top-hat transformation

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

It is the difference between the input and the output of an opening operation. Enhances the details (in the signal detects the thin sharp positive variations).







Mathematical Morphology

Xite

Name

morphDilate, morphErode, morphClose, morphOpen - morphological grayscale operations on an image

Syntax

morphDilate <-a | -b> <struct_element> <inimage> <outimage>

morphErode <-a | -b> <struct_element> <inimage> <outimage>

morphClose <-a | -b> <struct_element> <inimage> <outimage>

morphOpen <-a | -b> <struct_element> <inimage> <outimage>

Description

morphDilate calculates the morphological operation dilate on the image inimage, with a structuring element given by struct_element (in combination with option -a or -b). The result is sent to the image specified in outimage.

morphErode calculates the morphological operation erode.

morphClose calculates the morphological operation close.

morphOpen calculates the morphological operation open.

Mathematical Morphology

Xite

The actual calculation for the different operations are described by the expressions below (where i and j are inside the structuring element B, and (0,0) is the center pixel of B).

dilate : outband(x,y) = max[inimage(x+i,y+j)+B(i,j)]

erode: outband(x,y) = min[inimage(x+i,y+j)-B(-i,-j)]

close: outband = morphErode(morphDilate(inband))

open: outband = morphDilate(morphErode(inband))

The output values are clipped outside the interval [0, 255] to make the result fit in pixels of type unsigned byte.

Be careful not to use too large values in B. If, e.g. with dilation, *inimage* is binary with the two values 0 and 255, and all the pixels in *struct_element* equal 255, then all the pixels in *outimage* will equal 255, regardless of the distribution of zeros in *inimage*.

Mathematical Morphology

Xite

The actual calculation for the different operations are described by the expressions below (where i and j are inside the structuring element B, and (0,0) is the center pixel of B).

dilate : outband(x,y) = max[inimage(x+i,y+j)+B(i,j)]   **erode:** outband(x,y) = min[inimage(x+i,y+j)-B(-i,-j)]

close: outband = morphErode(morphDilate(inband))   **open:** outband = morphDilate(morphErode(inband))

The output values are clipped outside the interval [0, 255] to make the result fit in pixels of type unsigned byte. Be careful not to use too large values in B. If, e.g. with dilation, *inimage* is binary with the two values 0 and 255, and all the pixels in *struct_element* equal 255, then all the pixels in *outimage* will equal 255, regardless of the distribution of zeros in *inimage*.

Restrictions

inimage must have pixel type unsigned byte. *struct_element* must be of pixel type signed short (with option -b) or ascii characters (with option -a).

Options

-a struct_element : struct_element is an ascii file which contains the structuring element. See ascii2biff(1) for file format.
-b struct_element : struct_element is a BIFF image which contains the structuring element. Pixel type must be signed short.

Mathematical Morphology

Xite

thn_zs

Name: thn_zs - thin a binary image using algorithm by Zhang and Suen.

Syntax: thn_zs <inimage> <outimage>

Description: Thinning a binary image with algorithm by Zhang and Suen[1]

Restrictions: inimage must have bands with pixel type unsigned byte. The input is assumed to be binary valued with foreground = 0, background = 255.

References

[1] T. Y. Zhang and C. Y. Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns.", Comm. ACM, vol. 27, no. 3, pp 236-239, 1984.
[2] Rafael C. Gonzalez and Paul Wintz, "Digital Image Processing.", 2. edition, 1987, pp. 398-402.

Files: XITE_HOME/src/thin/thn_zs.c

Mathematical Morphology

Xite

thn_lc

Name: thn_lc - Lee and Chen's thinning method.

Syntax: thn_lc <inimage> <outimage>

Description: Thin a binary image with Lee and Chen's method. Black (0) is foreground, and white (255) is background. This method is simply an extension of Zhang and Suen's method. First, Zhang and Suen's method is used. The skeleton thus obtained is not truly 8-connected, since some non-junction pixels have more than two neighbors, making the skeleton useless for algorithms that require this constraint. Therefore, some pixels have to be removed. The skeleton is inspected, and each pixel is tested using a lookup table. The result is a true 8-connected skeleton where only junction pixels have more than two 8-neighbors.

Restrictions: inimage must have bands with pixel type unsigned byte. The input is assumed to be binary valued with foreground = 0, background = 255.

Reference:
H.J. Lee and B. Chen, "Recognition of handwritten chinese characters via short line segments," Pattern Recognition, vol. 25, no. 5, pp. 543-552, 1992.

Files: XITE_HOME/src/thin/thn_lc.c