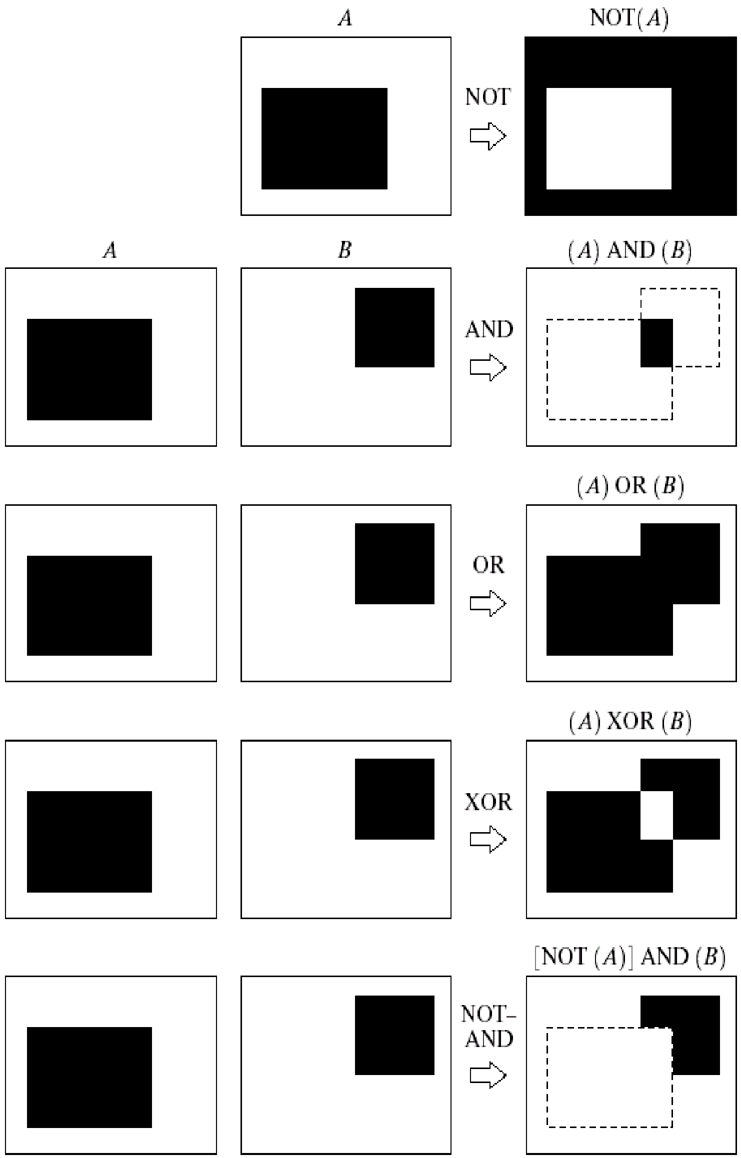
Mathematical Morphology first part: 2D

Mathematical Morphology

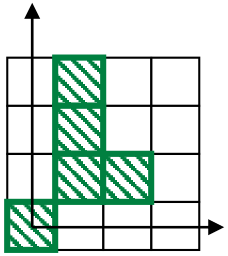
- Mathematical Morphology was developed in France (G. Motheron e J. Serra, Ecole des Mines) and in different form with the name Image Algebra in USA (S. R. Sternberg, Michigan University).

Logic operators between binary images

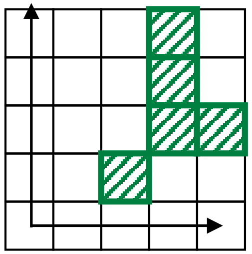


Preliminary Statements

- $A \subseteq E^n$, $t \in E^n$



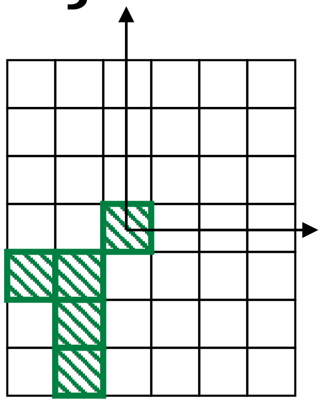
A



$A_{(2,1)}$

- Translation of A by a vector t

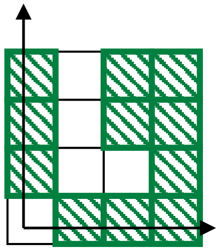
  $A_t = \{ c \in E^n \mid c = a + t, a \in A \}$



A^r

- Reflection of A

  $A_r = \{ c \mid c = -a, a \in A \}$



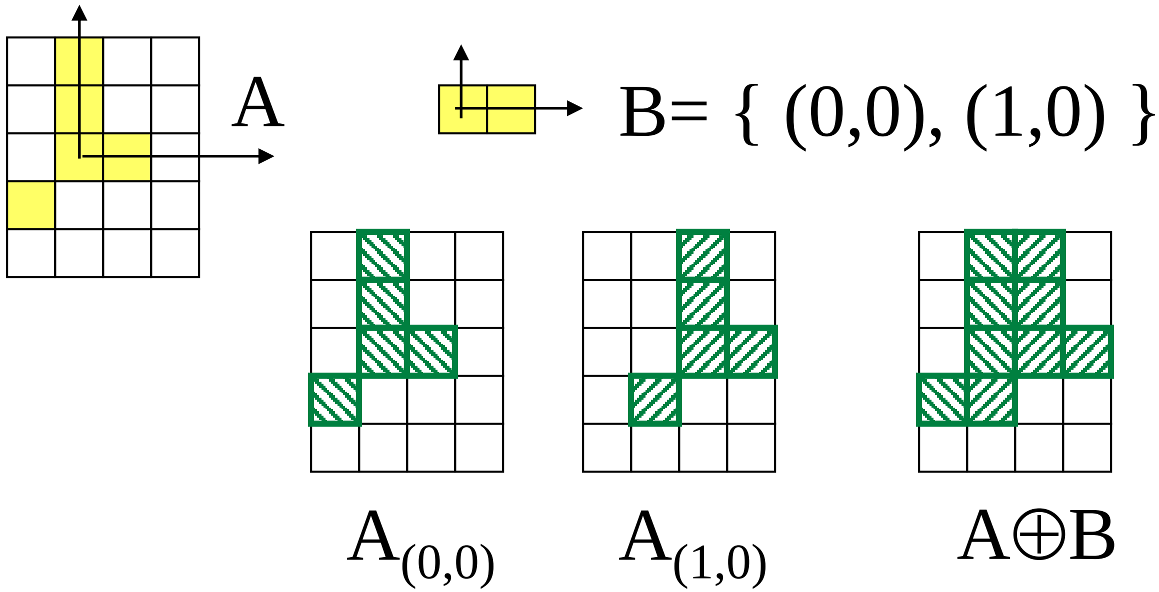
A^c

- Complement of A

  $A_c = E^n - A$

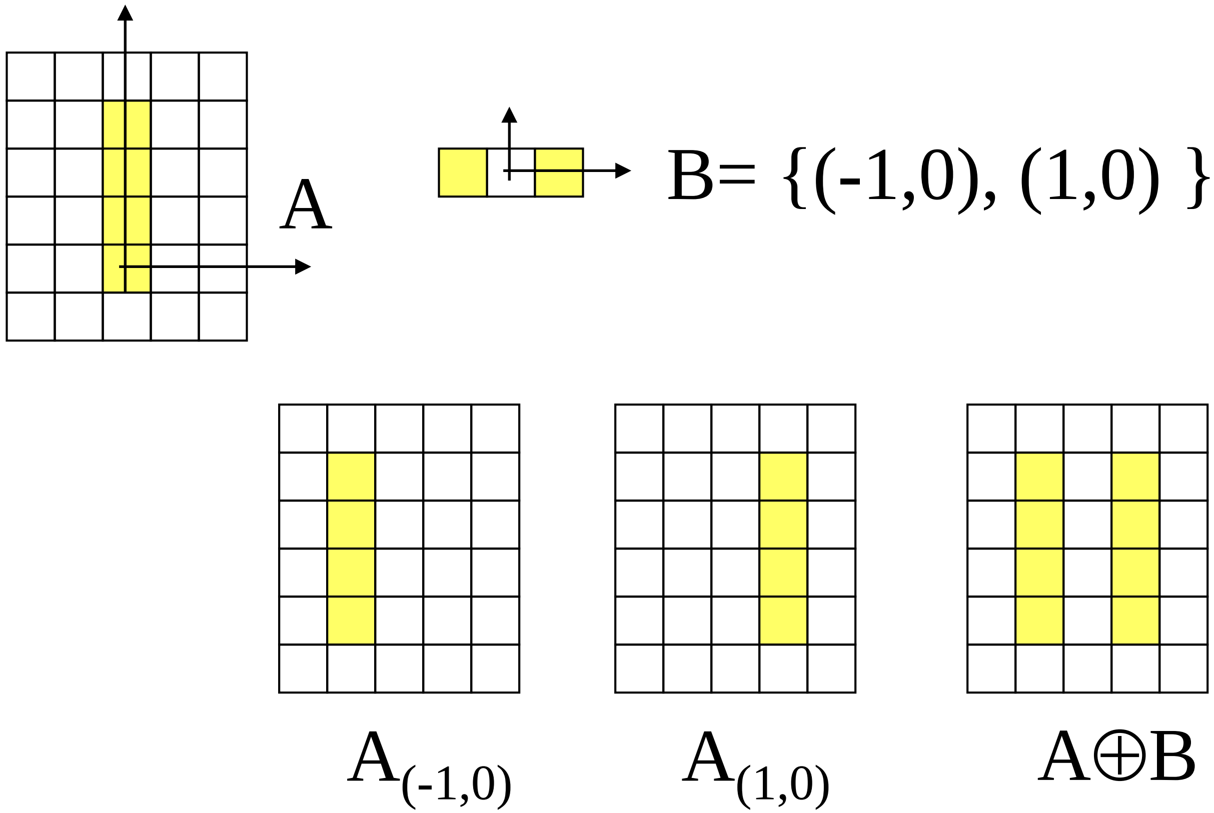
Minkowski sum (Dilation)

- $A \oplus B = \{ c \in E^n \mid c=a+b,\ a \in A,\ b \in B \}$
- $A \oplus B = \cup\ A_b\ ,\ b \in B$
  - It can be easily shown that: $A \oplus B = B \oplus A$

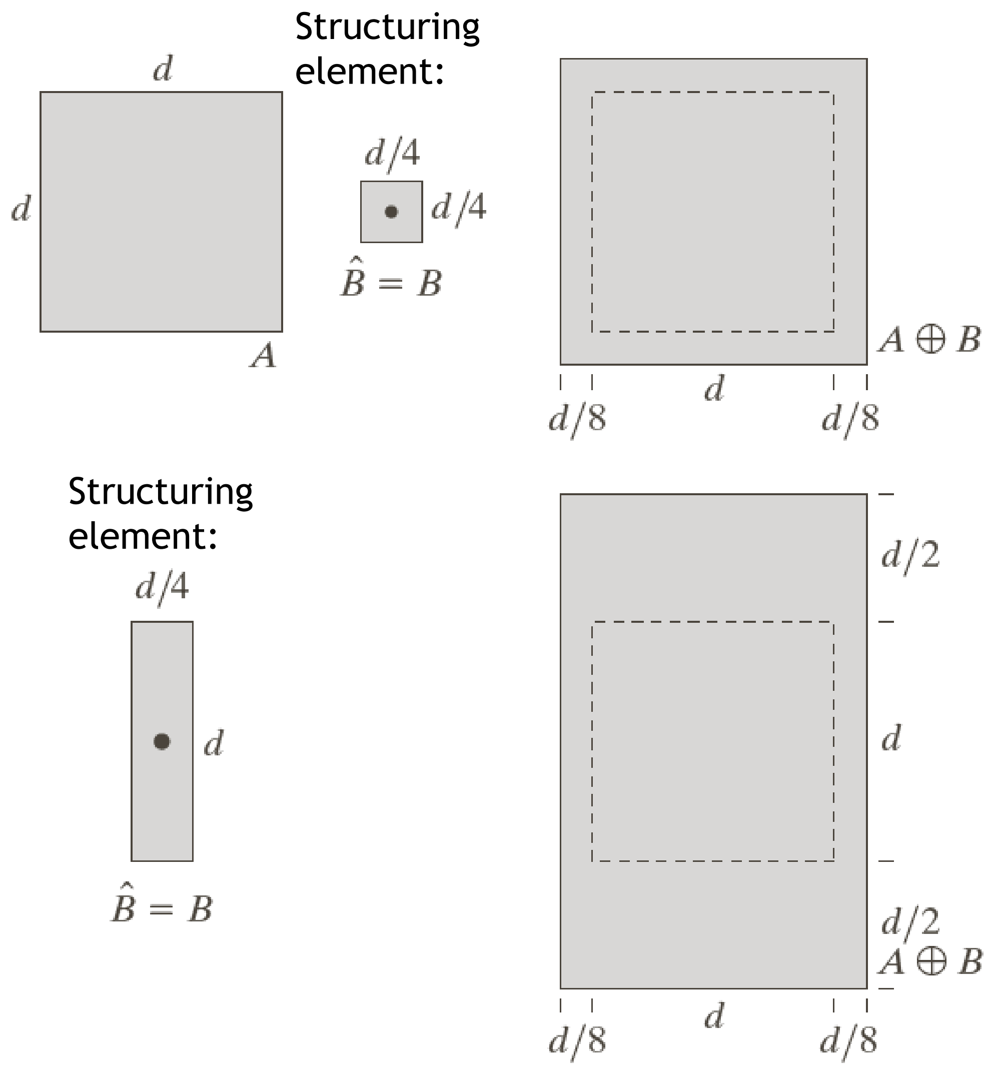


Dilation

- B is usually called ***structuring element***



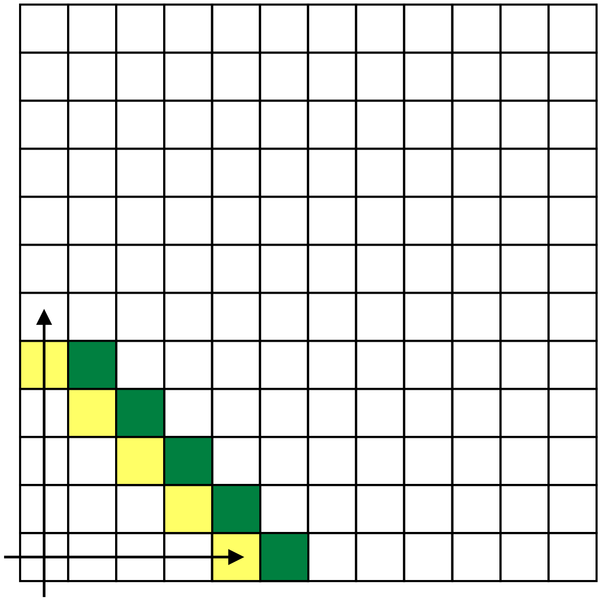
Dilation



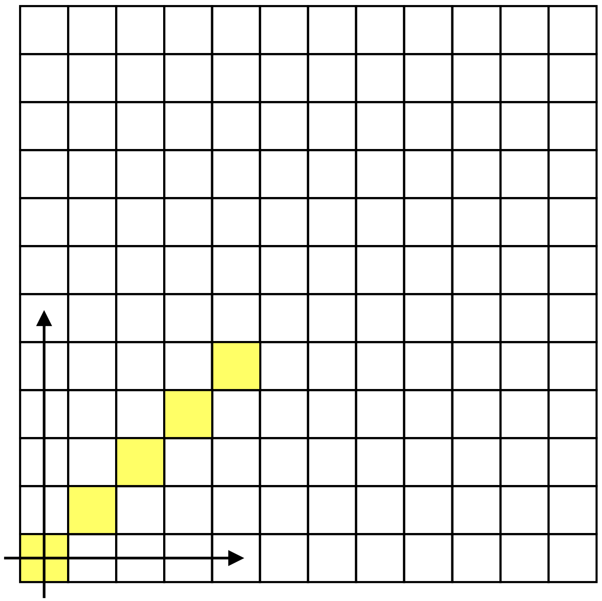
Gonzales-Woods

Dilation

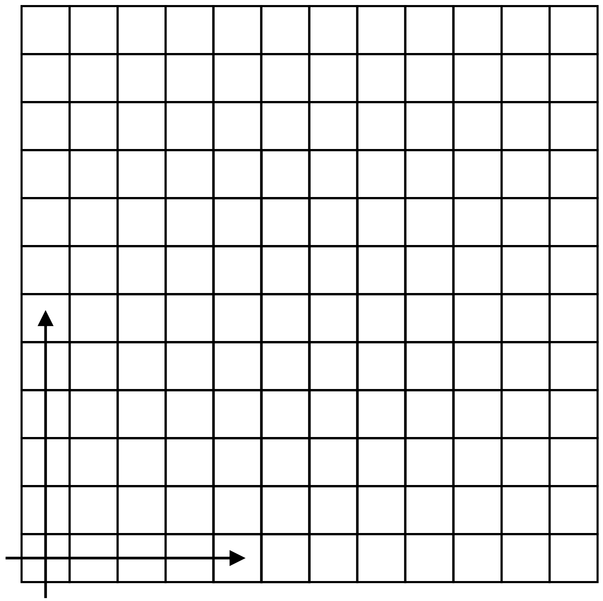
Structuring element:



B



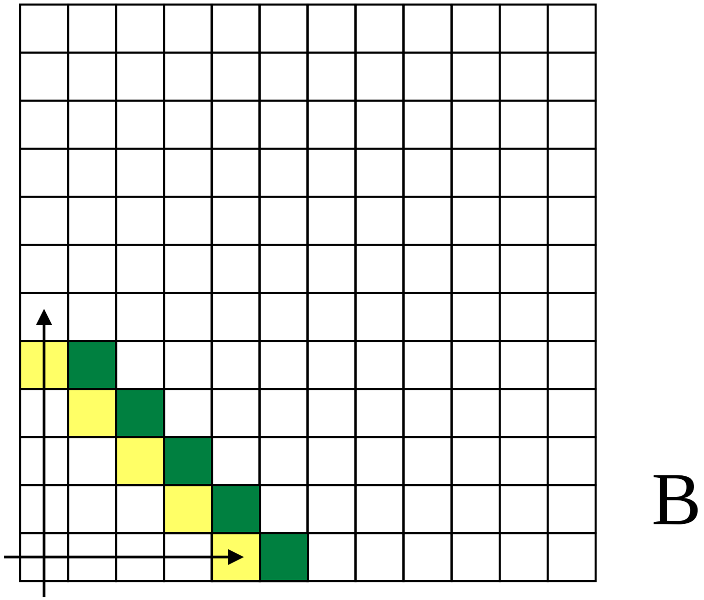
A

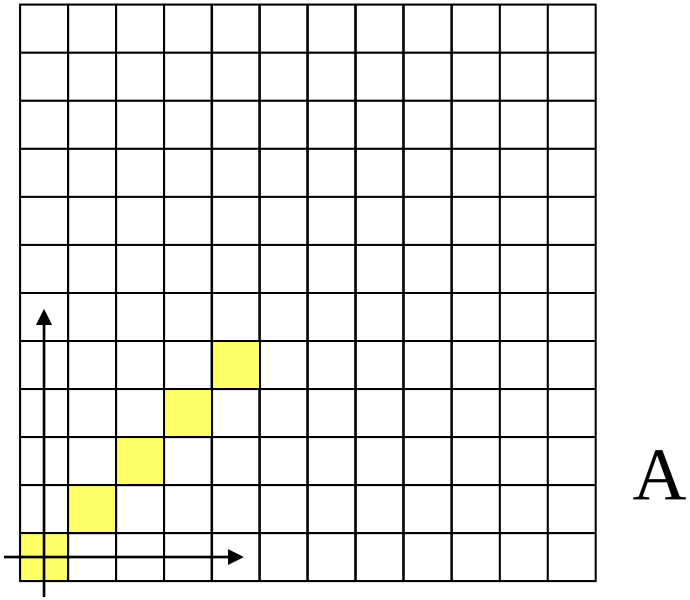


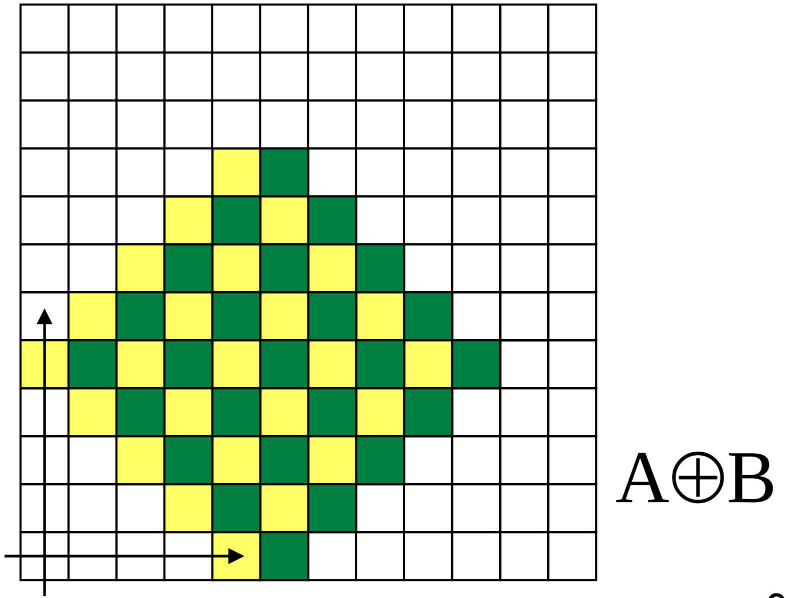
$A \oplus B$

Dilation

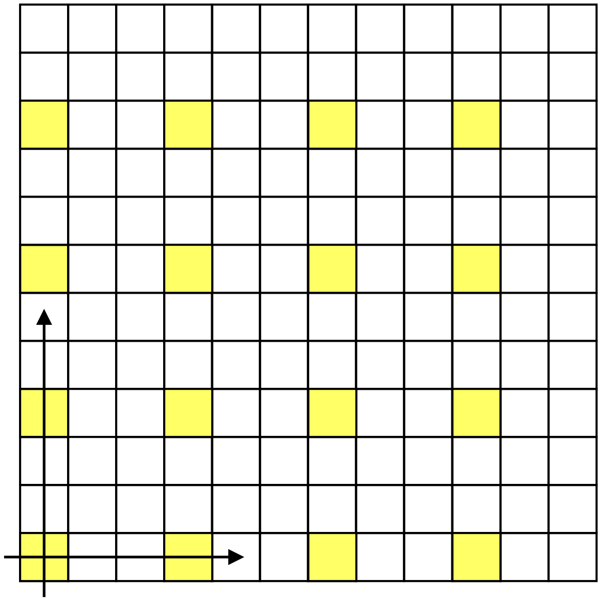
Structuring element:





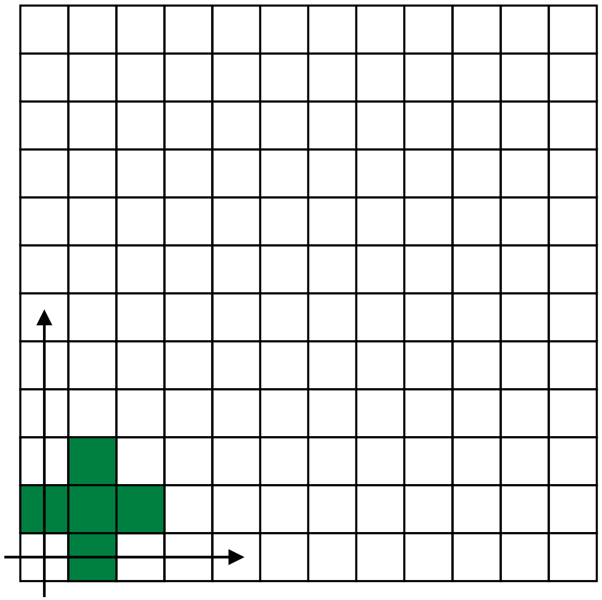


Dilation

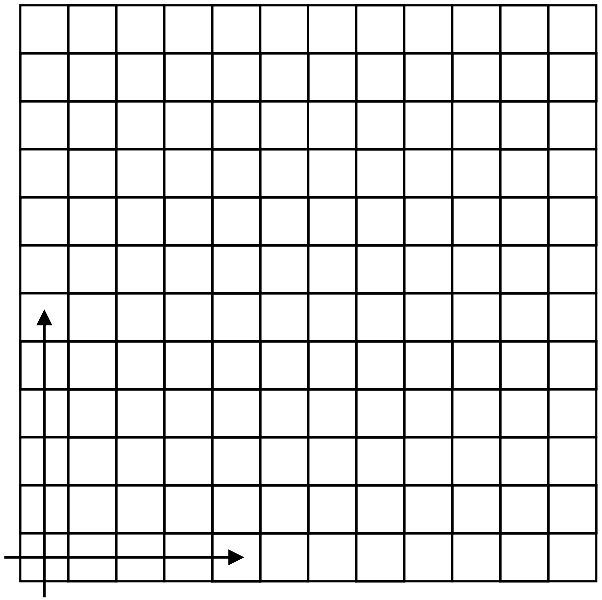


A

Structuring element:

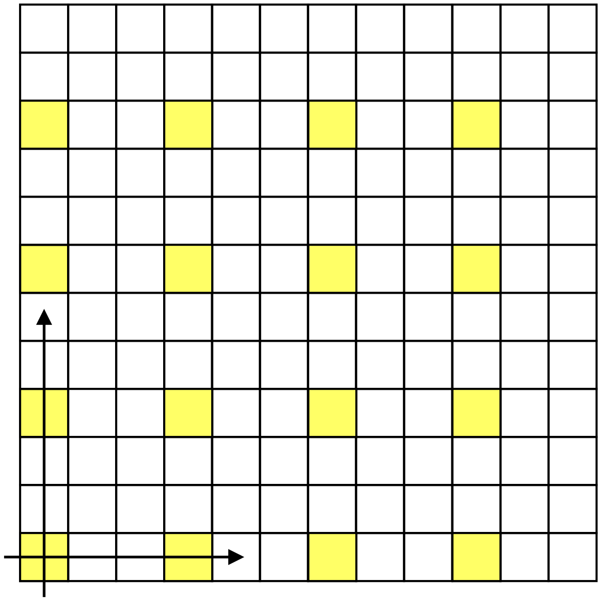


B



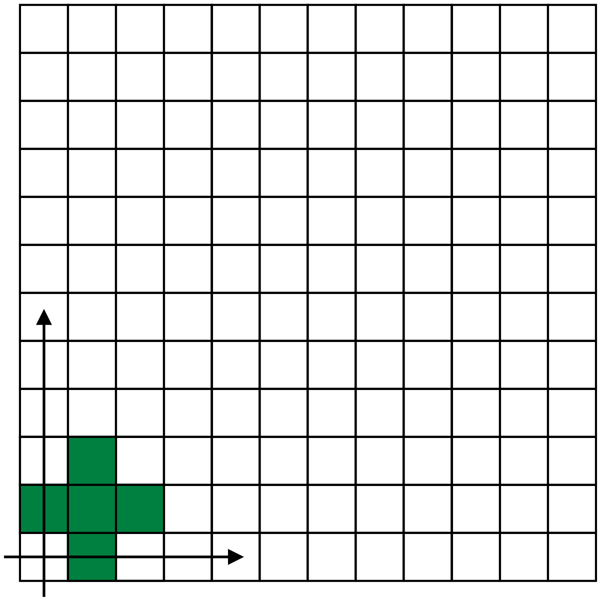
$A \oplus B$

Dilation

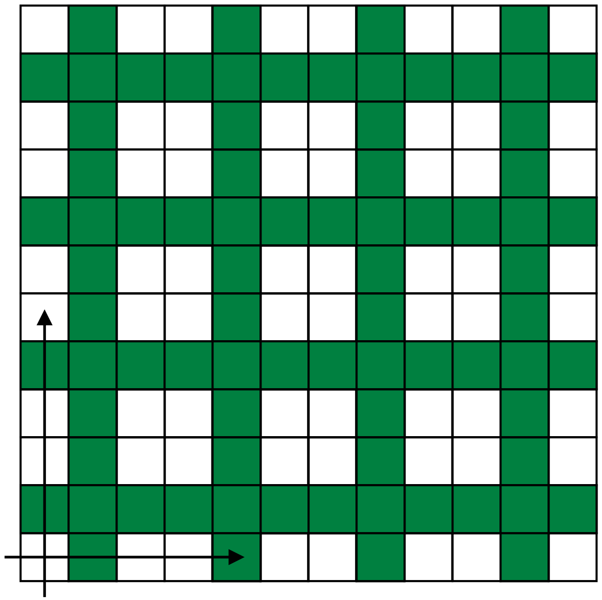


A

Structuring element:

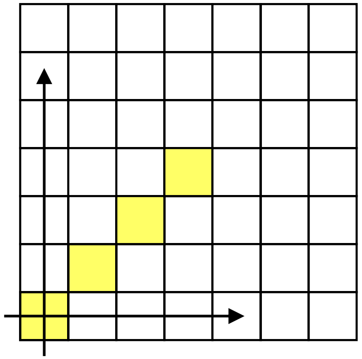


B



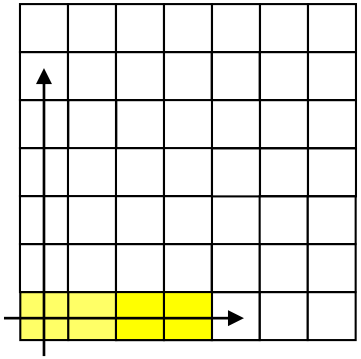
A⊕B

Dilation



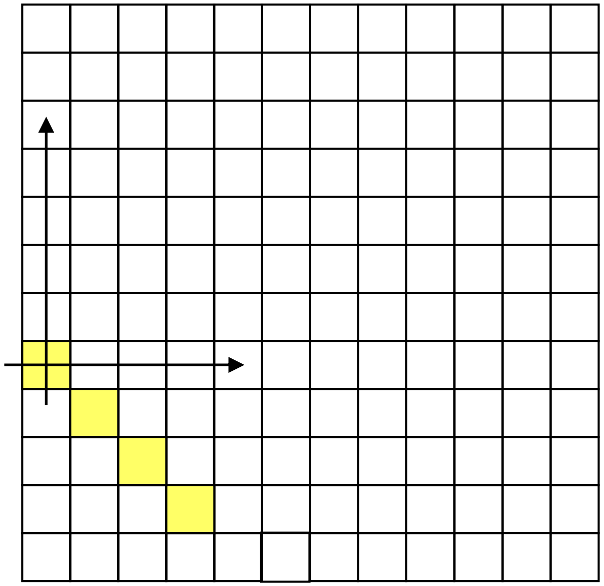
A

Structuring element 1:

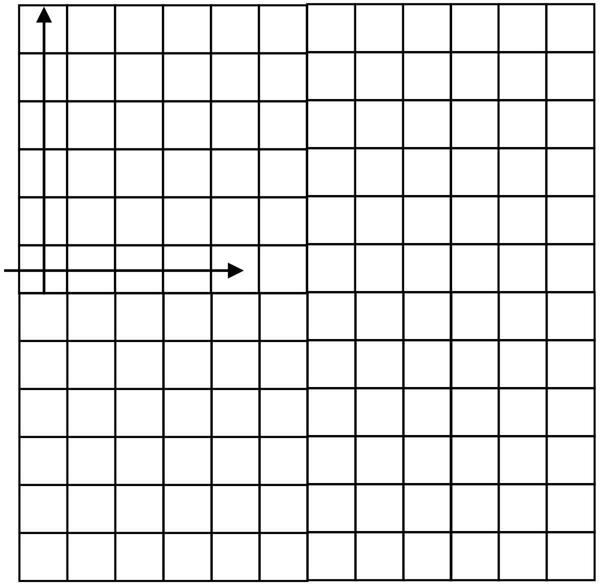


B

Structuring element 2:

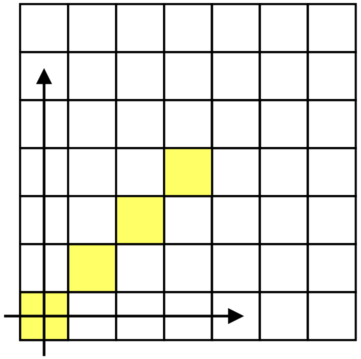


C

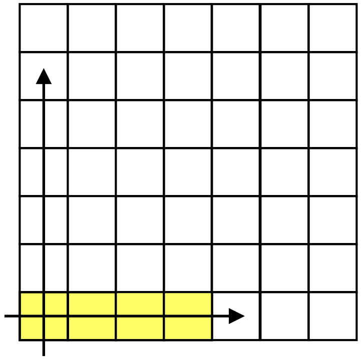


A⊕B⊕C

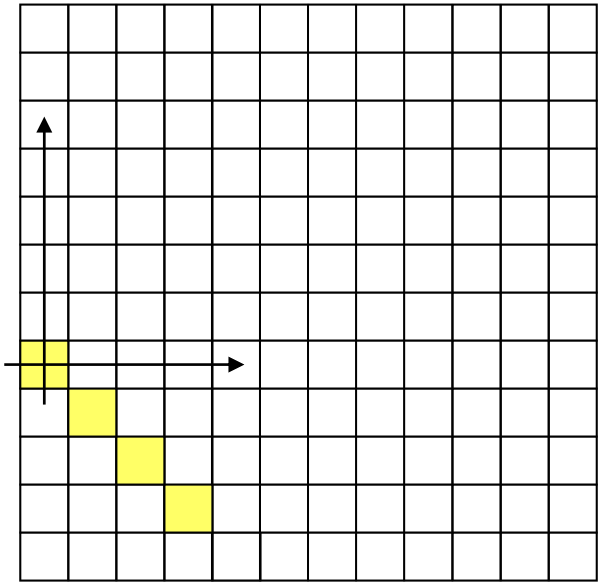
Dilation



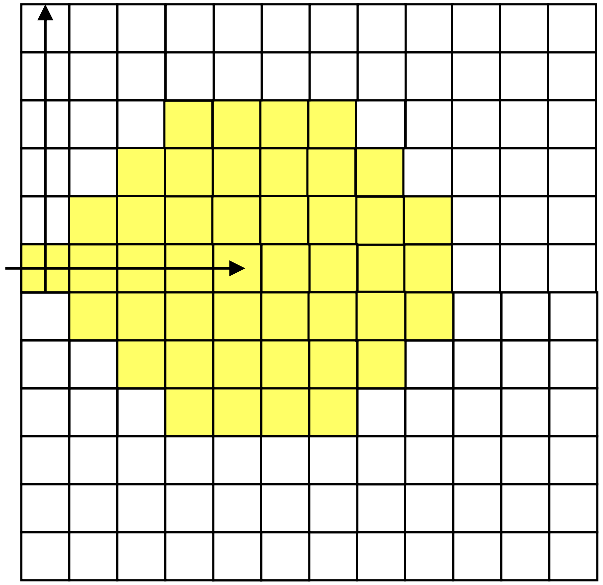
A



B

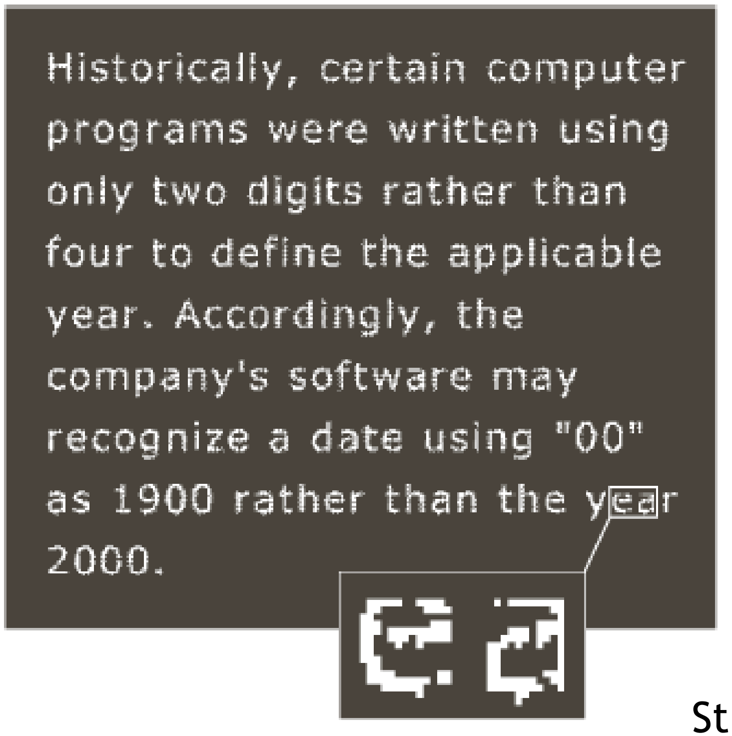


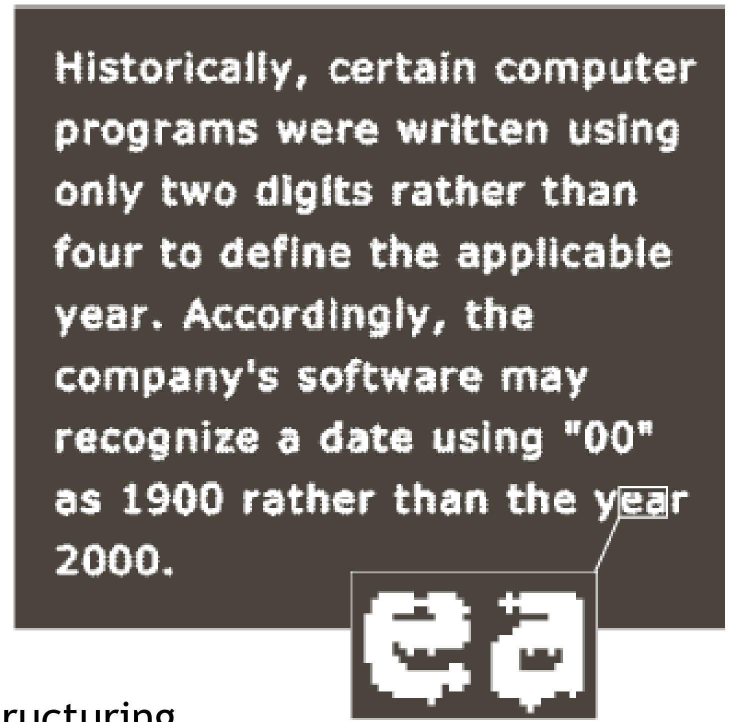
C



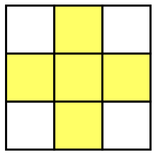
A⊕B⊕C

Dilation





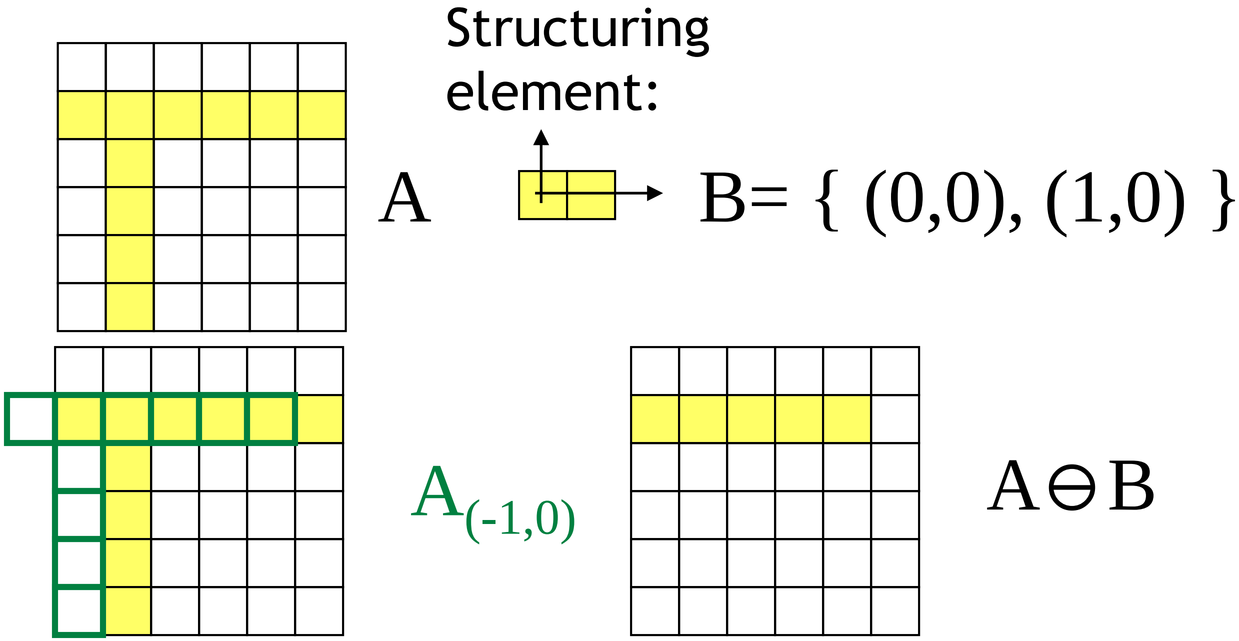
Structuring element:



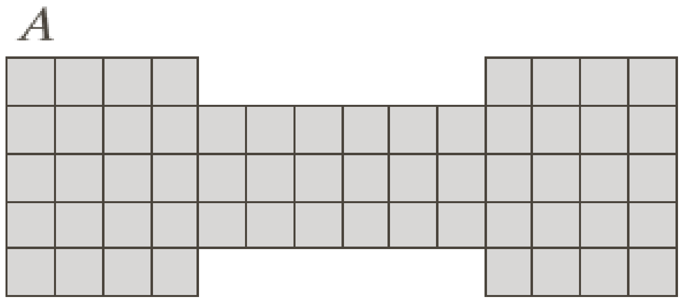
Gonzales-Woods

Minkowski difference (Erosion)

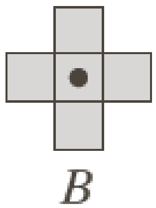
- $A \ominus B = \{ c \in E^n \mid c + b \in A, \text{ per ogni } b \in B \}$
- $A \ominus B = \cap A_{-b} \; b \in B$
- $A \ominus B = \{ c \in E^n \mid B_c \subset A$

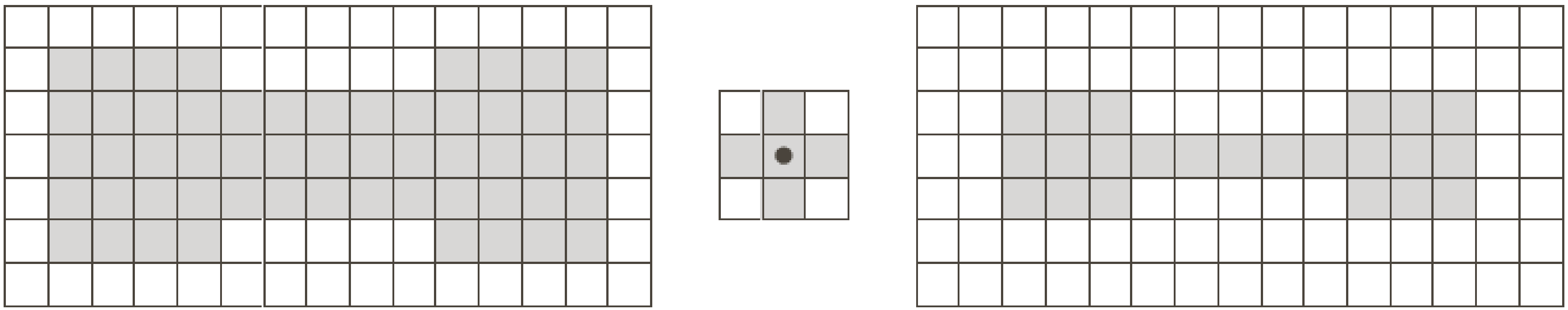


Erosion



Structuring element:

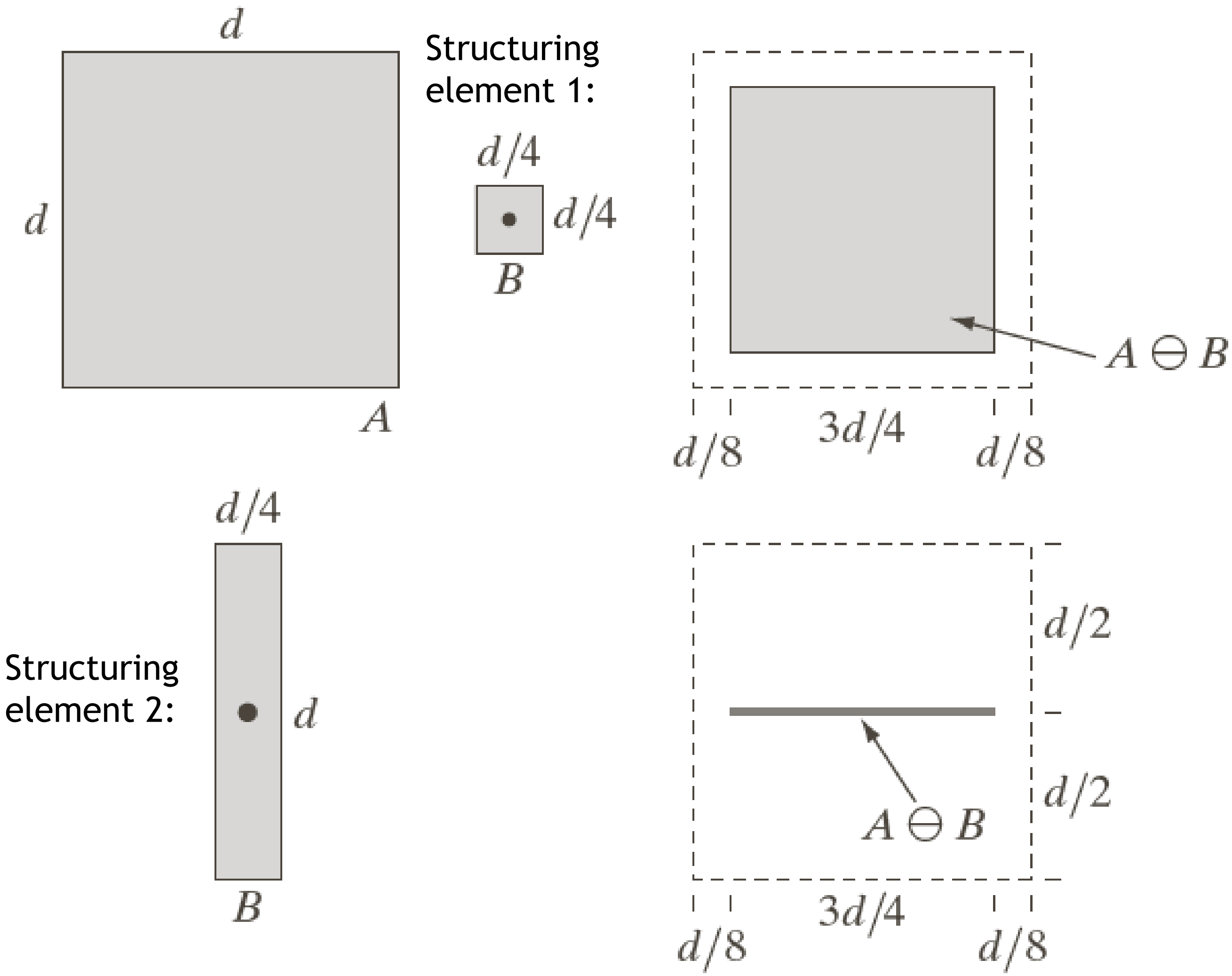




$$A \ominus B$$

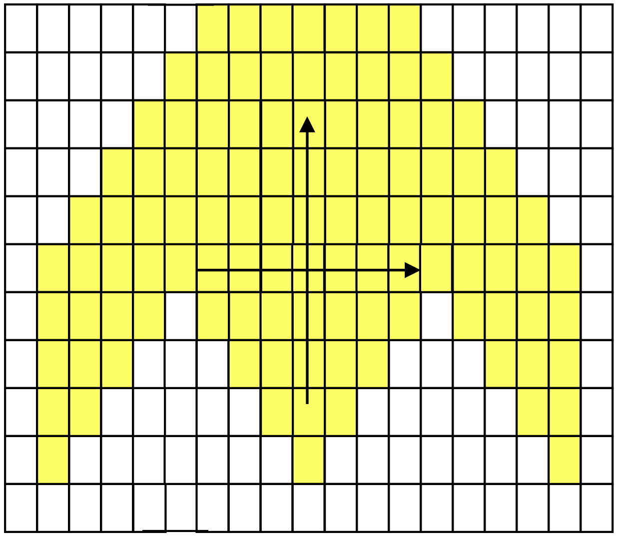
Gonzales-Woods

Erosion



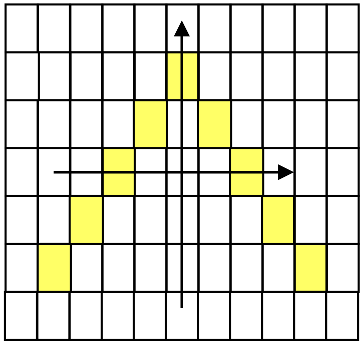
Gonzales-Woods

Erosion

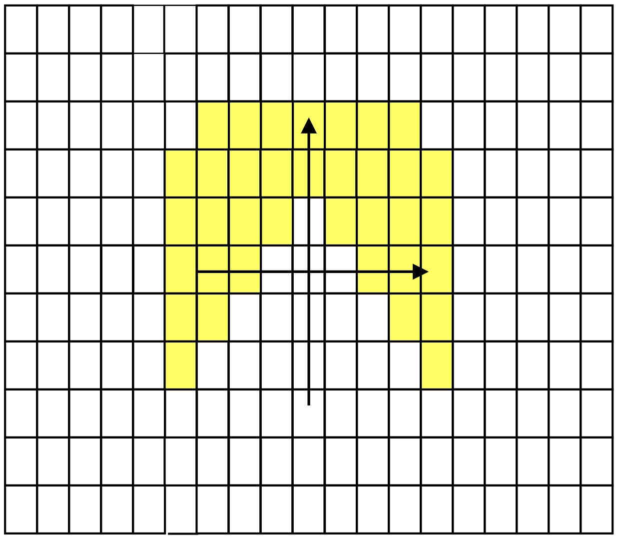


A

Structuring element:

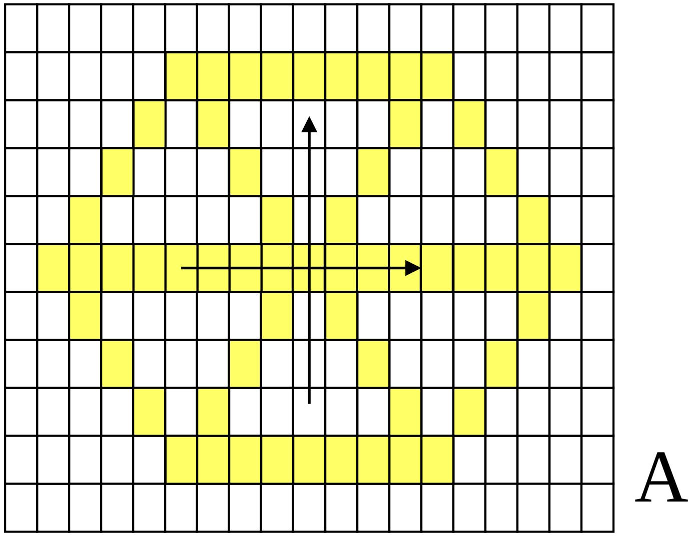


B

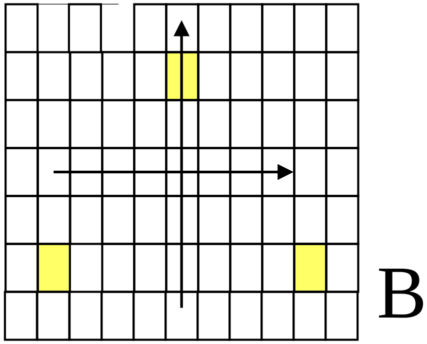


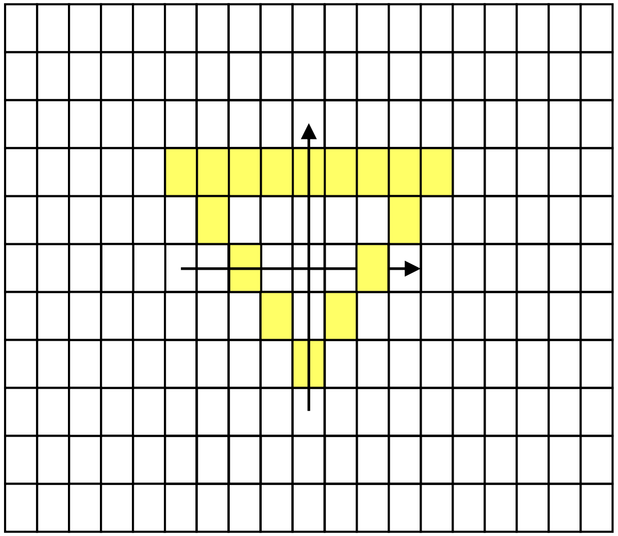
$A \ominus B$

Erosion



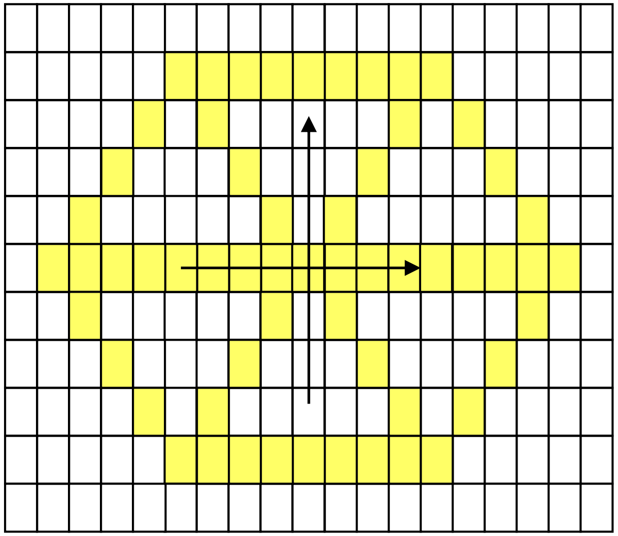
Structuring element:





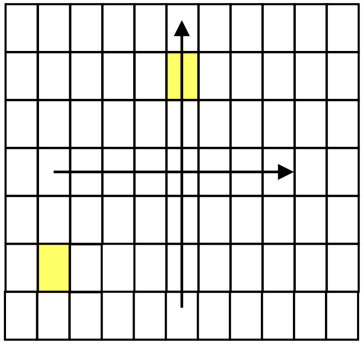
$A \ominus B$

Erosion

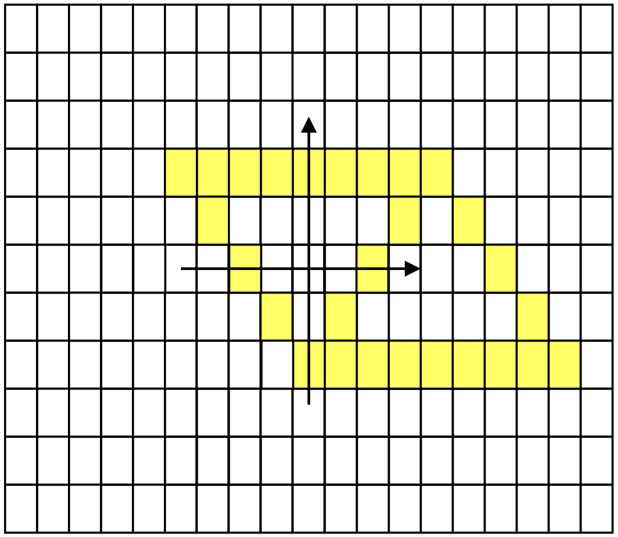


A

Structuring element:



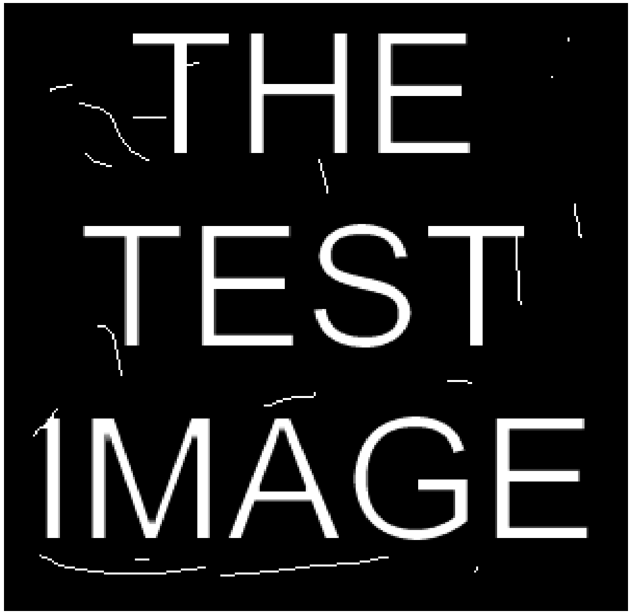
B



$A \ominus B$

Erosion

Original image



Eroded once

Eroded twice

Structuring
element: .

Dilation (+) and Erosion (-) properties

$A + \{\varnothing\} = A - \{\varnothing\} = A$

$A + \{a\} = A - \{a\}^r = A_a$, translation

$A + B = (A^c - B^r)^c$

$A - B = (A^c + B^r)^c$

$(A+B)^c = A^c - B^r$

$A+B_t = (A+B)_t$

$A-B_t = (A-B)_{-t}$

Erosion and Dilation Duality Theorem: Dilation and Erosion transformation bear a similarity, what one does to image foreground and the other does for the image background.

Similar but not identical to De Morgan rule in Boolean Algebra

Decomposition: $B = B_1 + B_2 + B_3 + \ldots + B_n$

$A+B = (\ldots(((A+B_1)+B_2)+B_3)+\ldots)+B_n$

$A-B = (\ldots(((A-B_1)-B_2)-B_3)-\ldots)-B_n$

Dilation (+) and Erosion (-) properties

$(A+B)+C=A+(B+C)$ $(A-B)-C=A-(B+C)$

$(A\cup B)+C=(A+C)\cup(B+C)$ $(A \cap B)-C=(A-C) \cap (B-C)$

$A+(B\cup C)=(A+B)\cup(A+C)$ $A-(B\cup C)=(A-C)\cap(B-C)$

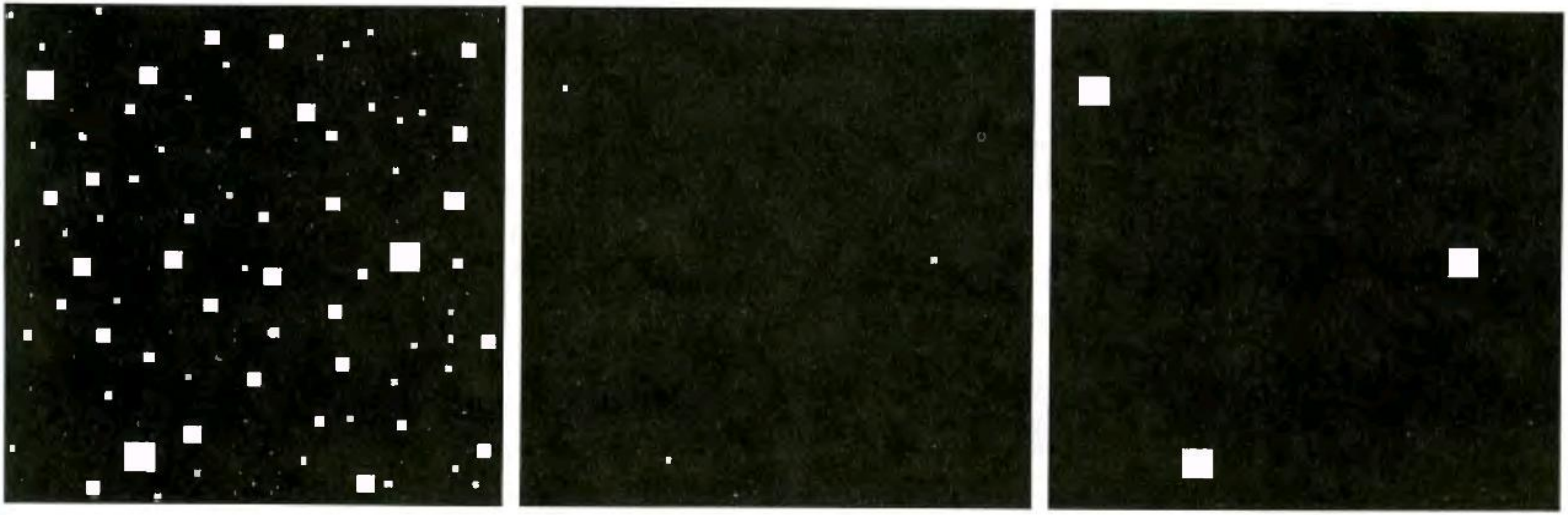
$A\subseteq B\Rightarrow(A+C)\subseteq (B+C)$ $A\subseteq B\Rightarrow(A-C)\subseteq (B-C)$

$B\subseteq C\Rightarrow(A-B) \supseteq (A-C)$

$(A\cap B)+C\subseteq (A+C)\cap(B+C)$ $(A\cup B)-C\supseteq (A-C) \cup(B-C)$

$A-(B\cap C)\supseteq(A-C)\cup(B-C$

Erosion and Dilation summary



a b c

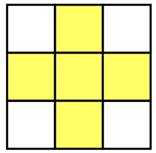
FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Structuring element: ■

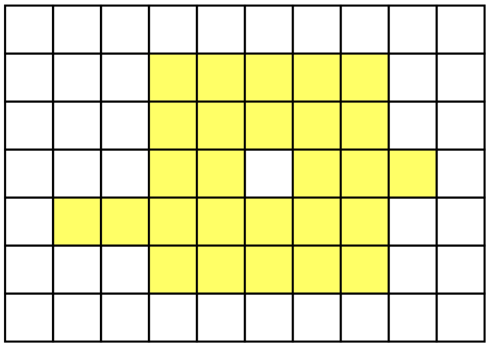
Closing operator

- $C(A, K) = (A \oplus K) \ominus K$
- Operator idempotent (the reapplication has not further effects): $A \subseteq C(A,K) = C(C(A,K),K)$

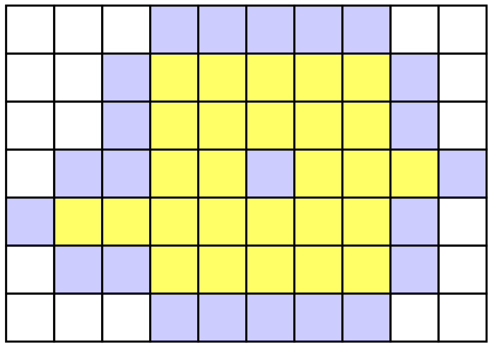
Structuring element:



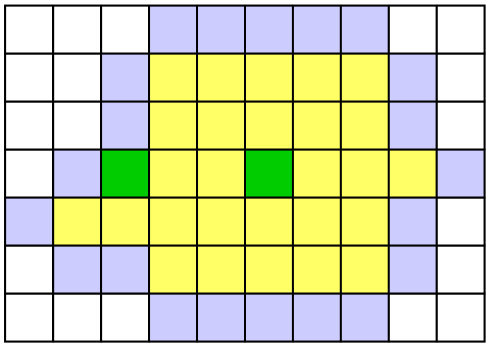
K



A



$A \oplus K$

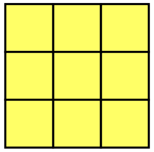


$(A \oplus K) \ominus K$

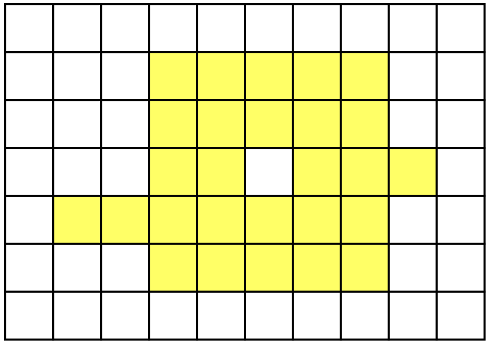
Closing operator

- $C(A, K) = (A \oplus K) \ominus K$

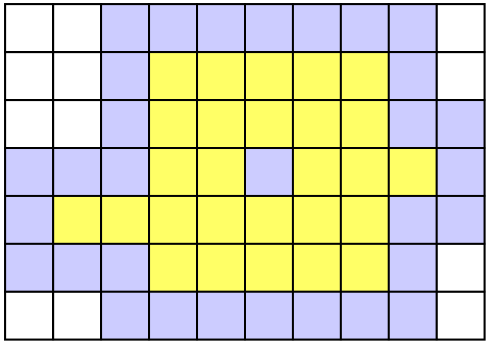
Structuring element:



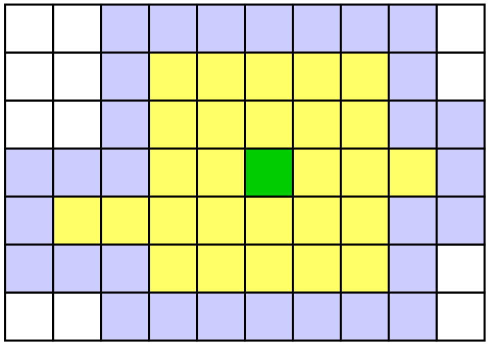
K



A



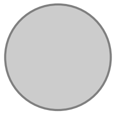
$A \oplus K$

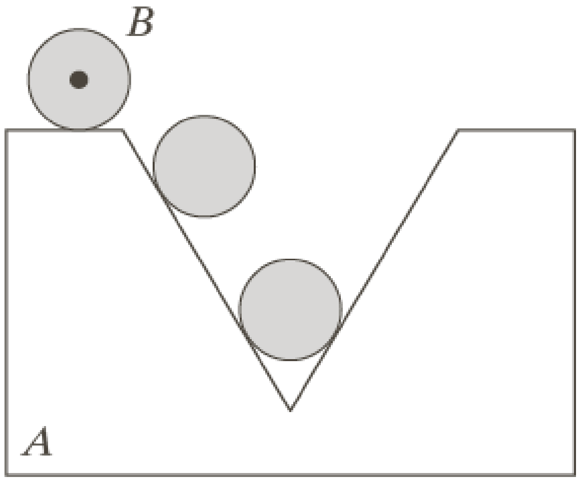


$(A \oplus K) \ominus K$

Closing

Structuring element:



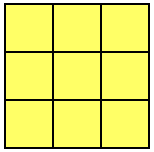


Gonzales-Woods

Closing operator

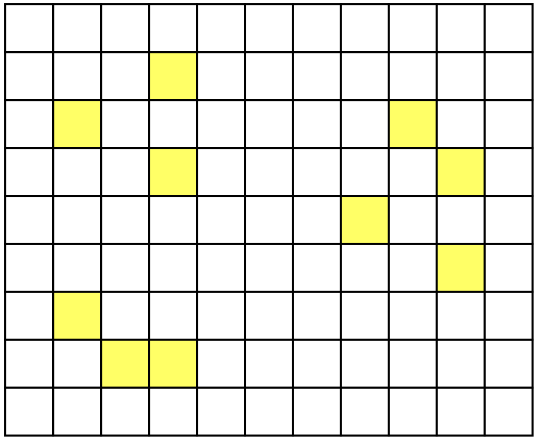
- $C(A, K) = (A \oplus K) \ominus K$

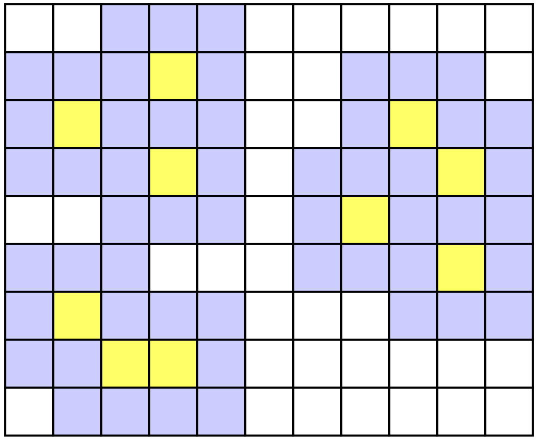
Structuring element:



K

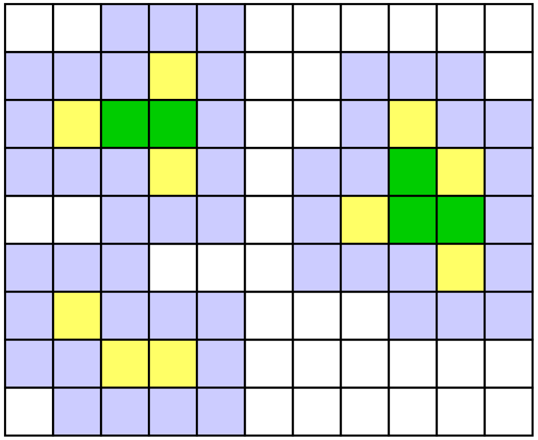
A





$A \oplus K$

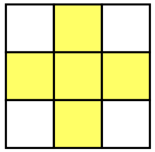
$(A \oplus K) \ominus K$



Opening operator

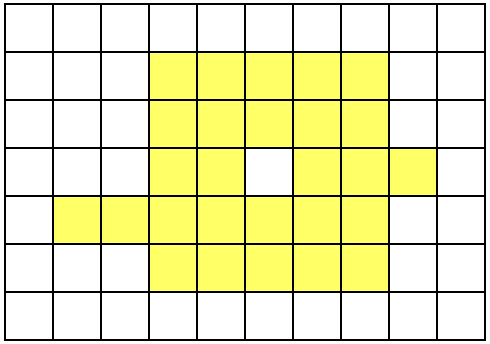
- $O(A, K) = (A \ominus K) \oplus K$
- Operator idempotent (the reapplication has not further effects): $O(O(A,K),K)=O(A,K) \subseteq A$

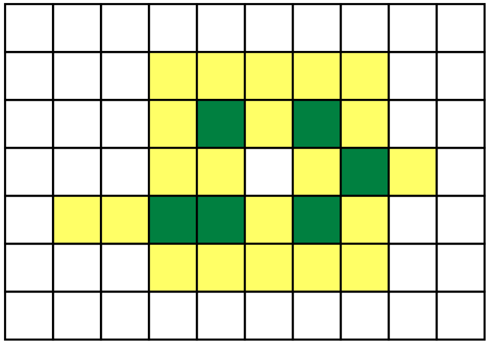
Structuring element:



K

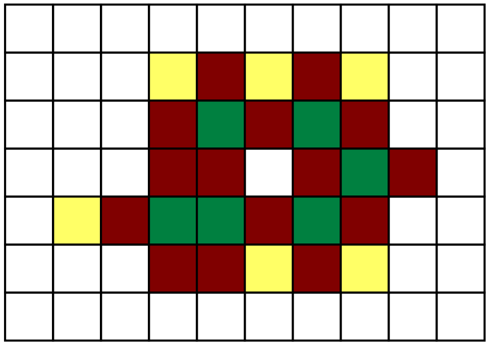
A





$A \ominus K$

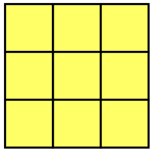
$(A \ominus K) \oplus K$



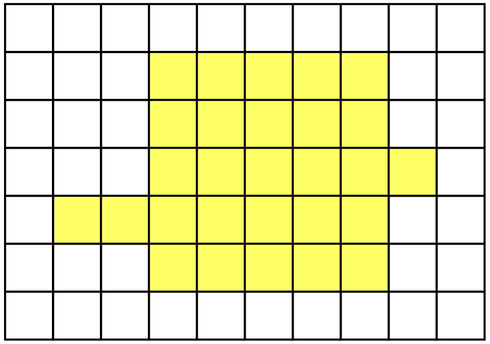
Opening operator

- $O(A,K) = (A \ominus K) \oplus K$

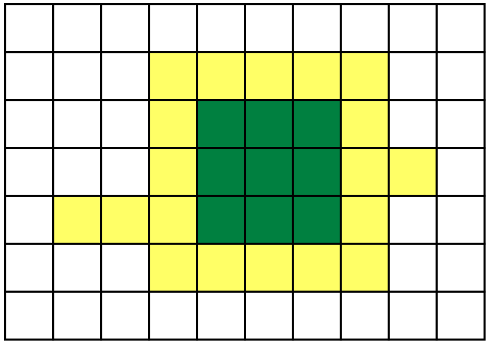
Structuring element:



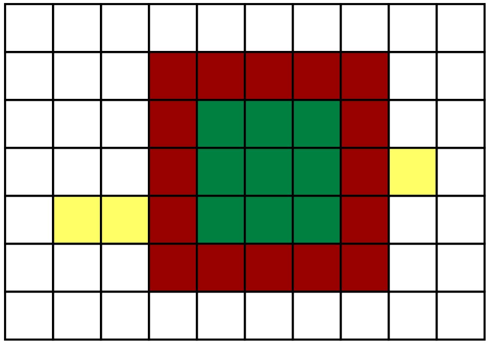
K



A

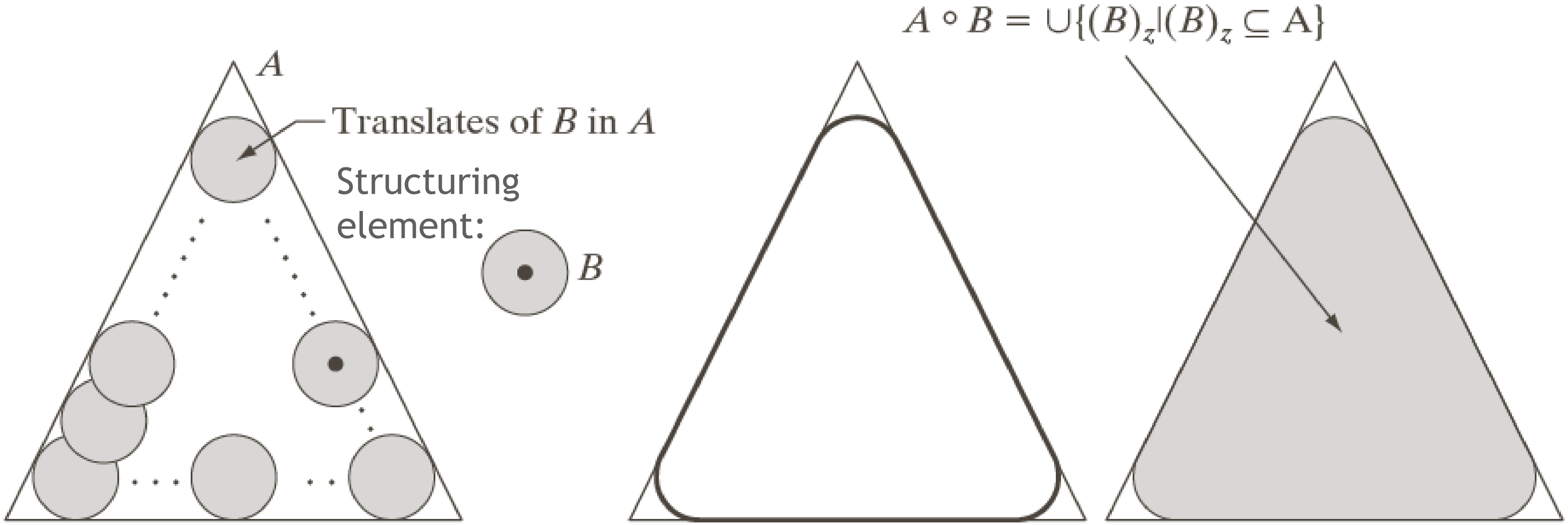


$A \ominus K$



$(A \ominus K) \oplus K$

Opening

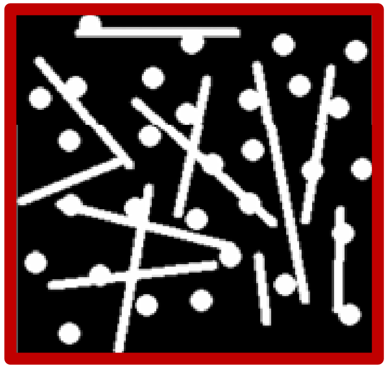


Gonzales-Woods

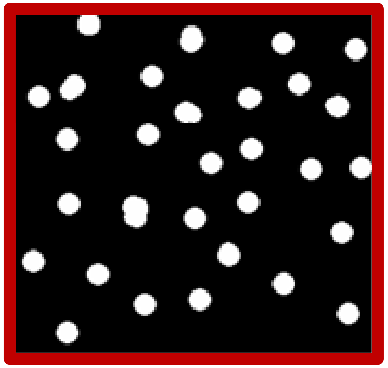
Opening

- Erode, then dilate
- Remove small objects, keep original shape

Structuring element: •



Before opening

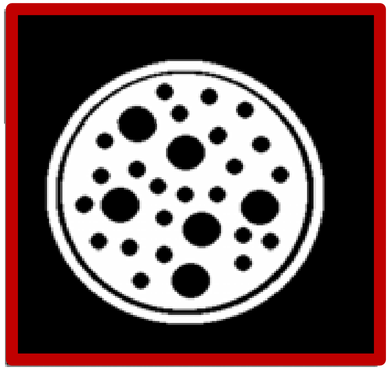


After opening

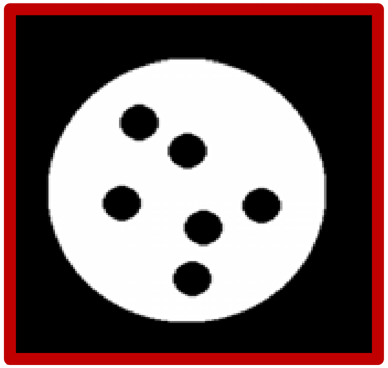
Opening

- Erode, then dilate
- Fill holes, but keep original shape

Structuring element: •



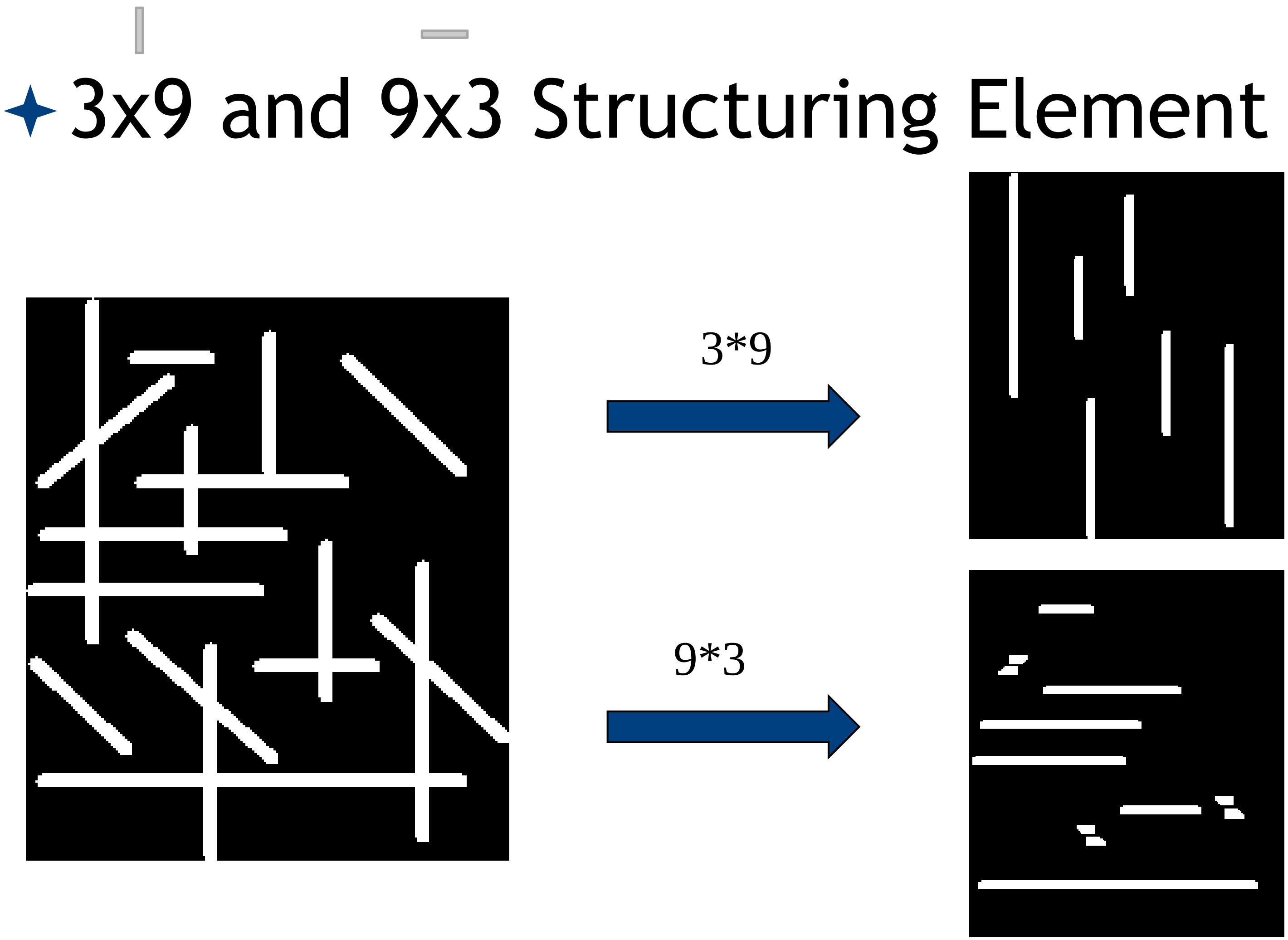
Before Opening



After Opening

Opening Example

- 3x9 and 9x3 Structuring Element

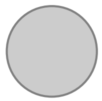


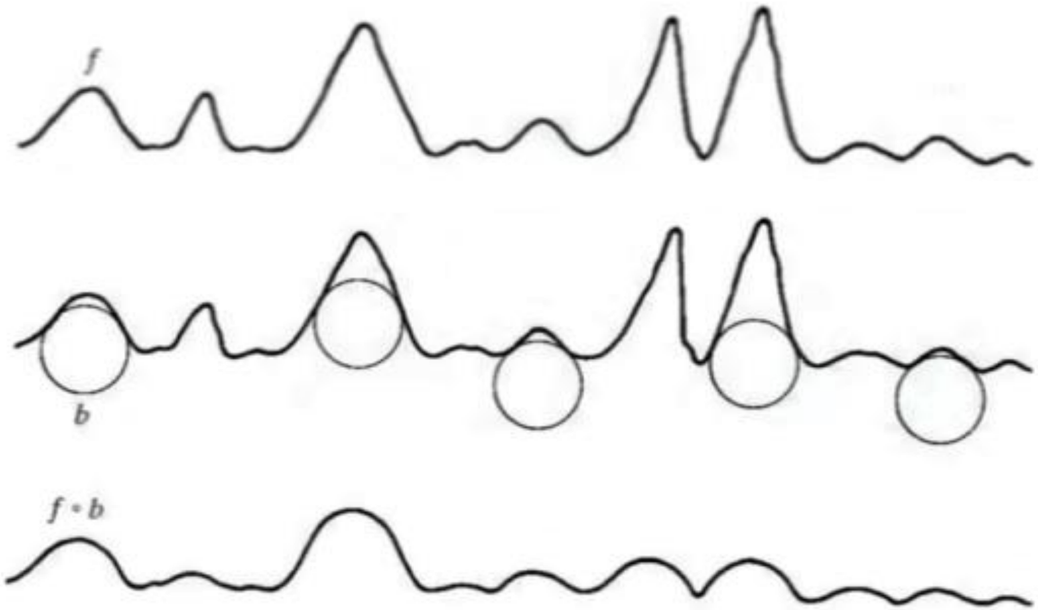
Opening and Closing: contour curvature

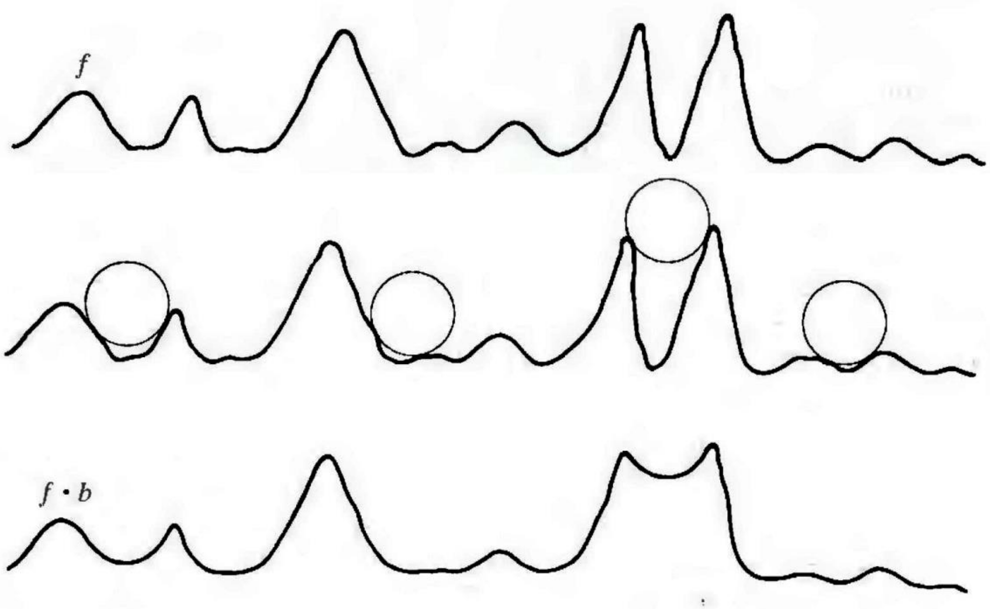
- **Opening** a picture is describable as pushing object B under the scan-line graph, while traversing the graph according the curvature of B
- The valleys usually remains in their original form

- **Closing** a picture is describable as pushing object B on top of the scan-line graph, while traversing the graph according the curvature of B
- The peaks usually remains in their original form

Structuring element:

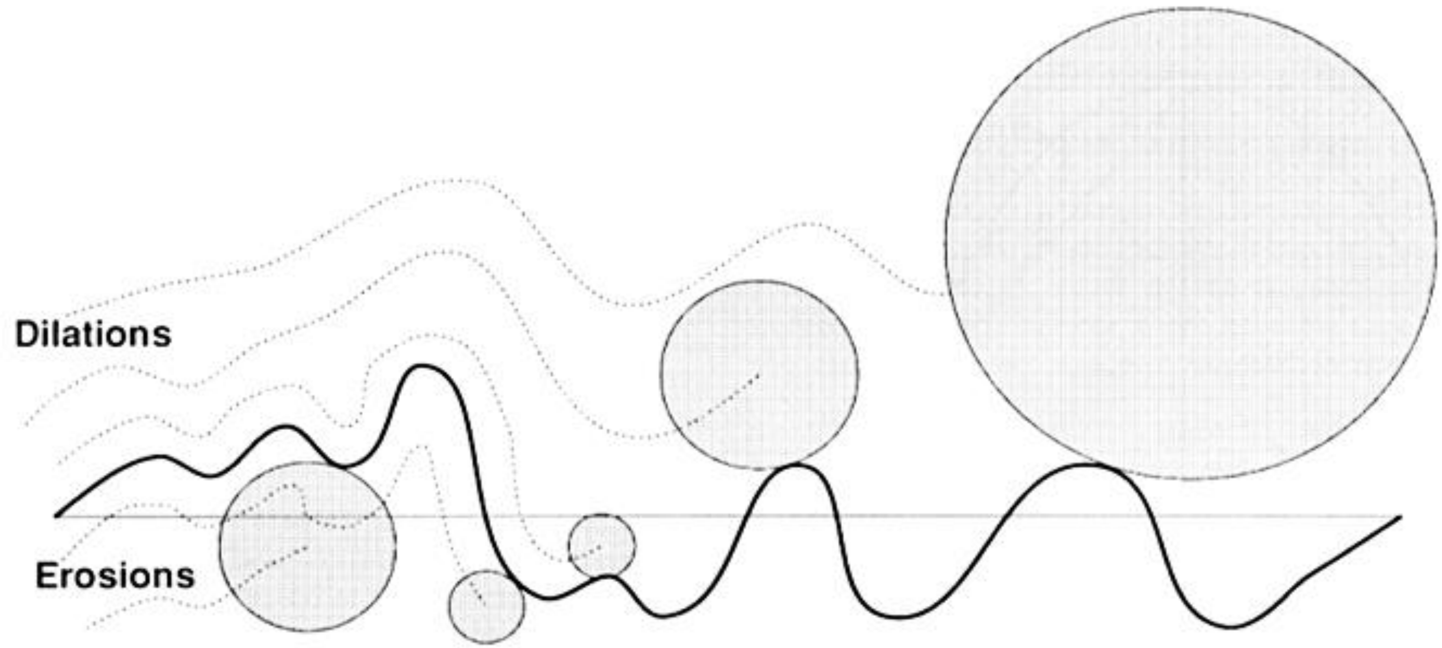






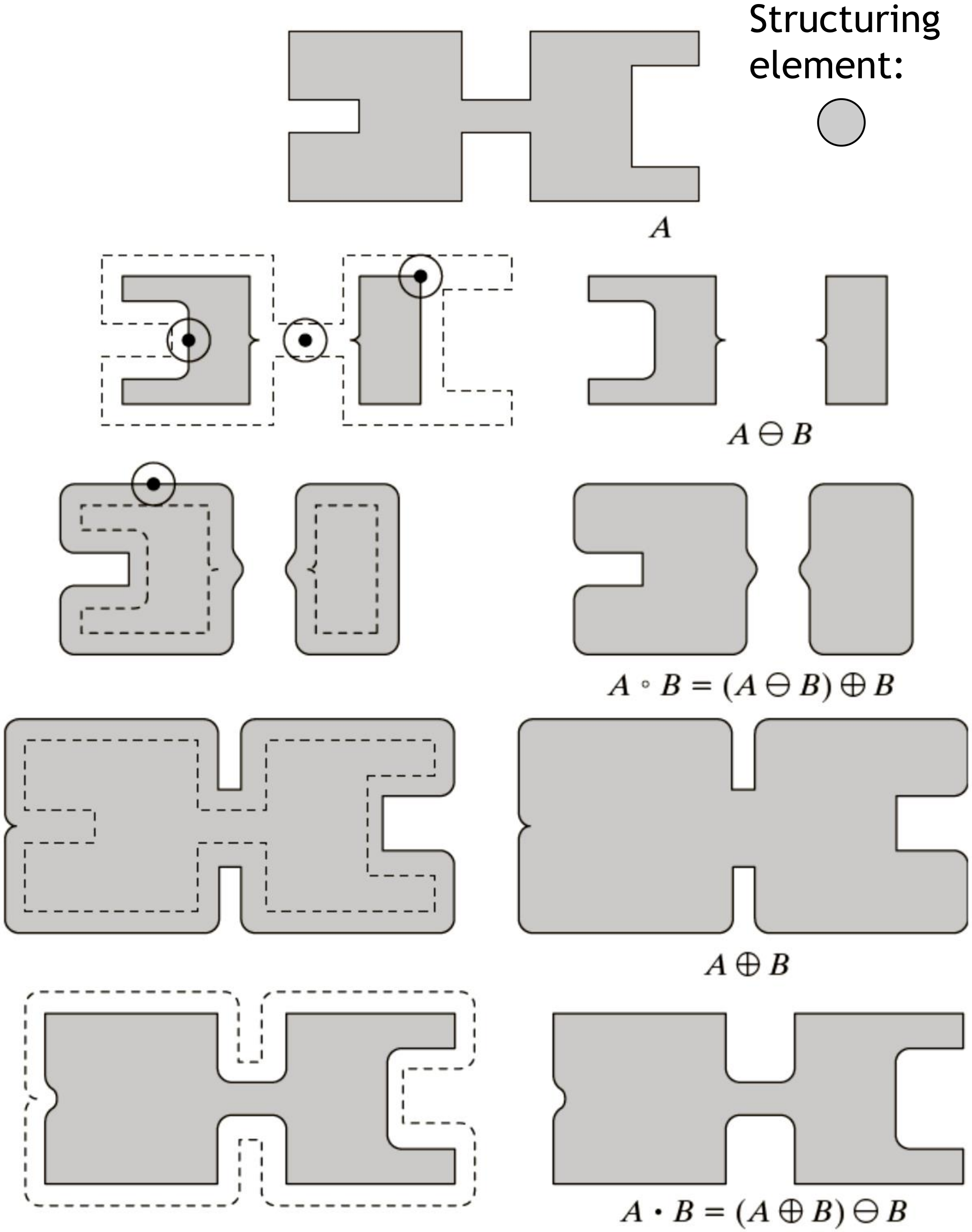
The ‘good’ contour

- Opening and Closing operator with a circle as structural elements change the boundaries as shown in figure: closing extends the boundary as if a ball rolles over the outer border; opening restricts it rolling the inner border
- The larger the circle the smoothed the result. The maximum resulting curvature is that of the structural element



Arjan Kuijper

Opening vs Closing



Gonzales-Woods

Opening
Closing

Gonzales-Woods

Hit or Miss operator

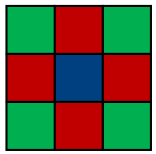
- $A \otimes (J,K) = (A \ominus J) \cap (A^c \ominus K)$
  - con il vincolo $J \cap K = \emptyset$
- Suitable for 'template' matching

Two structuring elements J and K

Hit or Miss

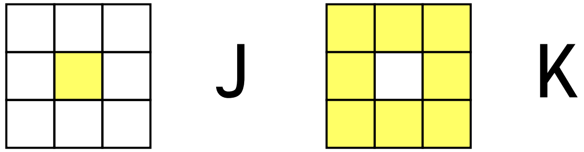
- J and K can be seen as a single template with three values:
  - Foreground points
  - Background points
  - Do not care points

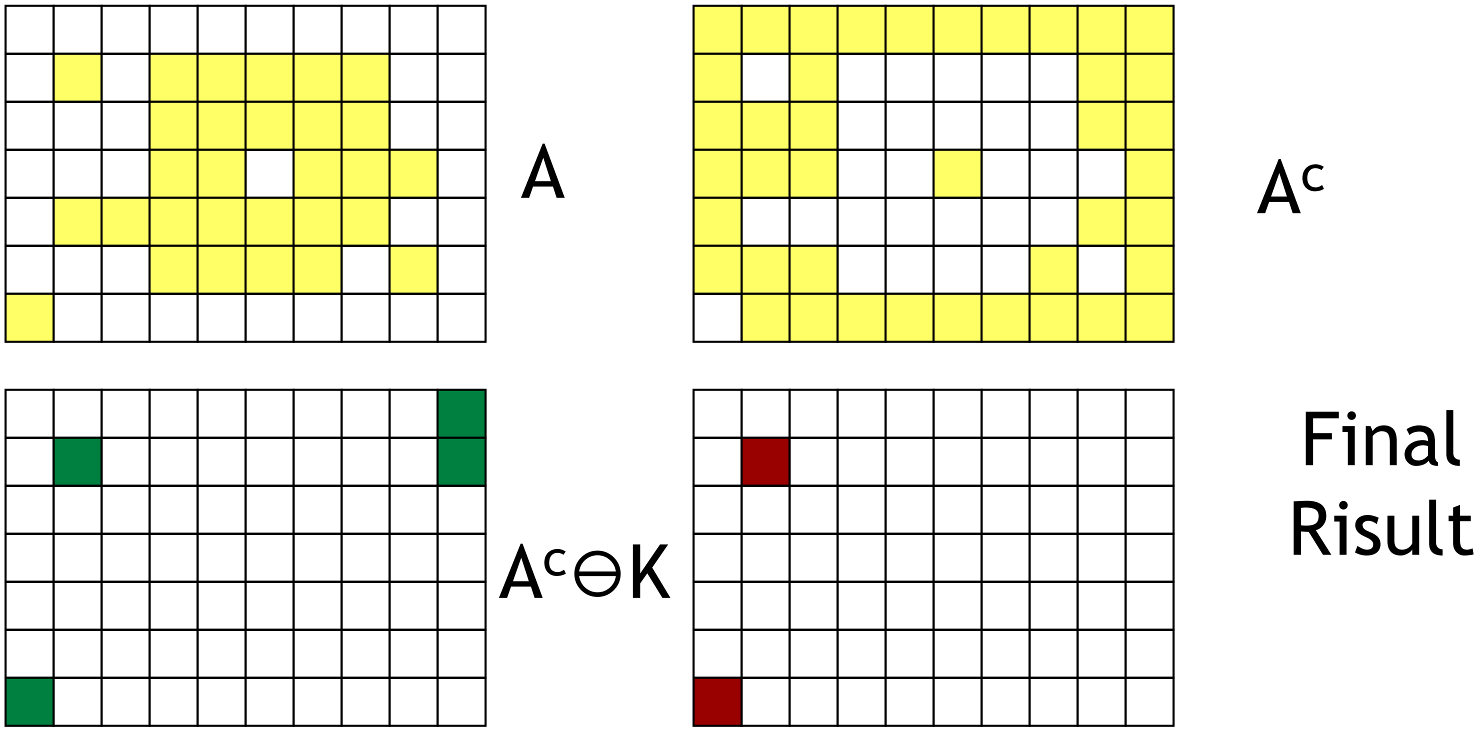
M



Hit or Miss

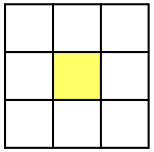
- Search of isolated points(8-connection)
- $A \ominus J = A$



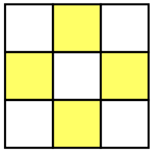


Hit or Miss

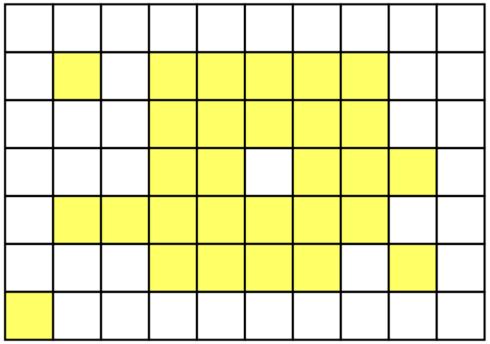
- Search of isolated points(4-connection)
- $A \ominus J = A$



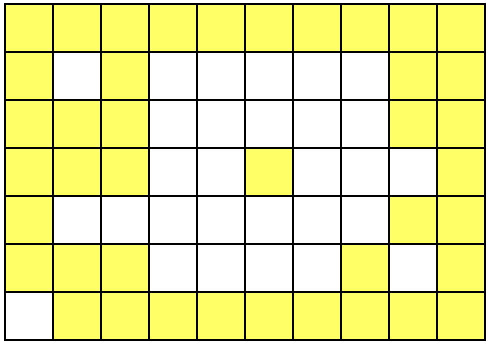
J



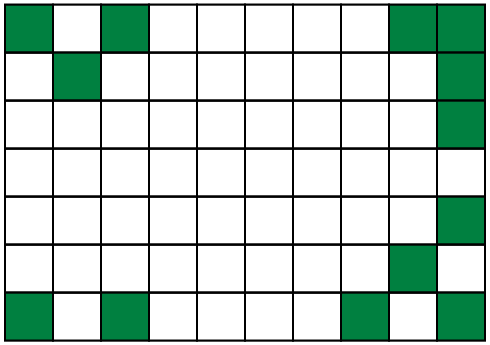
K



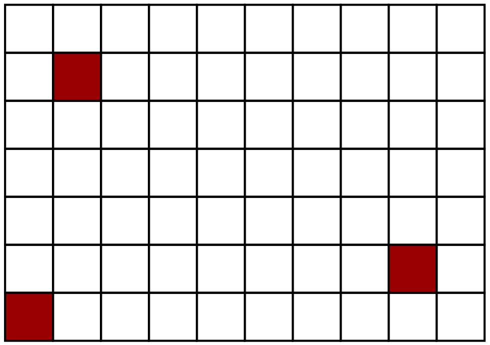
A



A^c

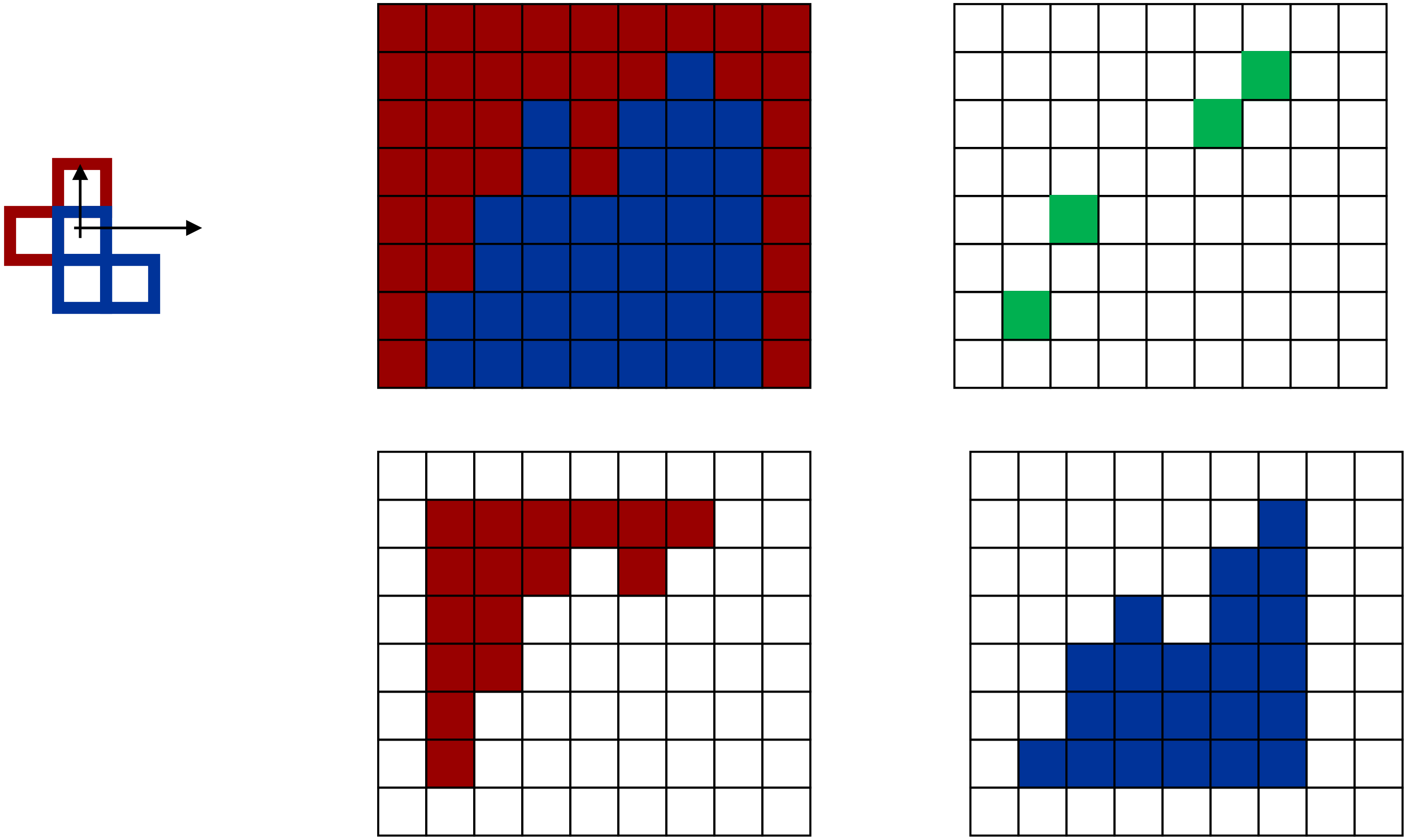


$A^c \ominus K$



Final Risult

Hit or Miss

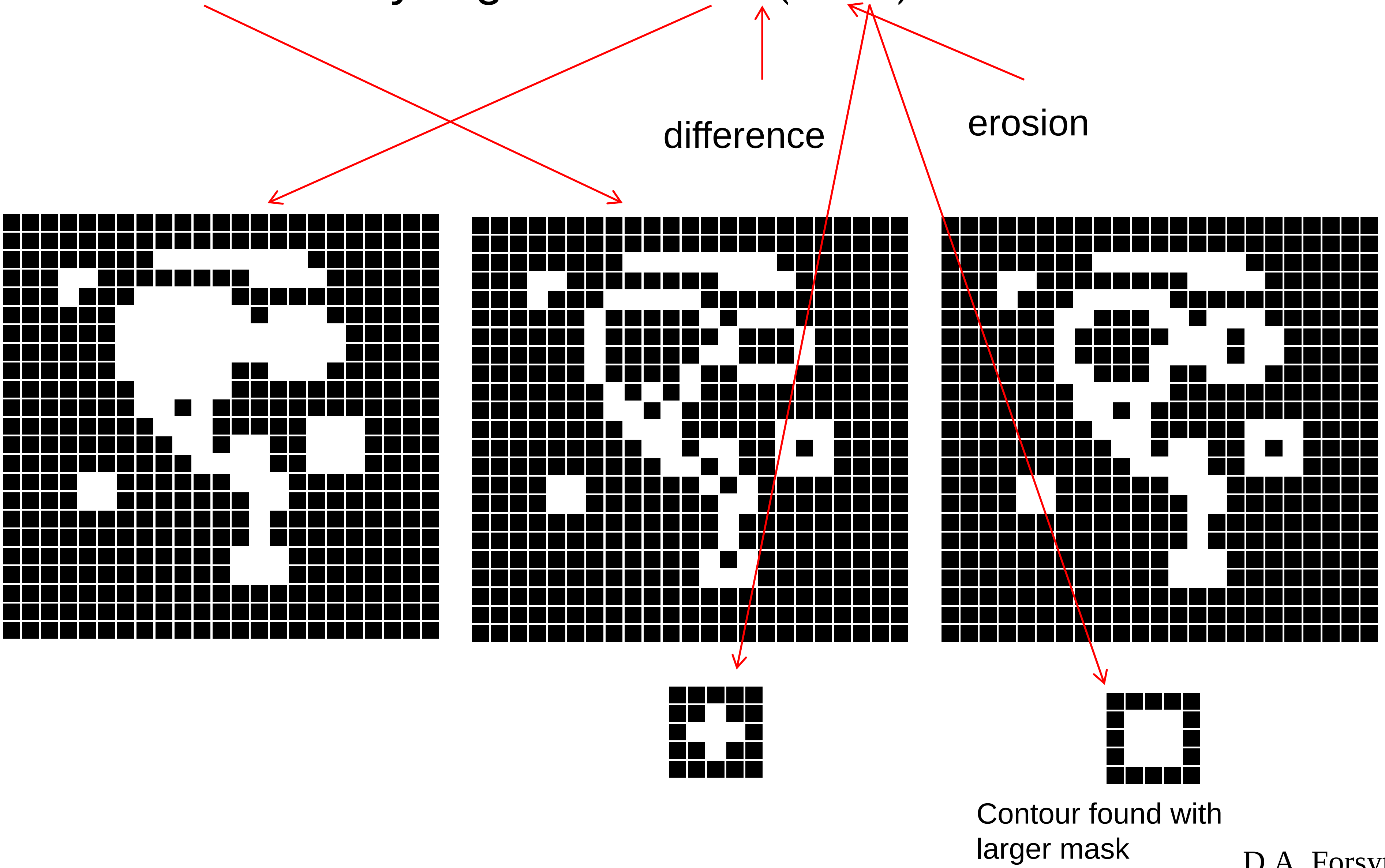


Pixels satisfying the background constraints

Pixels satisfying the foreground constraints

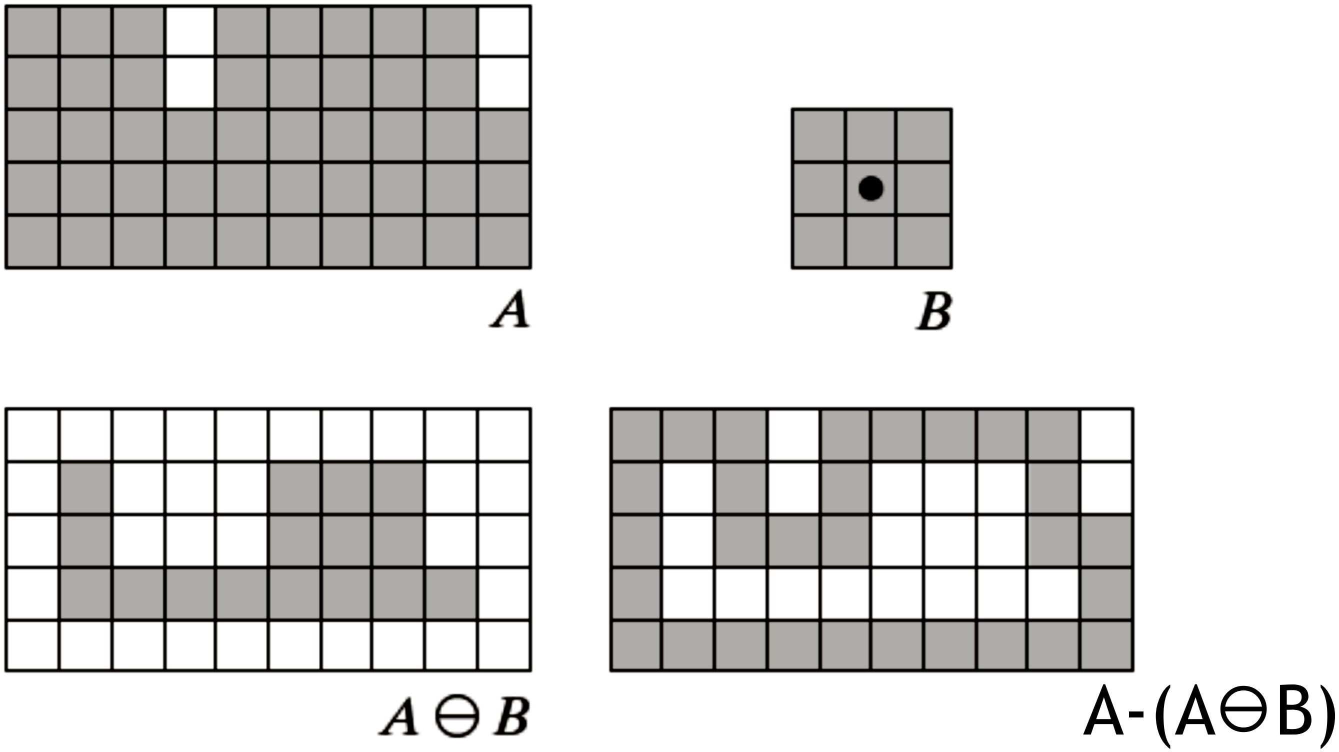
Using erosion to find contours

Contours of binary regions : A-(A⊖K)



D.A. Forsyth

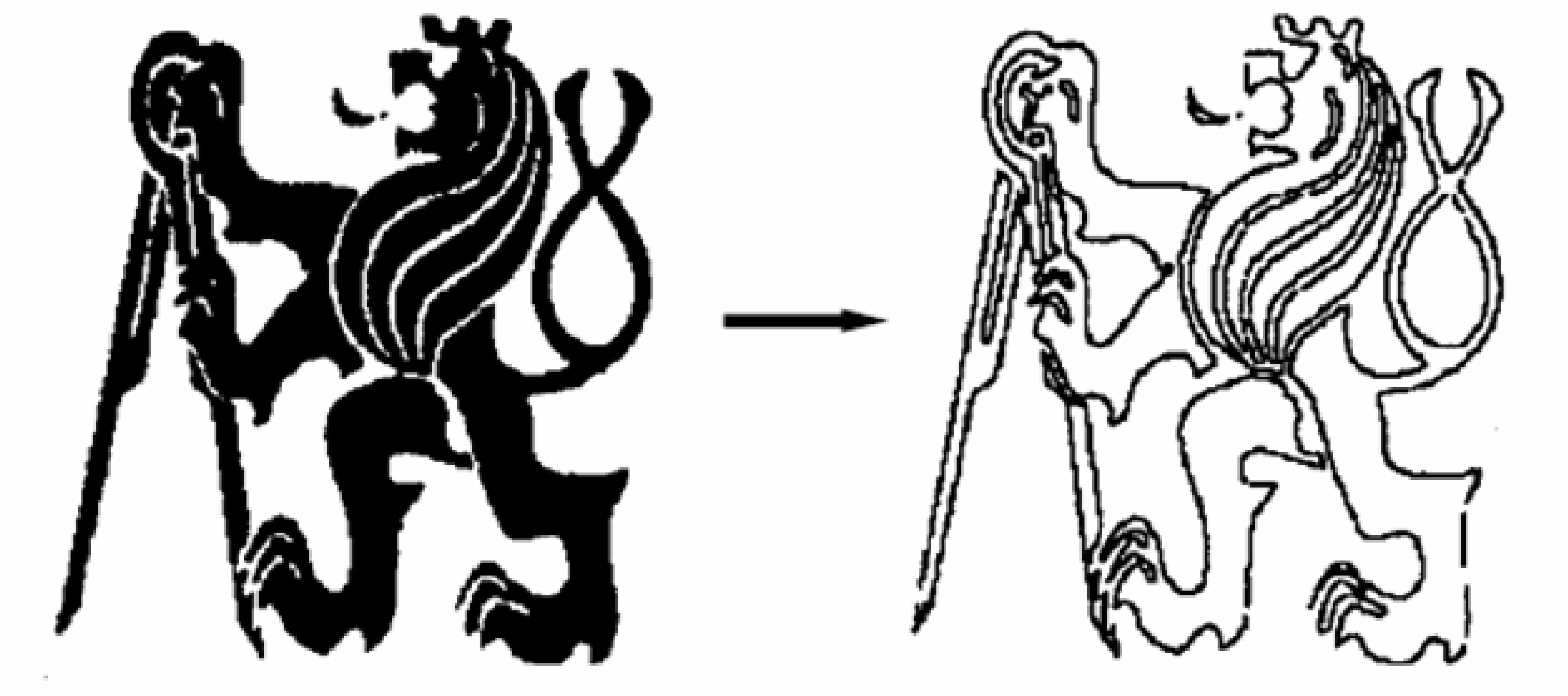
Contour example



Gonzales-Woods

Examples: Boundary Extraction

- Contour
  - Internal: $A-(A\ominus K)$
  - External: $(A \oplus K)\cap\bar{A}$ or $(A \oplus K)-A$
  - Double: $(A \oplus K)\cap\overline{(A \ominus K)} = (A \oplus K)-(A\ominus K)$



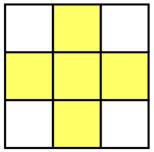
Example 2

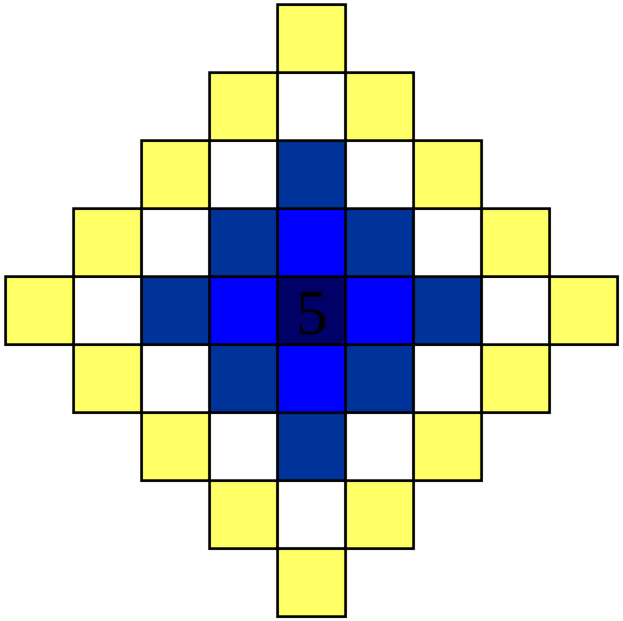
$A+A^c \ominus K$

Gonzales-Woods

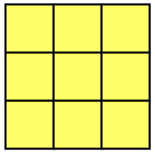
Iteration: disks in 4 and 8 connectivity

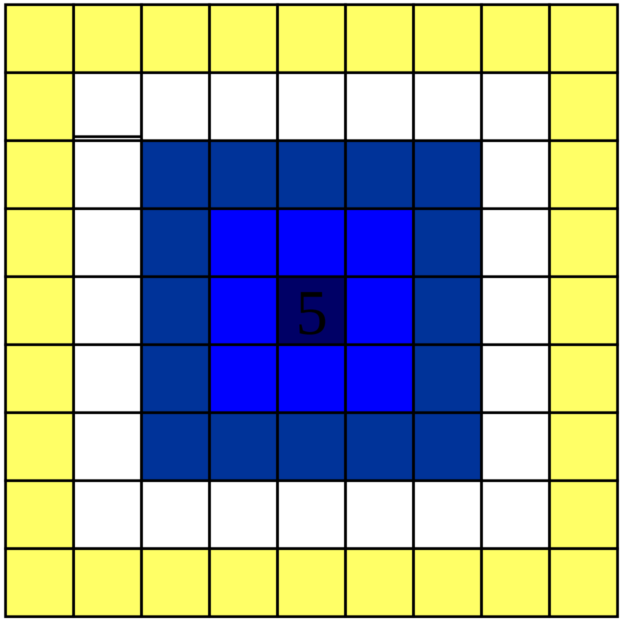
Structuring element:





Structuring element:





$X = \{ C \}$; I=1

for i=1,R do $X=(X \oplus K)$

C = center pixel

X = evolving image

R = radius (4 in ex.)

Recursion: Propagation

- Propagation in a connected component
  - Let A be a set containing one or more connected components (mask), and consider an array X_0 (of the same size of the array A) whose elements are 0s, except to a point of A foreground (marker).

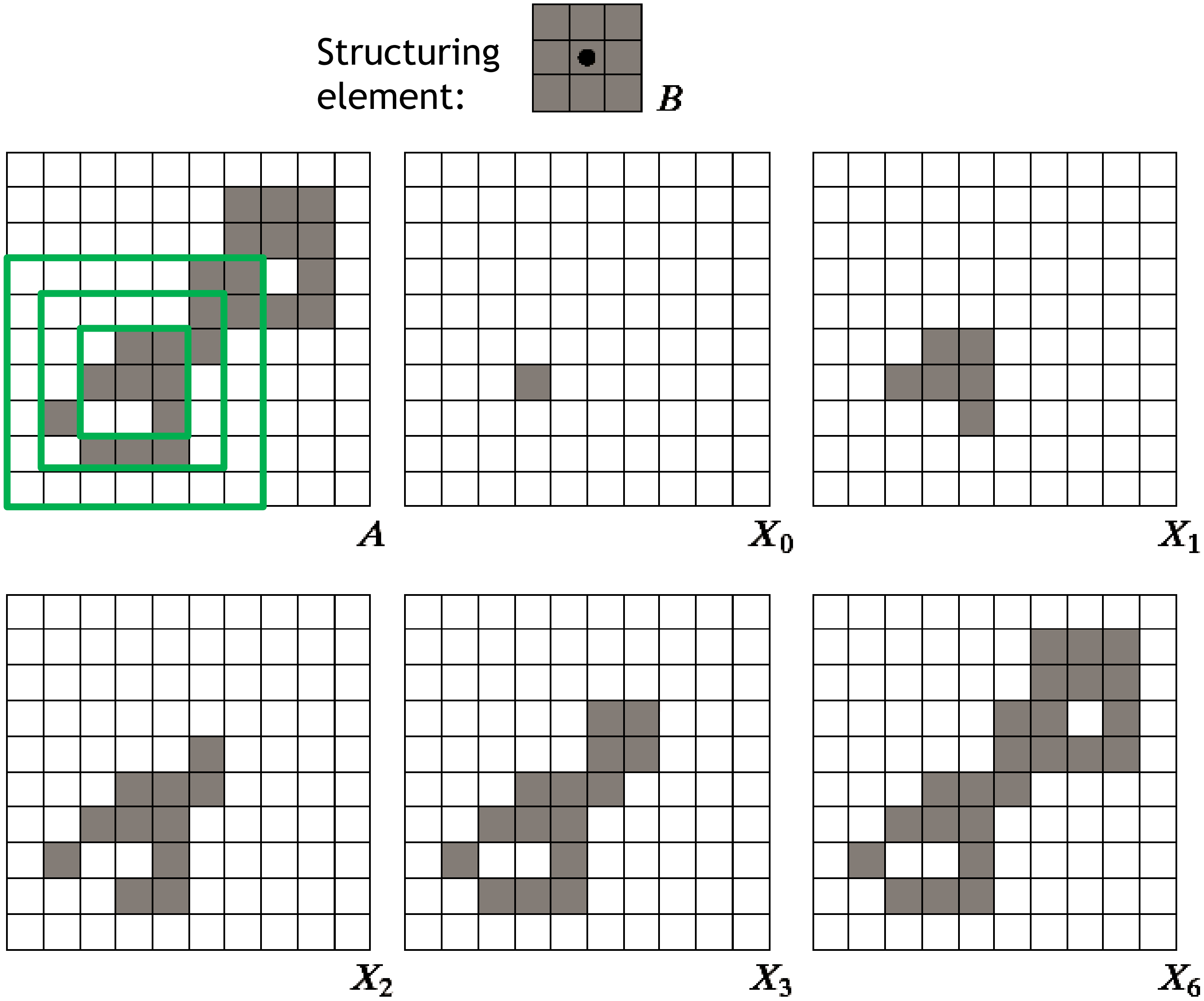
  - $X = \{ X_0 \};$
  - do $D = X$
    - $X = (X \oplus K) \cap A$
    - while($D \neq X$)

X = evolving image

A = original image

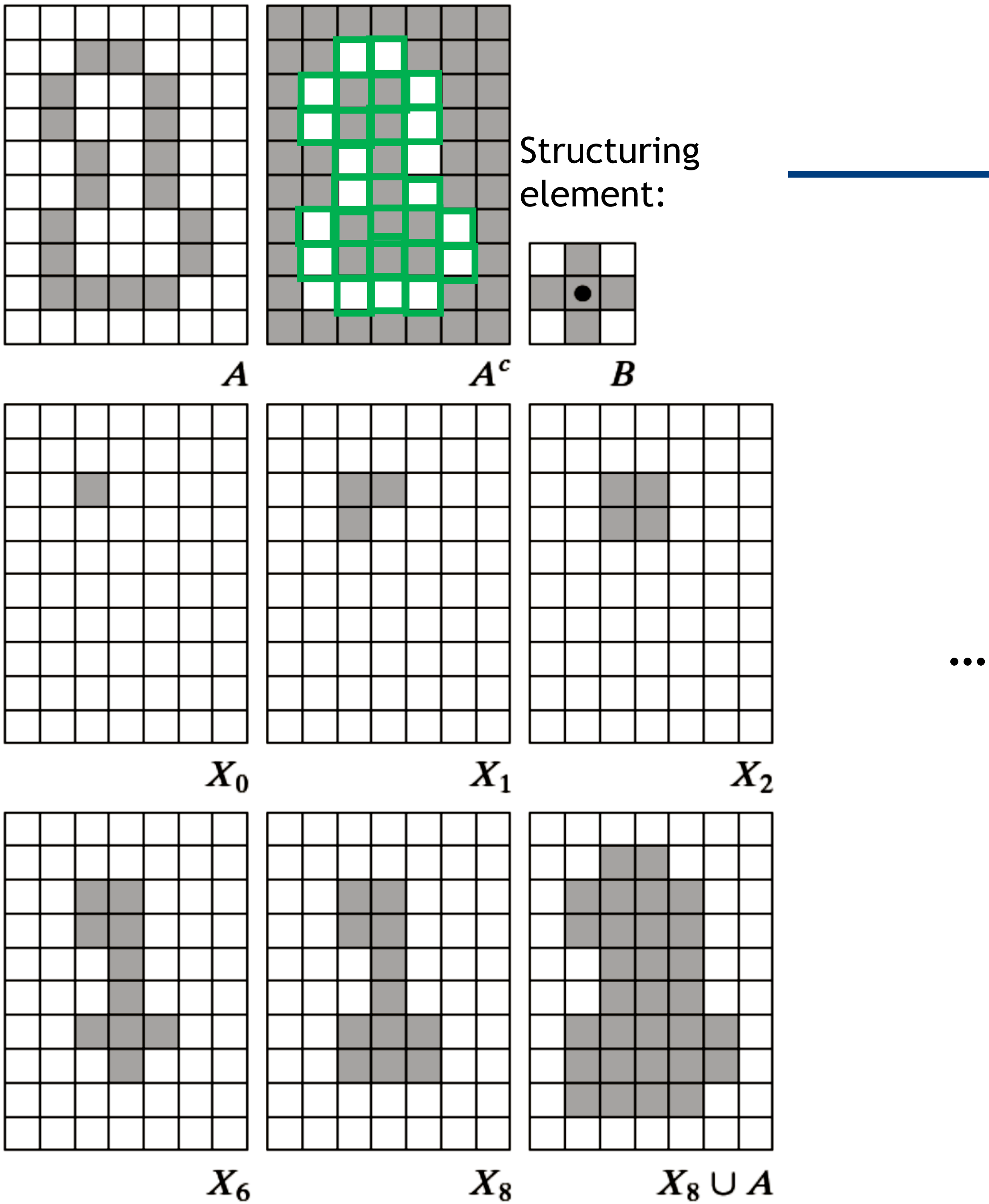
K is the unitary circle in the adopted metric

Example: Connected Components



Gonzales-Woods

Hole Filling

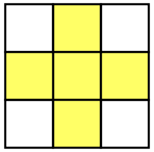


Gonzales-Woods

DT - algorithm

			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		

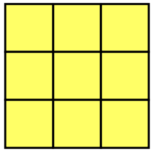
Structuring element:



			1	1	1	1	1		
		1	2	2	2	2	2	1	
		1	2	3	3	3	2	1	
	1	2	3	4	4	4	3	2	1
1	2	2	3	3	3	3	2	1	
	1	1	2	2	2	2	2	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		

Structuring element:

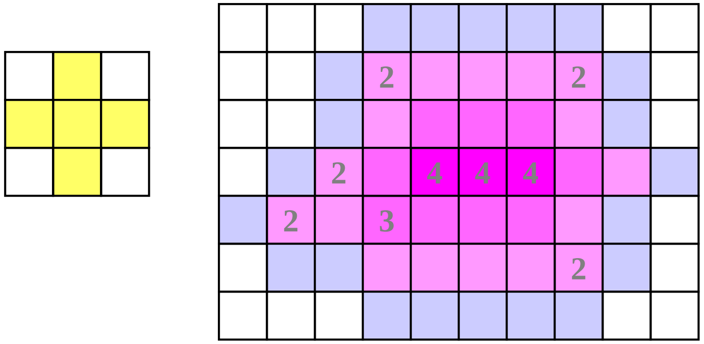


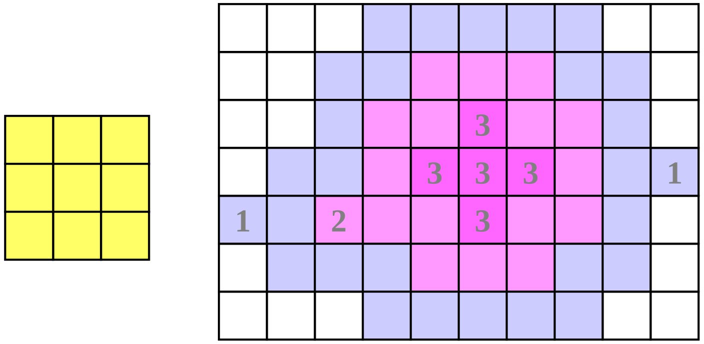
			1	1	1	1	1		
		1	1	2	2	2	1	1	
		1	2	2	3	2	2	1	
	1	1	2	3	3	3	2	1	1
1	1	2	2	2	3	2	2	1	
	1	1	1	2	2	2	1	1	
			1	1	1	1	1		

Distance transform

- DT implementation using dilation and addition operators:
  - $R = \emptyset$ R = evolving image
  - while($A \neq \emptyset$) do at the end DT
    - $R = R + A$
    - $A = A \ominus K$
  - done

DT – local maxima

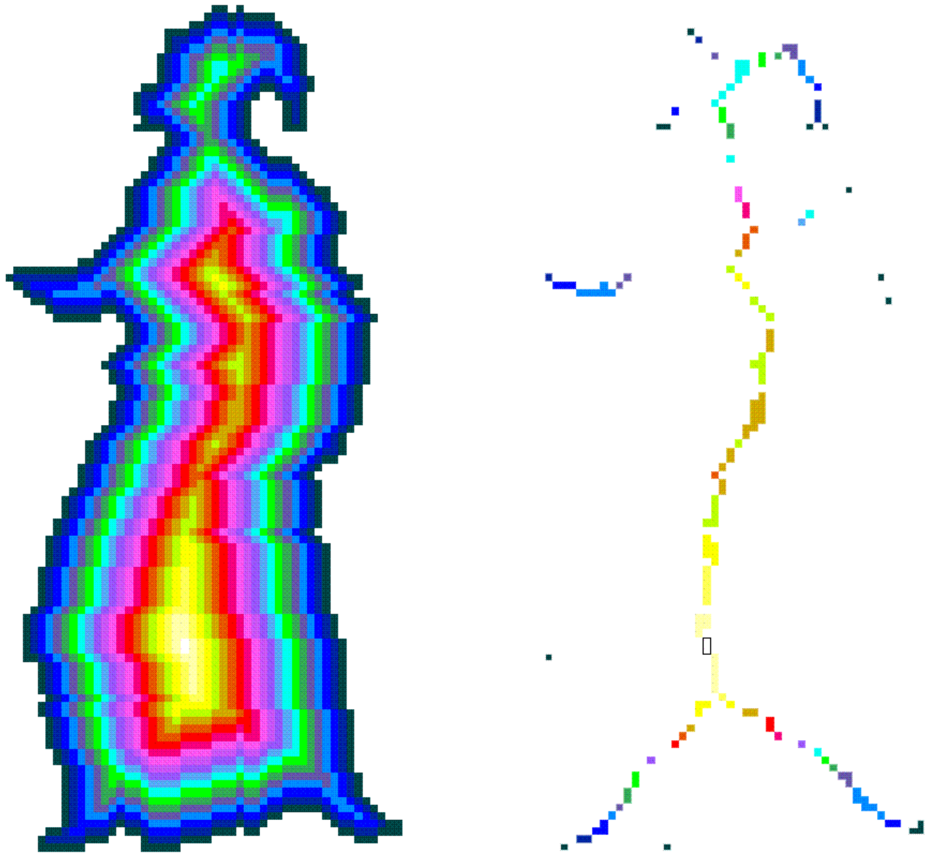




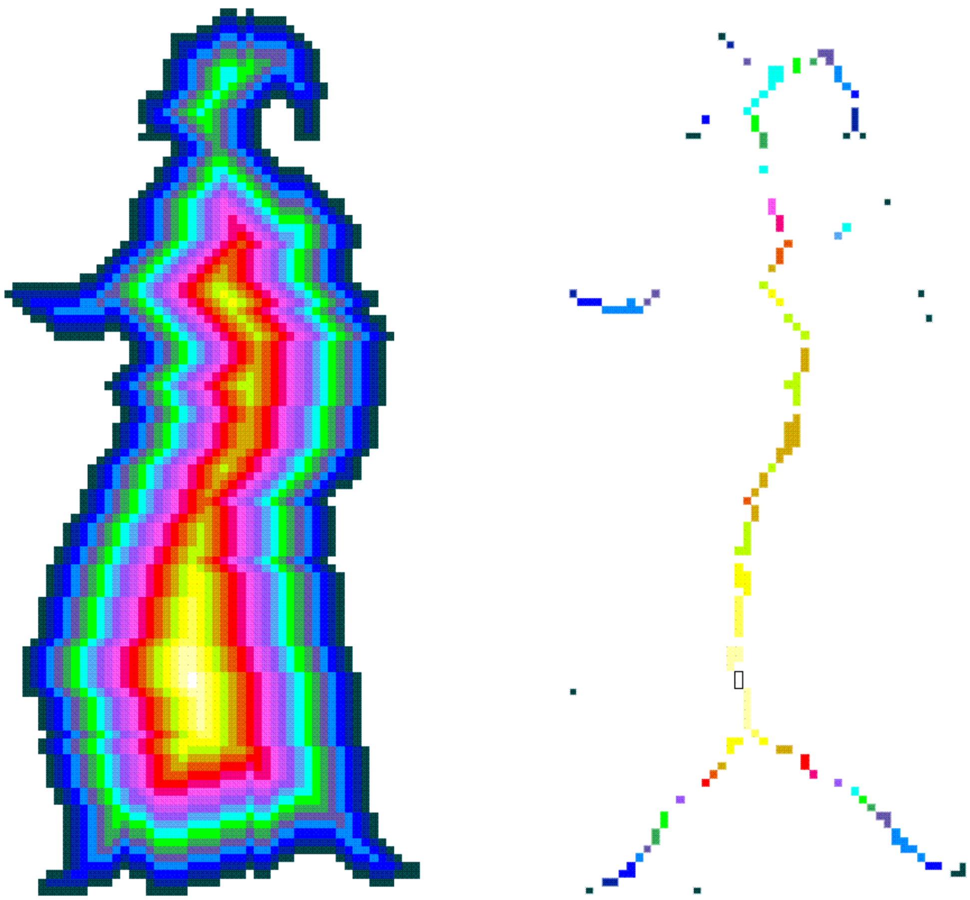
- The local maxima set is a compact object representation
- The object can be rebuilt as union of the maximal digital disks

Distance transform and MAT

- The Distance Transform (DT) is obtained by labeling all the pixels inside a binary object with their distance to the background
- Applying twenty iterations of the erosion operator (structural element: unit disk) twenty successive colored layers showing equi-distant contours from the background for a Manhattan distance metric are obtained
- Every pixel has a color corresponding to its distance label which increases going inwards. In practice, this value represents the side of the greatest digital disk having its centre on this pixel, which is completely contained in the binary object.
- Any pattern can be interpreted as the union of all its maximal digital disks (local maximum in DT). A maximal disk is a disk contained in the object that is not completely overlapped by any other disk.
- The set of the centers of the maximal disks with their labels, constitutes the MAT



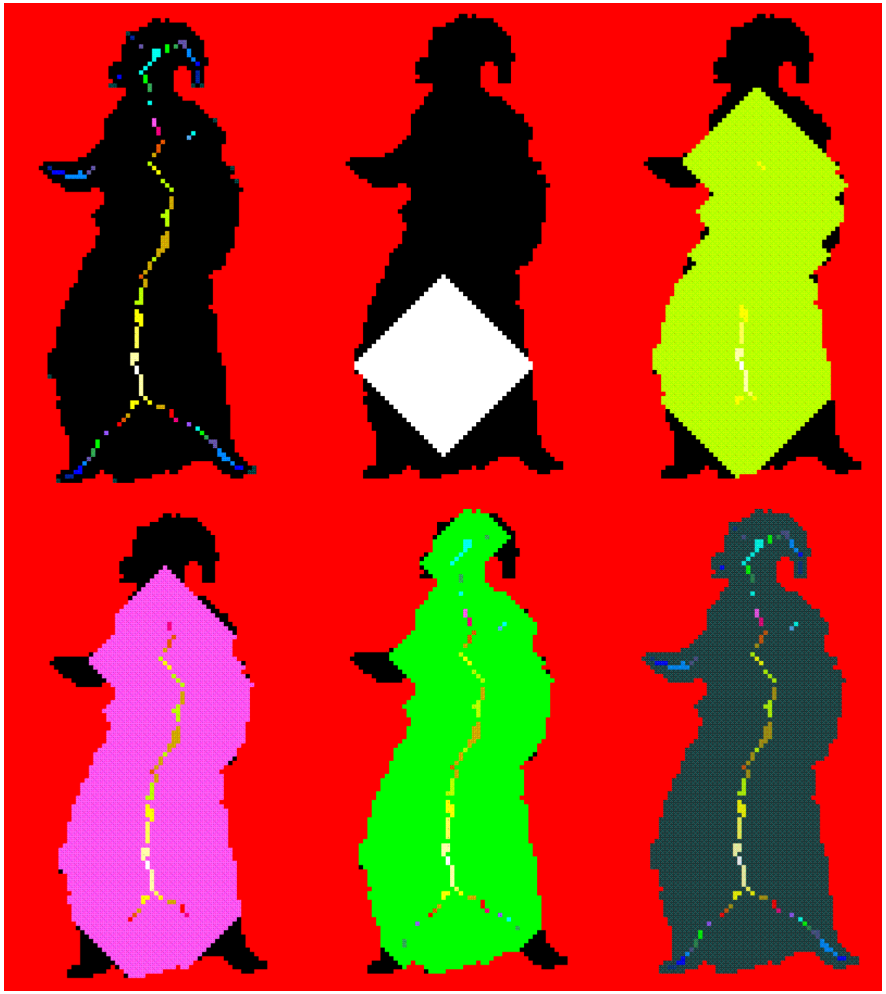
Distance transform and MAT



Reverting progressively MAT

- A procedure to derive the MAT from the DT is based on the comparison of neighboring labels to establish whether a local maximum exists
- This transform is complete in the sense that it is possible to revert it, so obtaining the original object back
- This recovery process can be implemented by expanding every pixel belonging to the MAT, using the corresponding maximal disc whose size is given by the pixel label. The logical union of all such discs reconstructs the original object
- This figure shows the progressive reconstruction, starting from the set of disks corresponding to the highest level (two white disks) until the sixth and last monk's profile, where discs, reduced to just one pixel, have been included
- This transform is compact since the full object may be described only by its labeled disk centers

$$U = \{\emptyset\}$$
$$\forall i,j : MAT_{i,j} > 0 \;\; \rightarrow U = U \cup D_{MAT_{i,j}}$$



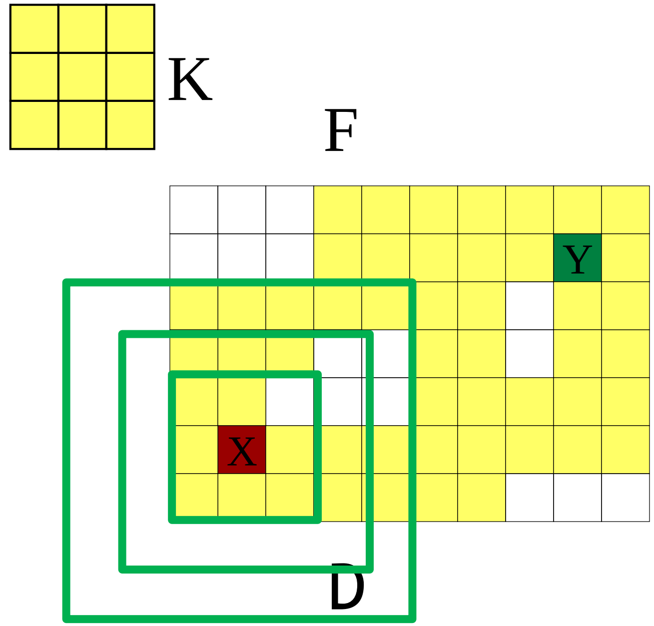
Distance between two points

- Distance between X,Y∈Z:
  - A = { X }; D=A Z={∅} A= evolving binary image
  - while(Y∉A) do
    - Z = A
    - A = (A ⊕ K)∩F
    - D = D + A
  - Done
  - If A ≡(A ⊕ K)∩F and Y has not been already reached: Z is not connected and Y is not reachable from X
- Following a path of max gradient we can find one of the minimum paths between X, Y

F= original image (mask)

Z= connected component

Structuring element:



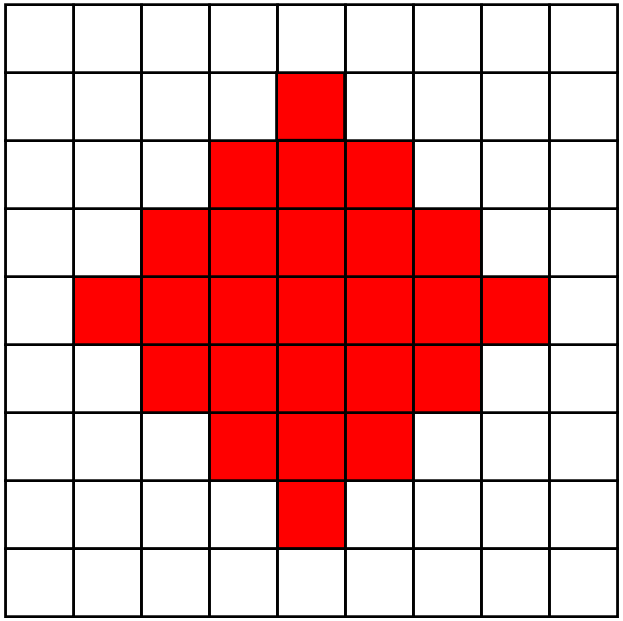
			4	4	4	3	2	1	
			5	5	4	3	2	1	
6	6	6	6	5	4	3		1	1
7	7	7			4	4		2	1
8	8				5	4	3	2	1
8	9	8	7	6	5	4	3	2	1
8	8	8	7	6	5	4			

Weighted DT

- In this case all neighbors are not considered at the same distance (e.g. 8-connectivity)
- Example: a good approximation to the Euclidean distance in 8-connectivity (the result is about doubled) is given by:

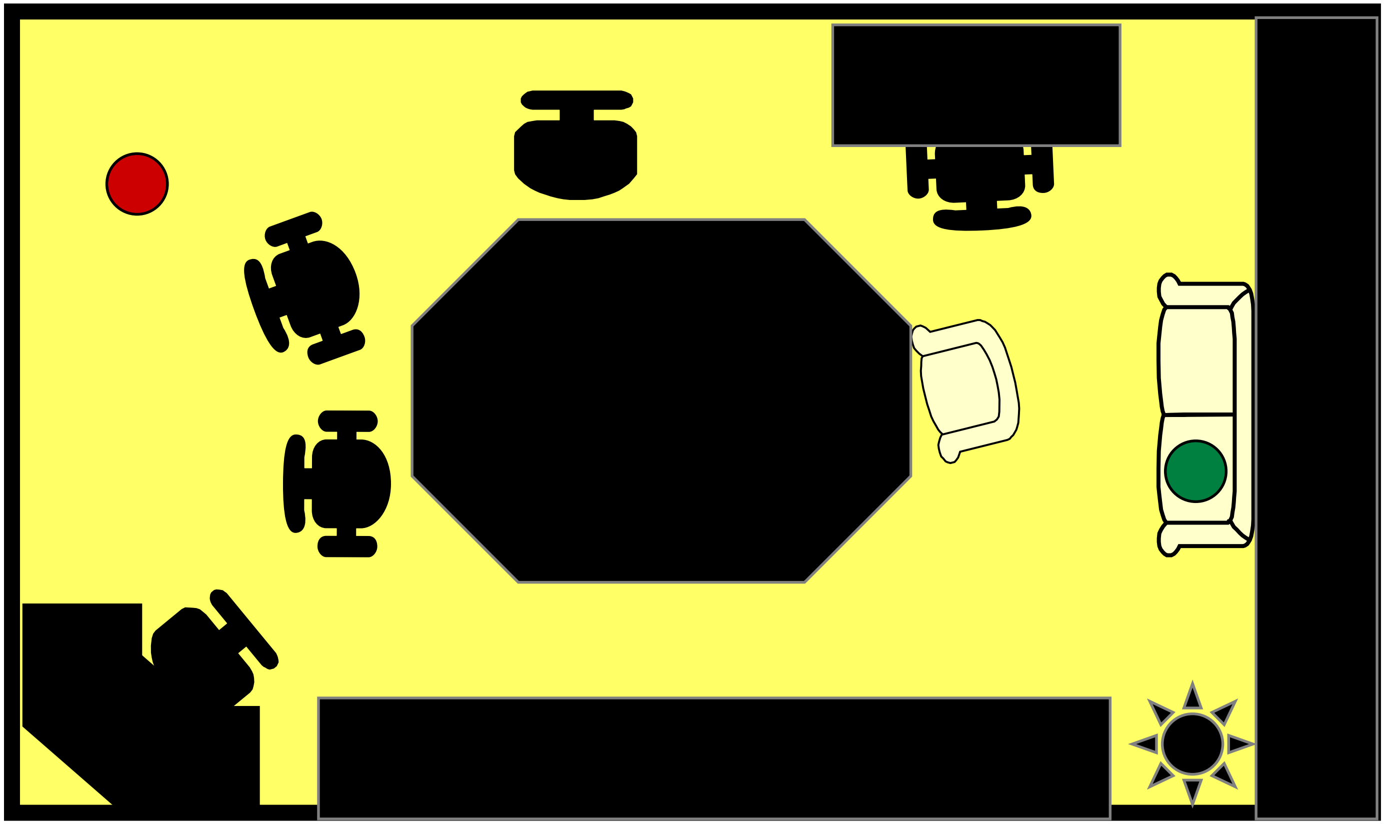
w=

3	2	3
2		2
3	2	3

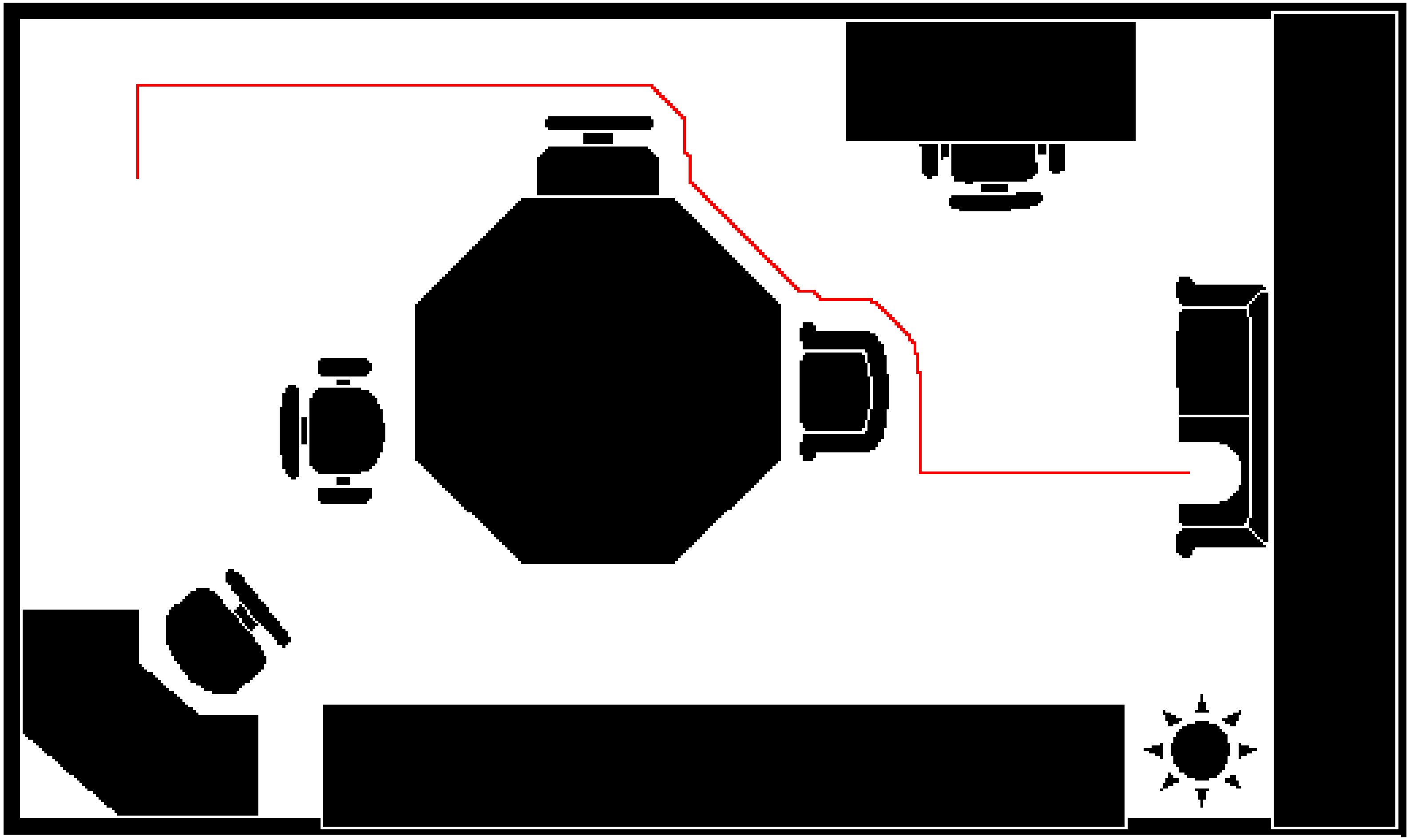


				2				
			2	3	2			
		2	3	5	3	2		
	2	3	5	6	5	3	2	
		2	3	5	3	2		
			2	3	2			
				2				

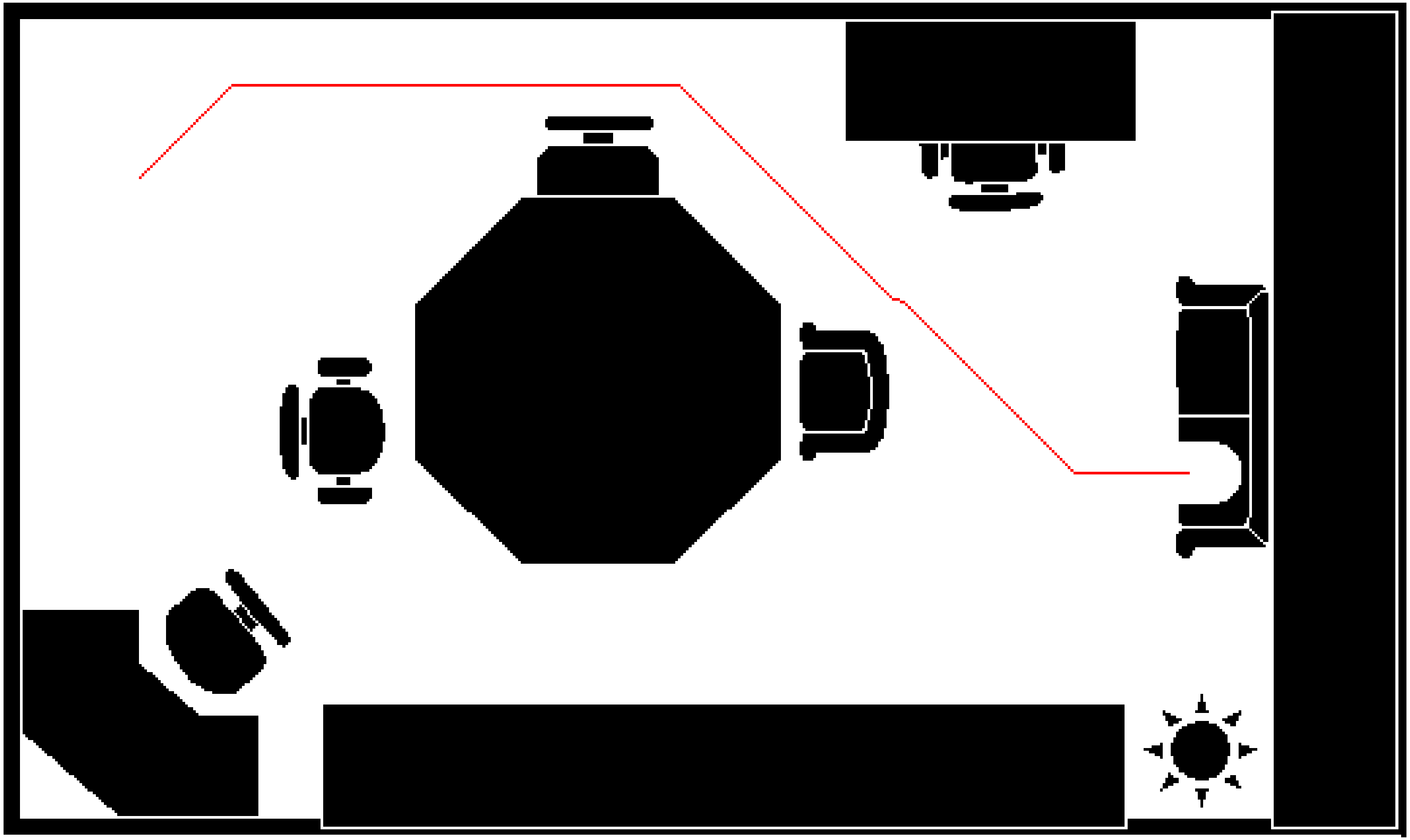
Example



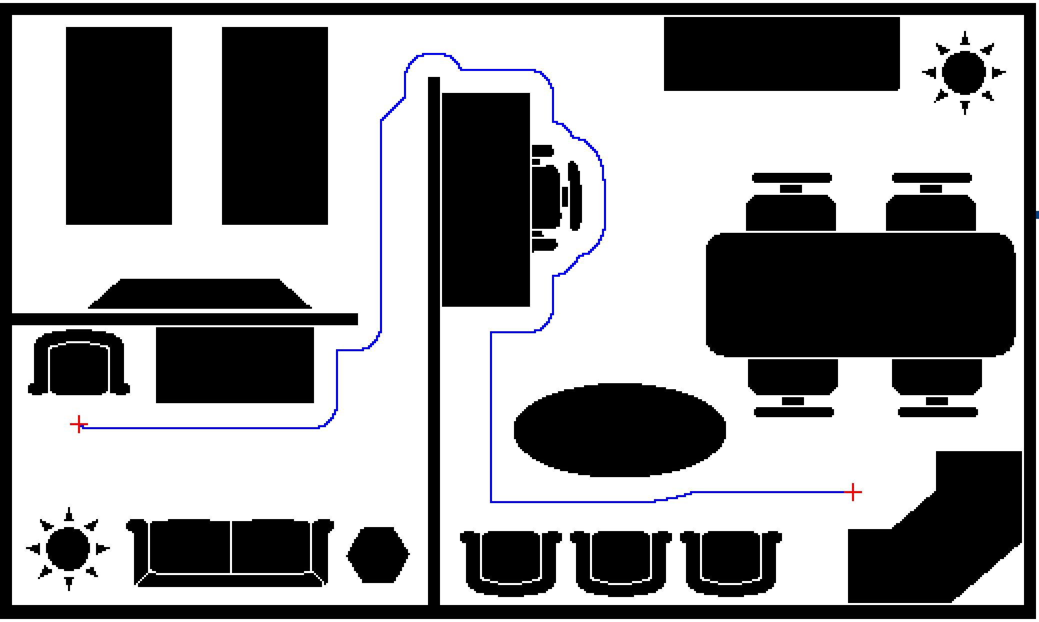
Minimum path 4-conn

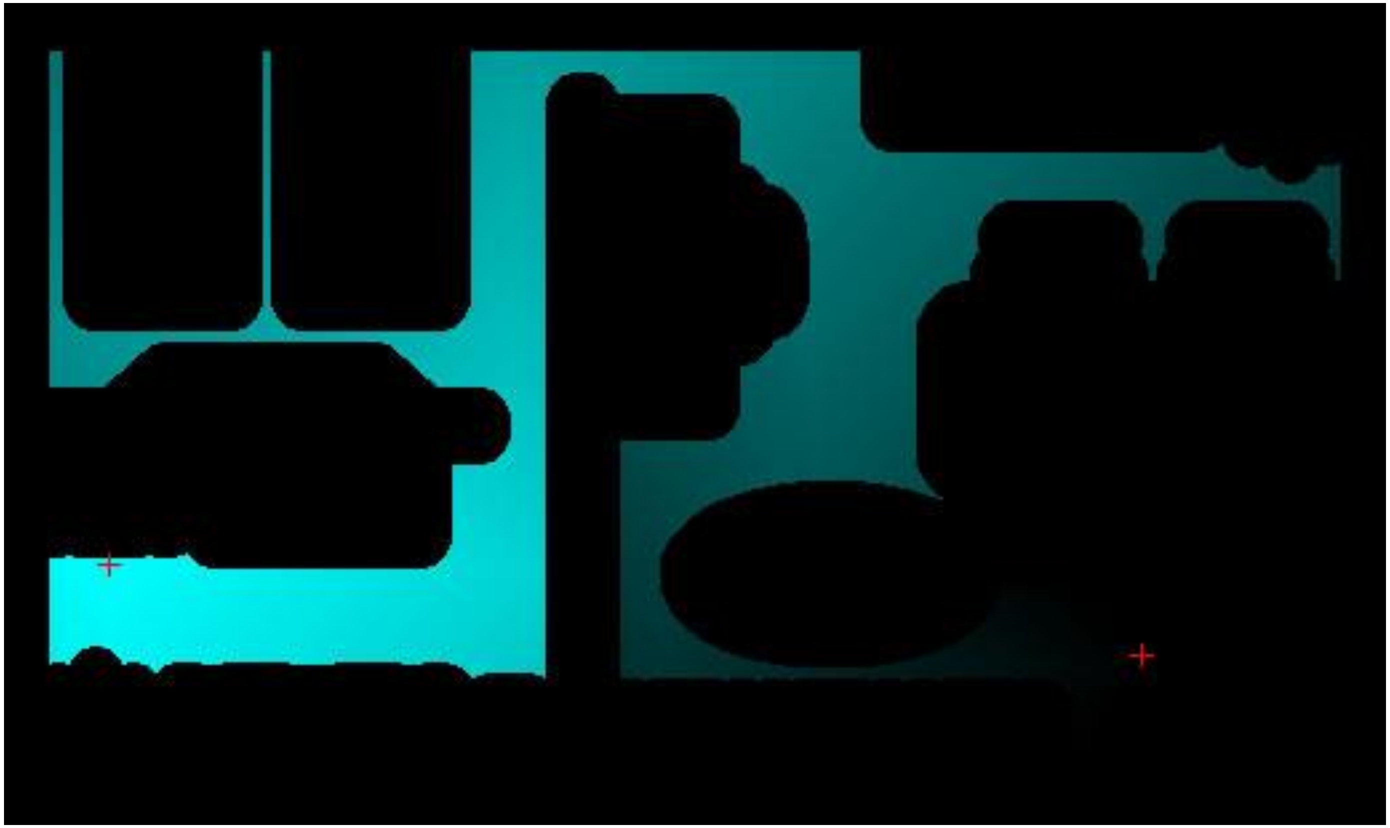


Minimum path
8-conn

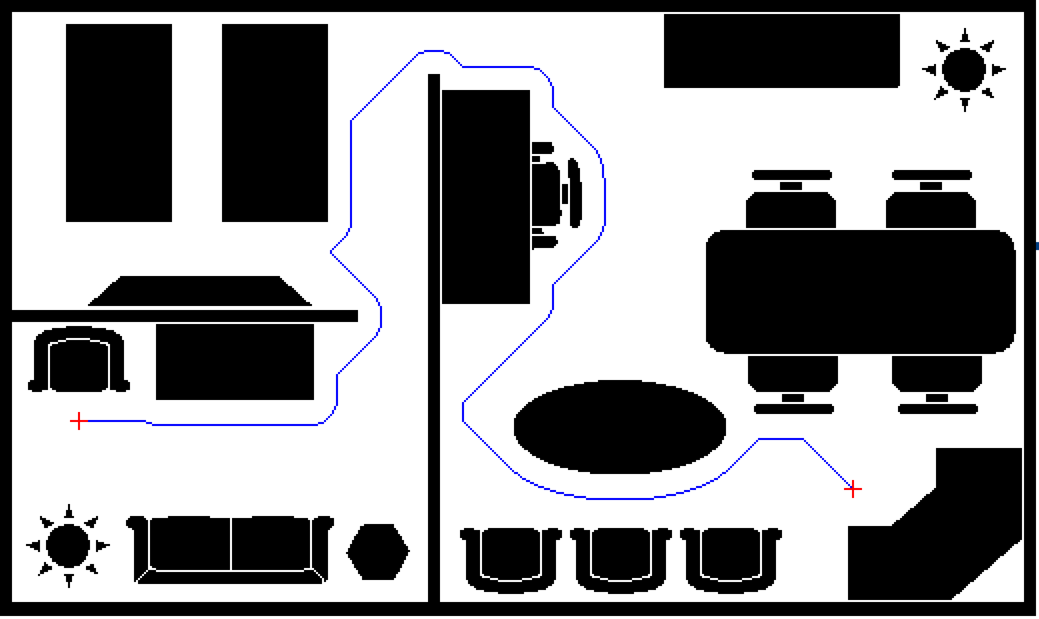


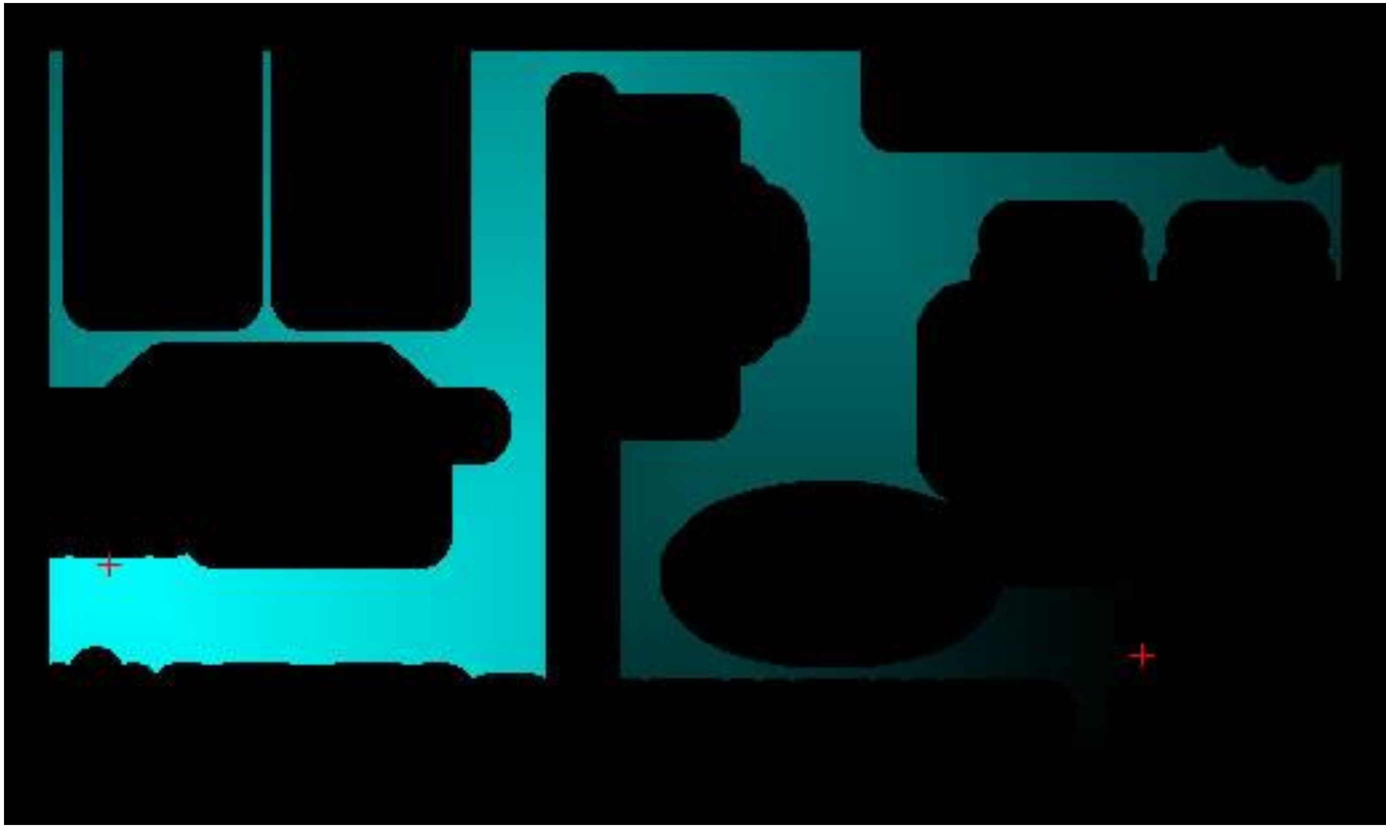
Minimum path
4-conn



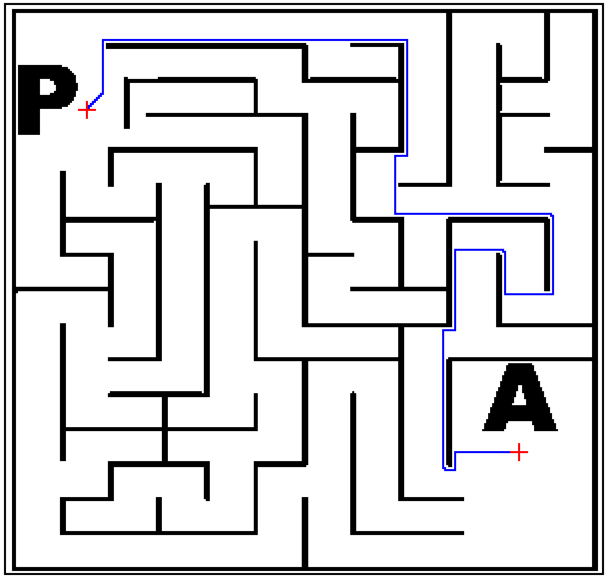


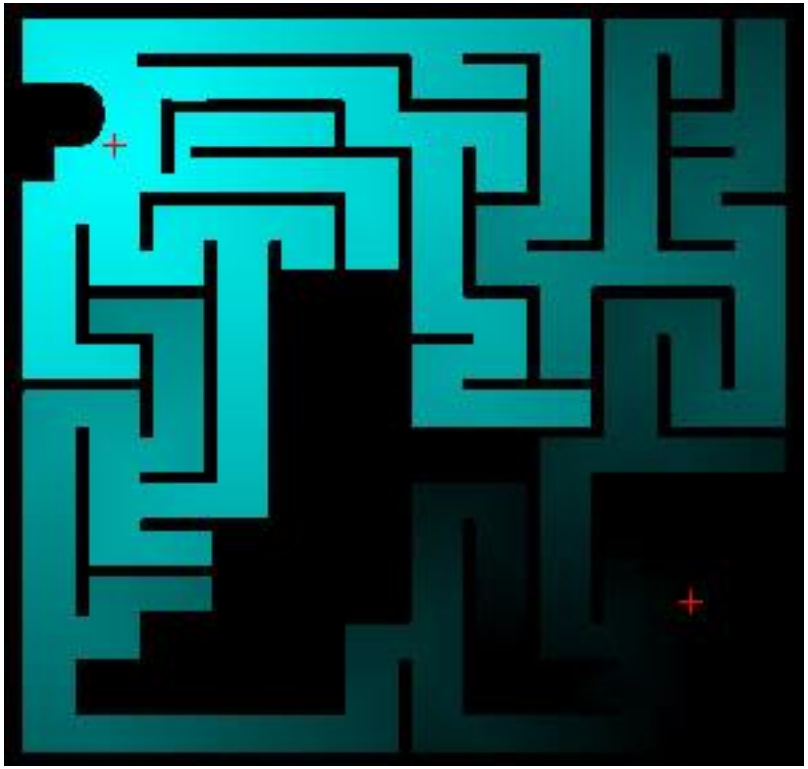
Minimum path 8-conn





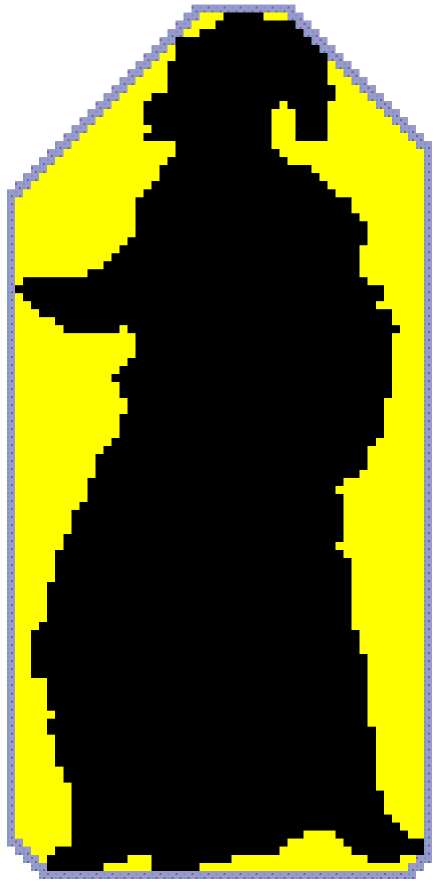
Minimum path
4-conn





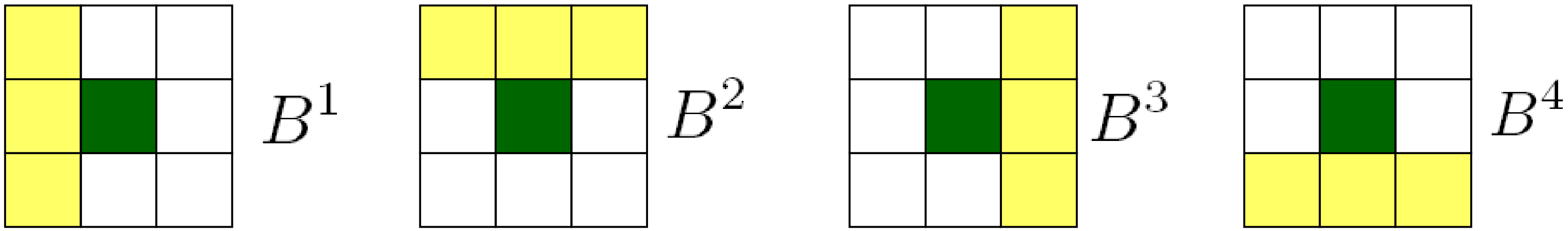
8-Convex Hull

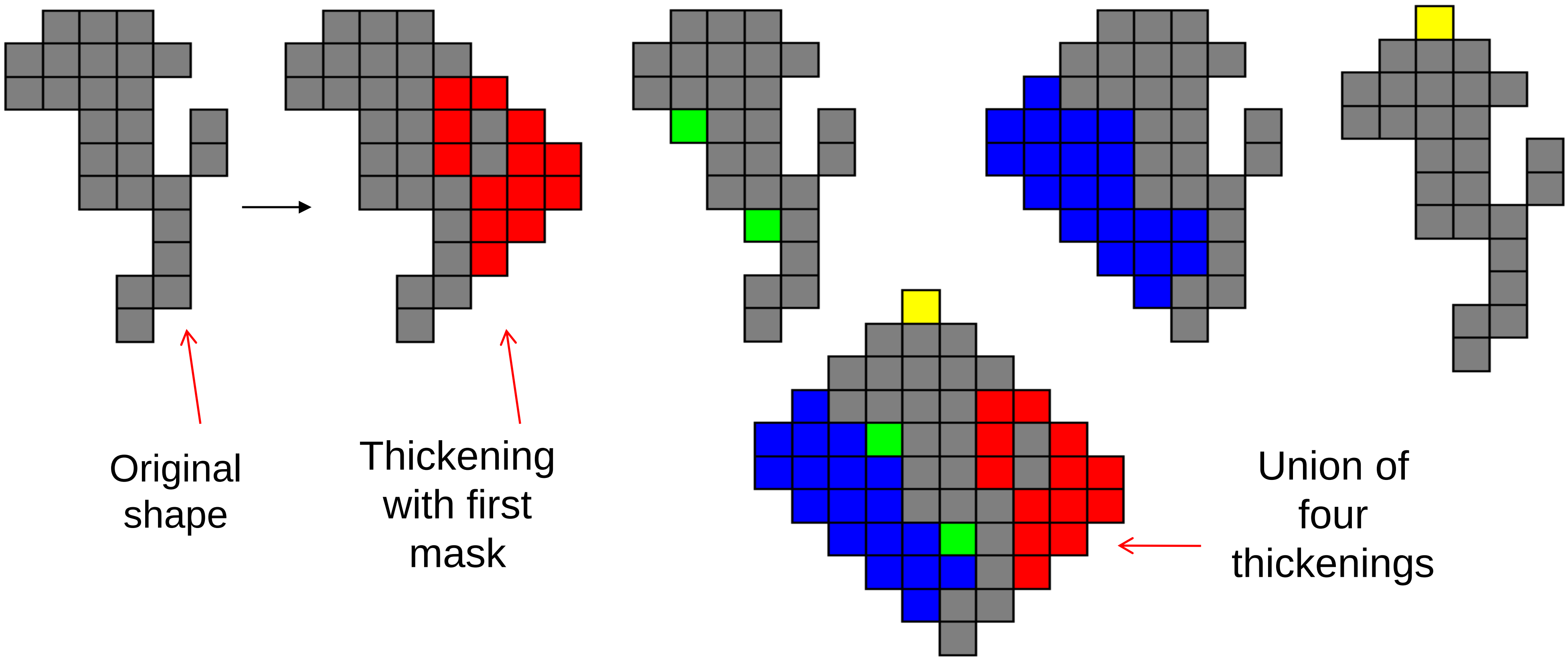
- A set A is said to be convex if the straight line segment joining any two points in A lies entirely within A.
- The convex hull is the minimum n-sided convex polygon that completely circumscribes an object, gives another possible description of a binary object. An example is given in figure where a constrained 8-sided polygon has been chosen to coarsely describe the monk silhouette.
- To obtain the convex hull a simple algorithm propagates the object along the eight (more generally 2n) orientations and then: i) logically OR the opposite propagated segments; and ii) logically AND the four (more generally n) resulting segments. The contour of the obtained polygon is the convex hull.



Use of thickening: Convex hull

- Convex hull: union of thickenings, each up to idempotence



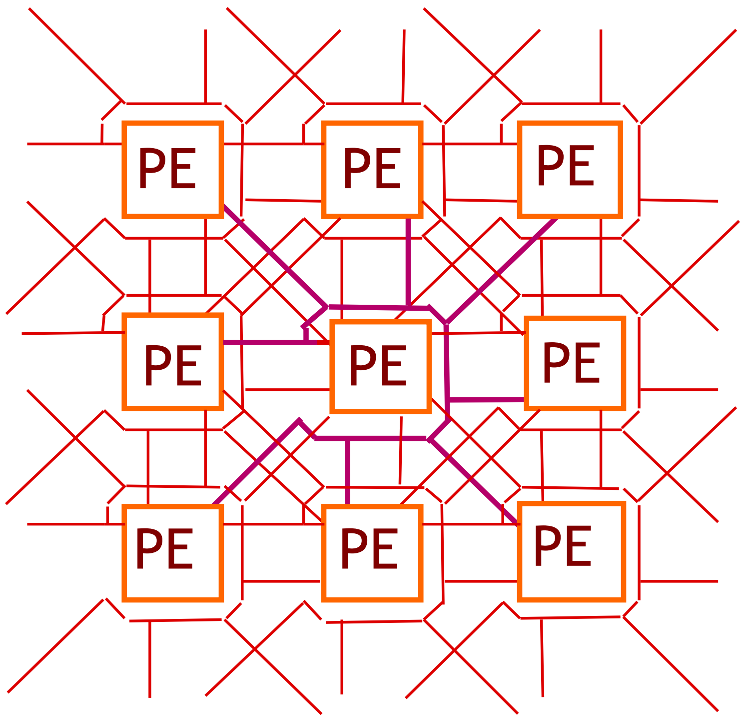


D.A. Forsyth

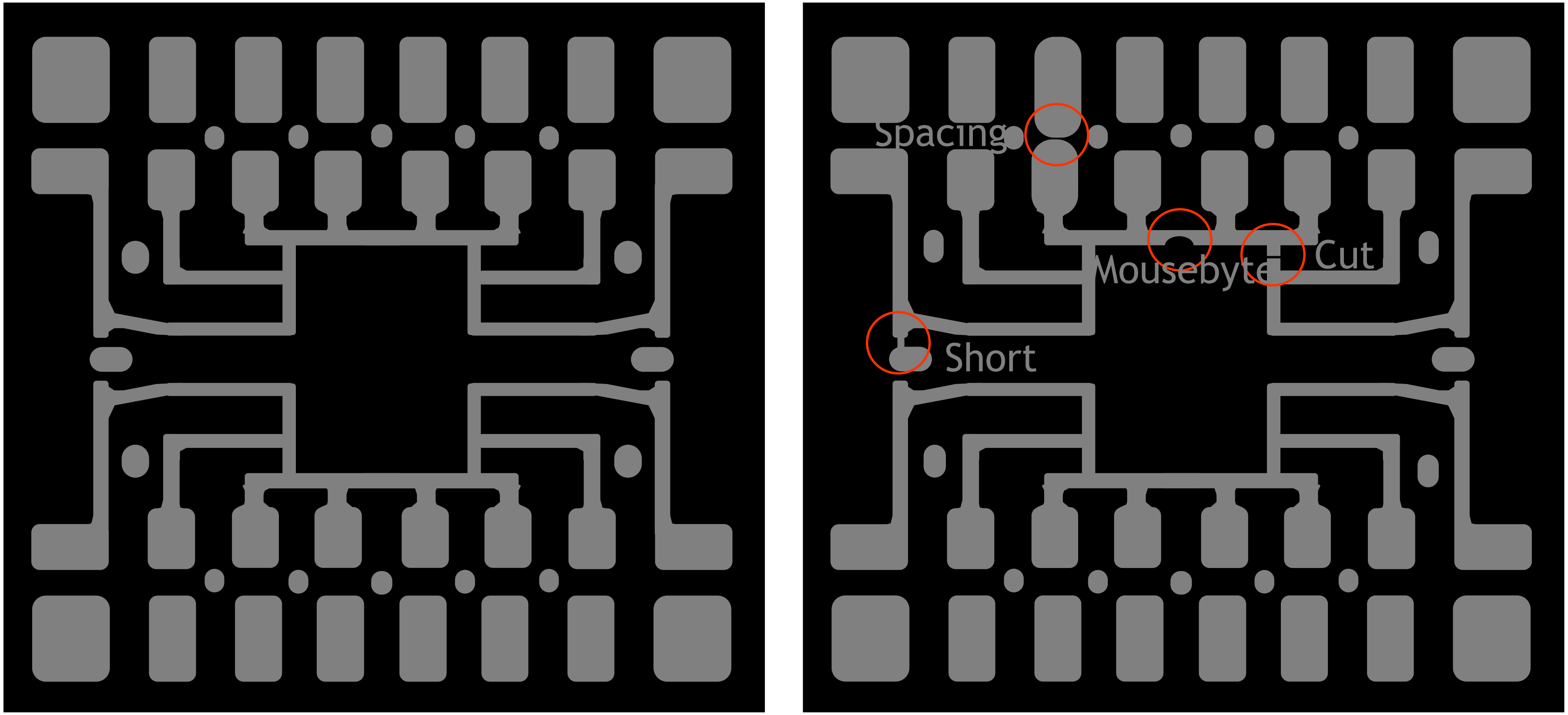
Example of using convex hull

Pixel Parallelism: Processor arrays

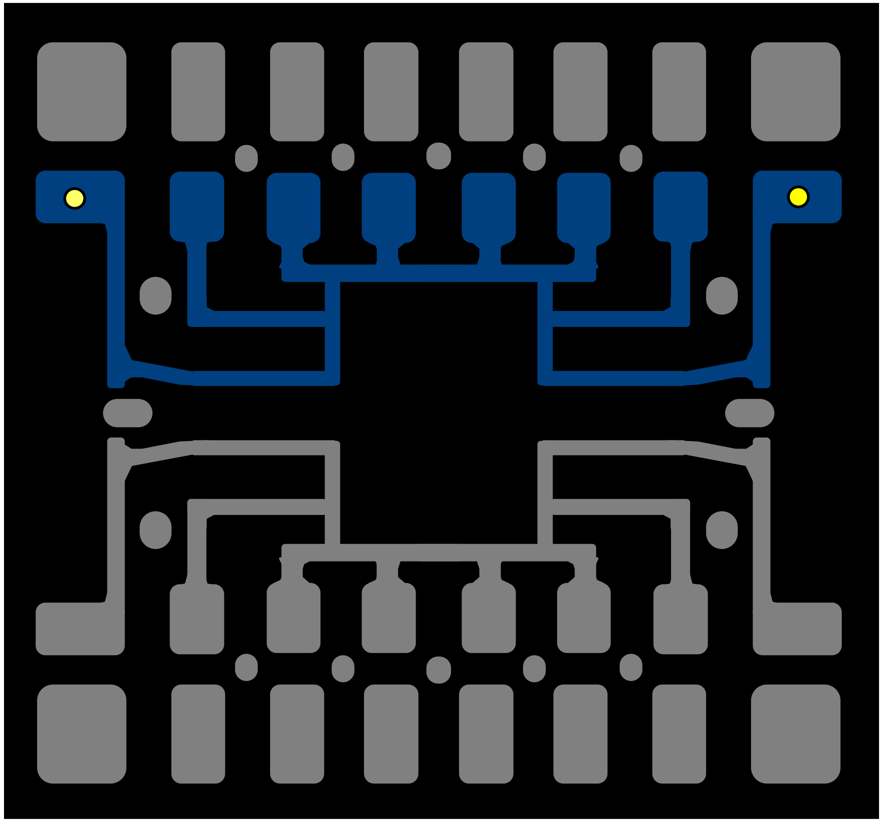
- Processor Element (PE) includes local memory
- Image distributed over all PE
- All PE run the same program (SIMD)



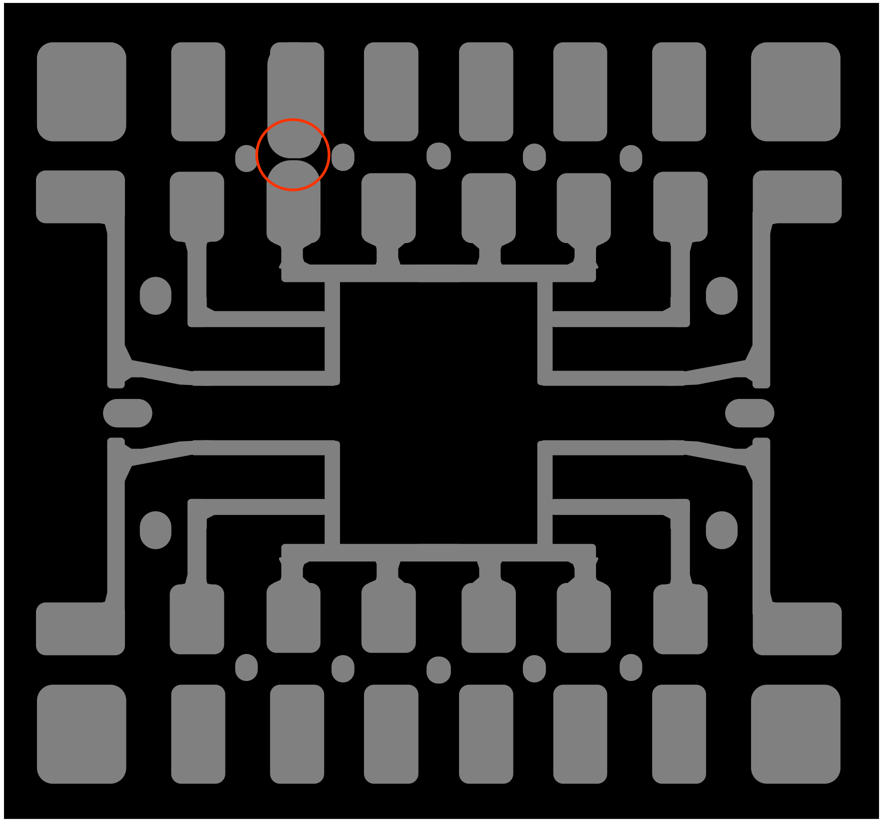
Propagation: examples



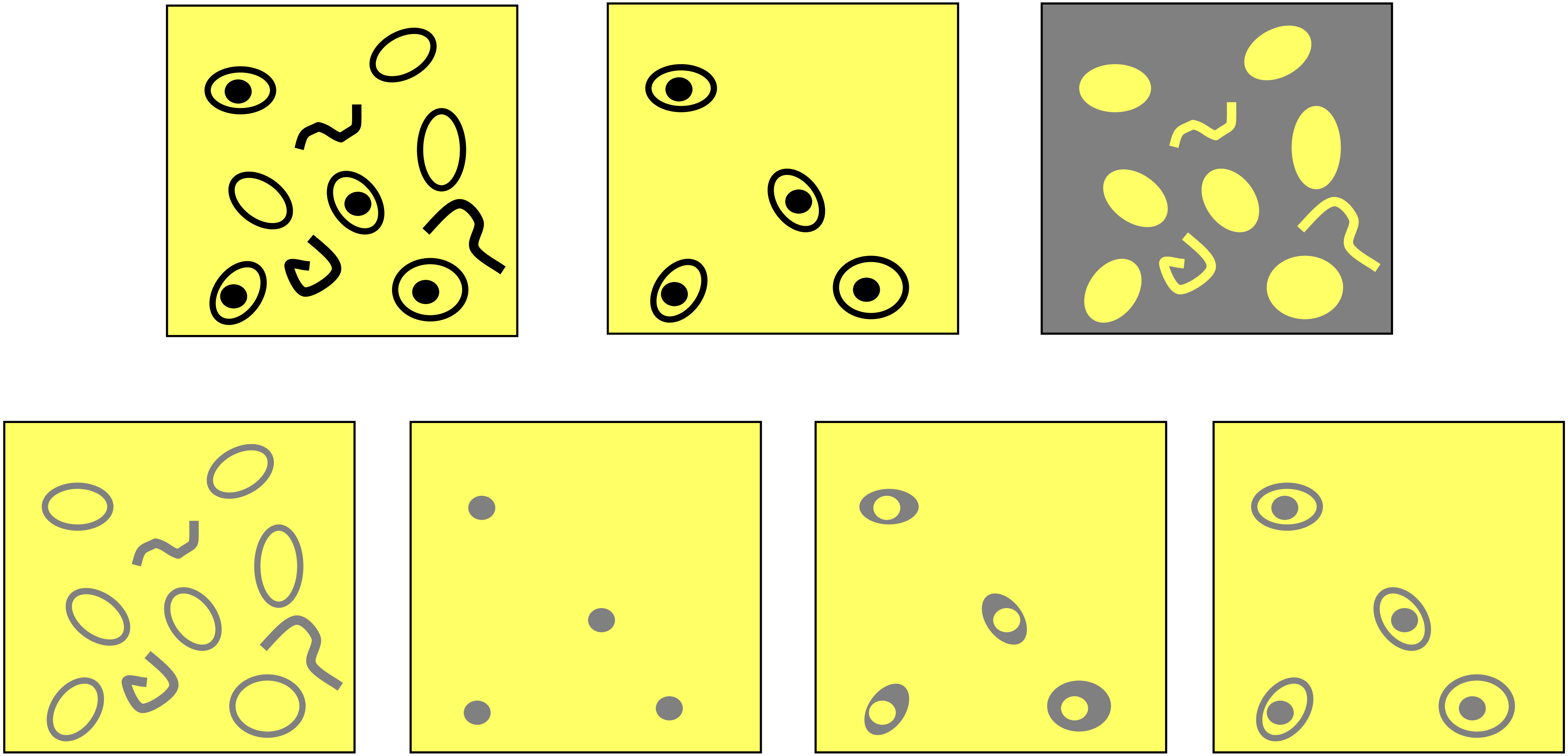
Mousebyte



Minimum distance



Global OR (Or-sum-tree)



Global OR (Or-sum-tree)