.

Morphology

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it

Methods for Image Processing

academic year 2015–2016

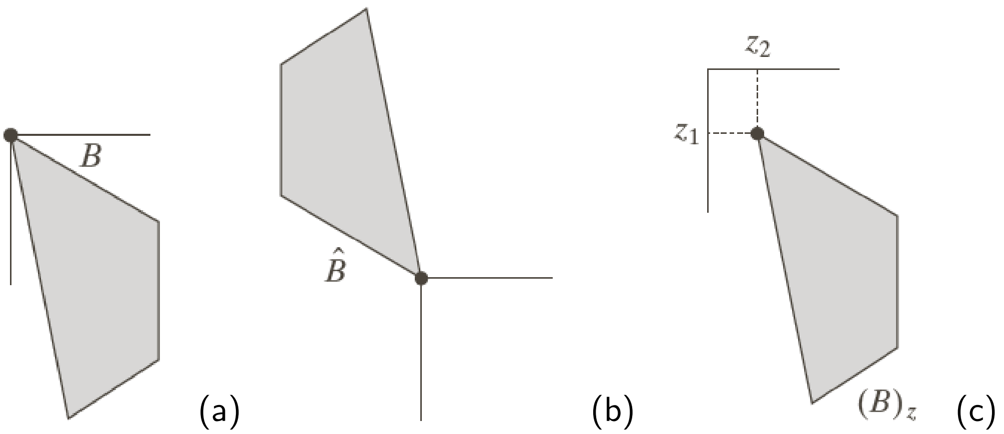
Morphological processing

- The *morphology* of an image describes the shapes represented in the image.
- At a low level, the objects are represented as clusters of pixels, which are distributed in the spatial plane with a law that depends by the features of the represented object.
- Morphology based processing exploit the a-priori knowledge on these features.
- In particular, they make use of local features of the neighboring pixels.
- The morphological processes can be formalized as set operation on set of points of the plane.
  - For sake of simplicity, points belonging to $\mathbb{Z}^2$ are considered, but the morphological operation can be generalized for other domains (e.g., $\mathbb{Z}^n$, or $\mathbb{R}^2$).

Morphological processing (2)

- They are easily defined on binary images, where concepts of membership and complement can be associated to the pixel binary color, but they can be extended to gray level images.
- A binary image, f, can be used for describing a set of points of $\mathbb{Z}^2$, B:
  - if $f(x, y)$ is white, $(x, y) \in B$;
  - if $f(x, y)$ is black, $(x, y) \notin B$.
  - $B = \{(x, y) \mid f(x, y) = 1\}$
- Note: in the illustrative figures, the considered sets are depicted in gray, while the background is white.

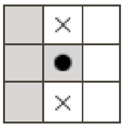
Definitions



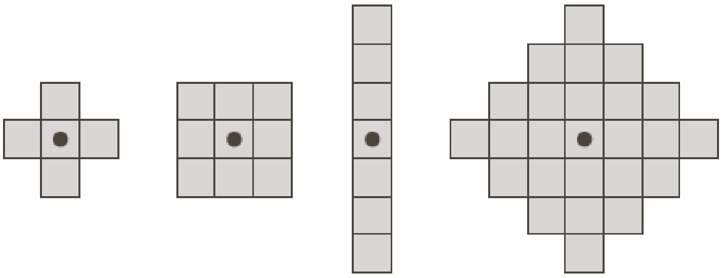
- Let a set, B, and the point *origin*, the operators *reflection* and *translation* can be defined.
- The reflection, $\hat{B}$, is defined as: $\hat{B} = \{-b \mid b \in B\}$.
- The translation of z, $(B)_z$, is defined as: $(B)_z = \{b + z \mid b \in B\}$

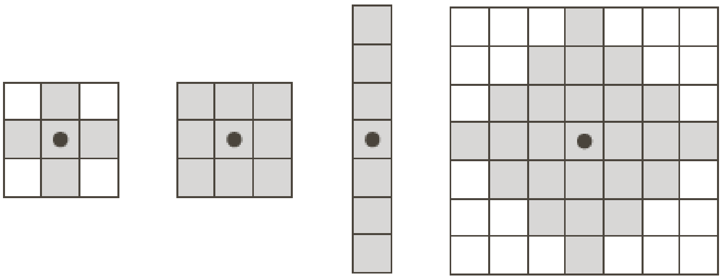
Structuring element

- The morphological operations are generally defined with respect to a particular set, called *structuring element*.
- The structuring elements for images are pixel arrays themselves.
- The structuring elements are defined relatively an origin.
  - Typically, it is the center of gravity.
- Structuring elements are described using the following convention:
  - filled cell: belong to the structuring element;
  - void cell: does not belong to the structuring element;
  - cross: don't care.



Structuring element (2)





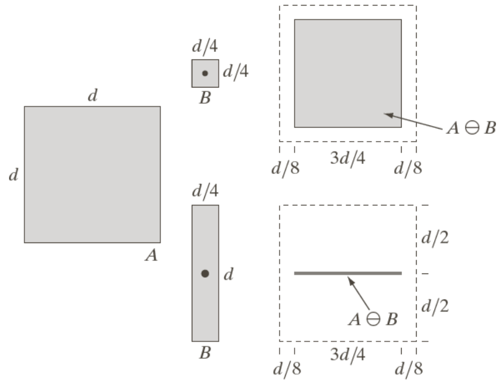
Erosion

Given A and B, the *erosion* of A through B, $A \ominus B$, is defined as:

$$A \ominus B = \{z \,|\, (B)_z \subseteq A\}$$

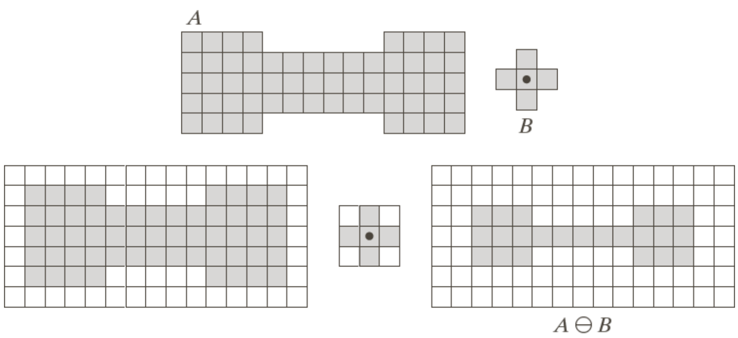
Equivalently, A eroded B can be defined as:

$$A \ominus B = \{z \,|\, (B)_z \cap A^c = \emptyset\}$$



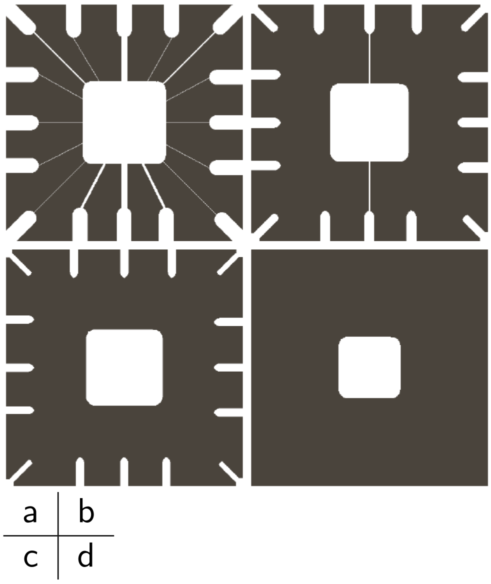
Erosion (2)

- If A and B are images, the morphological operations are computed shifting the origin of the structuring element in each pixel of the image A, evaluating if the definition of the operation is satisfied.
  - Padding can be required.
- For the erosion:
  - the origin of B is translated on a pixel $a \in A$;
  - if all the elements of B are covered by an element of A, the pixel a belong to $A \ominus B$.



.

Morphological filtering



- The erosion can be used for realizing a shape based filtering (*morphological filtering*).

(a) 486×486 binary image;
(b) erosion of (a) with 11×11 square structuring element;
(c) with 15×15 square;
(d) and with 45×45 square.

- Erosion cancels the details smaller than the structuring element.

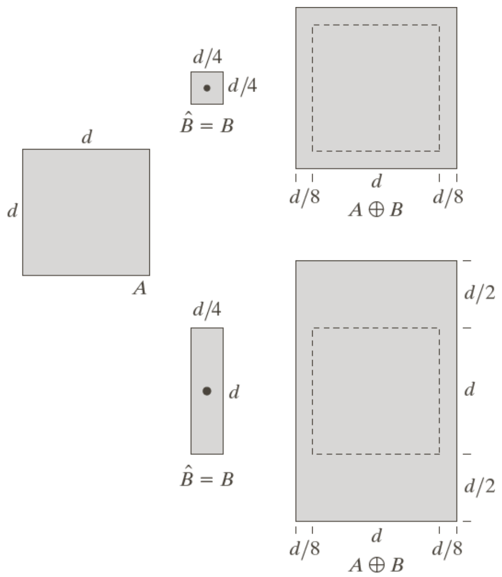
Dilation

Given the sets A and B, the *dilation* of A through B, $A \oplus B$, is defined as:

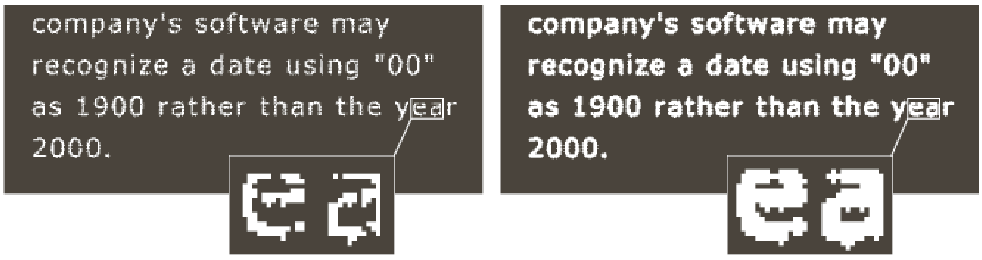
$$A \oplus B = \{z \,|\, (\hat{B})_z \cap A \neq \emptyset\}$$

Equivalently, A dilated B can be defined as:

$$A \oplus B = \{z \,|\, ((\hat{B})_z \cap A) \subseteq A\}$$



Dilation (example)



0	1	0
1	1	1
0	1	0

- The dilation has effects similar to those of the lowpass filtering: the details are absorbed.
- In the considered case, the dilation engrosses the characters, filling the spaces between the fragments.

Duality

- Erosion and dilation are operations dual with respect to the complement and reflection:

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

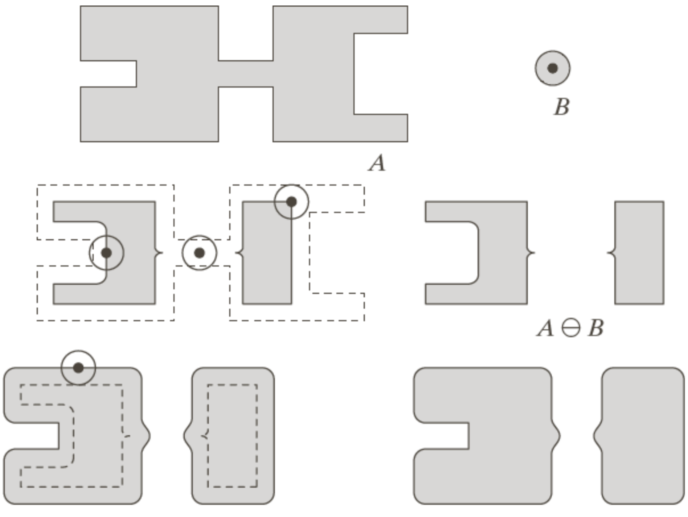
- If the structuring element is symmetric ($\hat{B} = B$), the erosion of A can be obtained dilating the background, A^C, with the same structuring element, and complementing the result (vice versa for the dilation).

.

Opening

The *opening* of a set A through B, $A \circ B$, is defined as:

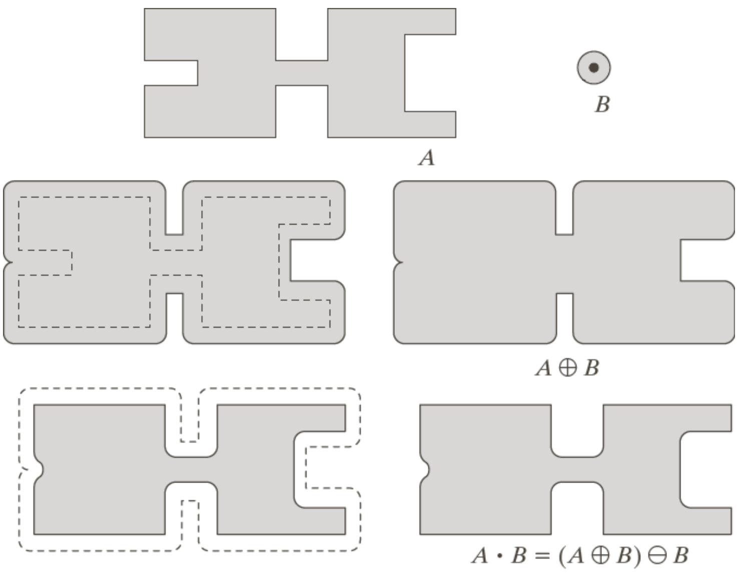
$$A \circ B = (A \ominus B) \oplus B$$



Closing

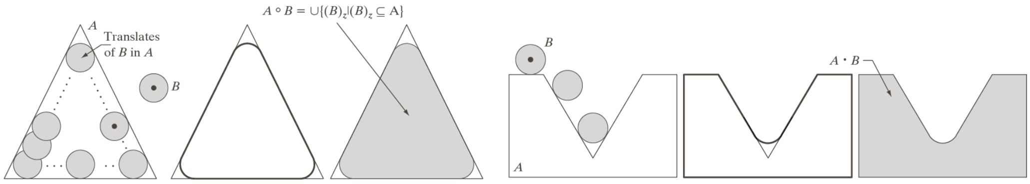
The *closing* of a set A through B, $A \bullet B$, is defined as:

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



Opening and closing

- Opening and closing eliminates the details:
  - opening eliminates protrusions and bridges that are too thin;
  - closing fill the gulfs and the holes that are too narrow.
- They have a simple geometrical interpretation:
  - opening results as the points of A covered by the translation of B along the inner border of A;
  - closing results adding to A those points of the background that are not covered by the translation of B along the outer border of A.



Opening and closing properties

Like dilation and erosion, also the opening and closing are dual operations with respect to complement and reflection:

- $(A \bullet B)^c = A^c \circ \hat{B}$
- $(A \circ B)^c = A^c \bullet \hat{B}$

Besides, the following properties hold:

- $A \circ B \subseteq A \subseteq A \bullet B$
- $(A \circ B) \circ B = A \circ B$
- $(A \bullet B) \bullet B = A \bullet B$
- $C \subseteq D \Rightarrow C \circ B \subseteq D \circ B$
- $C \subseteq D \Rightarrow C \bullet B \subseteq D \bullet B$

Opening and closing: an example

- Opening and closing can be used for noise filtering.
- The fingerprint in A is affected by noise.
- Eroding, the outer noise is eliminated, but inner noise is improved.
- Following with dilation, the original dimension of the ridges can be recovered and the inner noise is canceled.
- erosion + dilation = opening

A

1	1	1
1	1	1
1	1	1

B

$A \ominus B$

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

Opening and closing: an example (2)

- Noise has been removed by the opening, but this processing stage caused the interruption of some ridges.
- By dilating, the continuity of the ridges can be recovered.
- Then erosion can restore the original thickness of the ridges.
- dilation + erosion = closing

$(A \circ B) \oplus B$

$((A \circ B) \oplus B) \ominus B = (A \circ B) \bullet B$

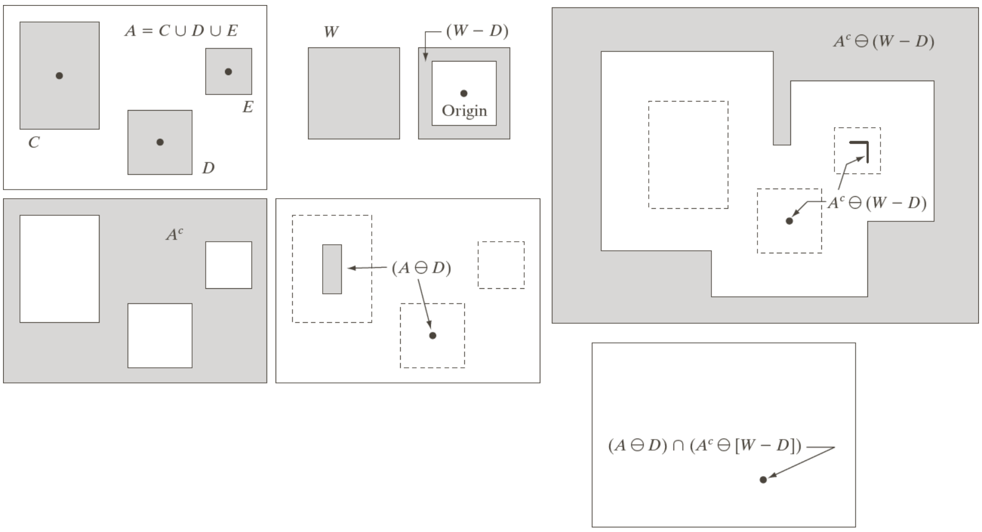
Hit or miss

- The *hit-or-miss* transformation allows to detect disjointed shapes.
  - The objects have to be separated by at least one background pixel.
- The processing is based on a structuring element (shaped as the object to be detected) and its local background (a window larger than the structuring element).
- Let A a set constituted of several regions, $A = C \cup D \cup E$, B the shape to be detected and its local background, $B = (D, W - D) = (B_1, B_2)$.
- The hit-or-miss transform $A \circledast B$ is defined as:

$$A \circledast B = A \ominus D \cap (A^c \ominus (W - D))$$

- Equivalent definitions:
  - $A \circledast B = A \ominus B_1 \cap A^c \ominus B_2$
  - $A \circledast B = A \ominus B_1 - A^c \oplus \hat{B}_2$

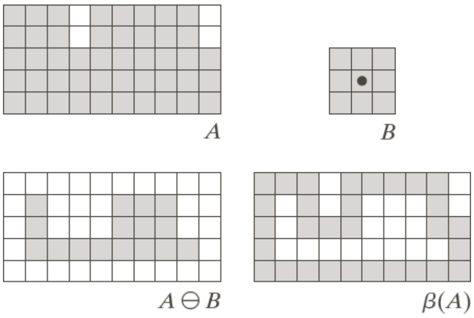
Hit or miss (2)



Border extraction

- The border of A, $\beta(A)$, can be obtained as:

$$\beta(A) = A - (A \ominus B)$$



- The shape (and the size) of B affects the thickness of the border.

Hole filling

- A *hole* is a background region surrounded by a connected border of *foreground* elements.
- Let A a set containing 8-connected borders that enclose a background region (holes), which have to be filled (i.e., set to 1).
- The sequence $X_0, \dots, X_k$ can be constructed, where X_0 is a set containing a point of each hole and X_j is defined as:

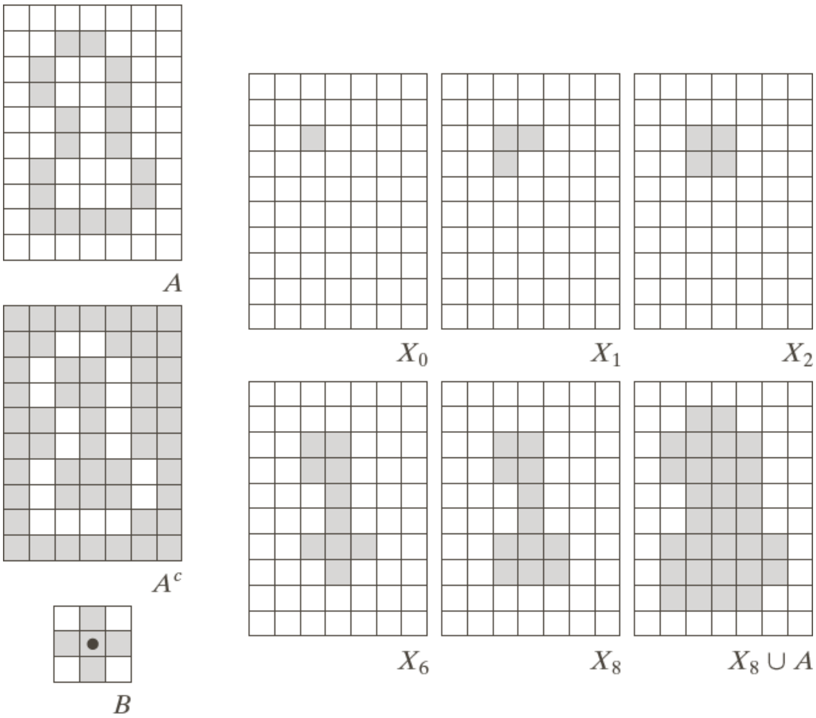
$$X_j = (X_{j-1} \oplus B) \cap A^c$$

for $B =$

0	1	0
1	1	1
0	1	0

- The algorithm ends for a value of k such that $X_k = X_{k-1}$, X_k contains all the filled holes.
- Hence, $A \cup X_k$ contains A without holes.
- The intersection with A^c constraints the dilation inside of the region of interest.

Hole filling (2)



Hole filling: an example

a | b | c

- The binarized image of metallic spheres contains inner regions caused by reflection.
- They can be eliminated using a hole filling algorithm:
  (a) a starting position have to be selected (manually or automatically);
  (b) the hole filling procedure eliminate the selected hole;
  (c) the procedure can be repeated for each hole.

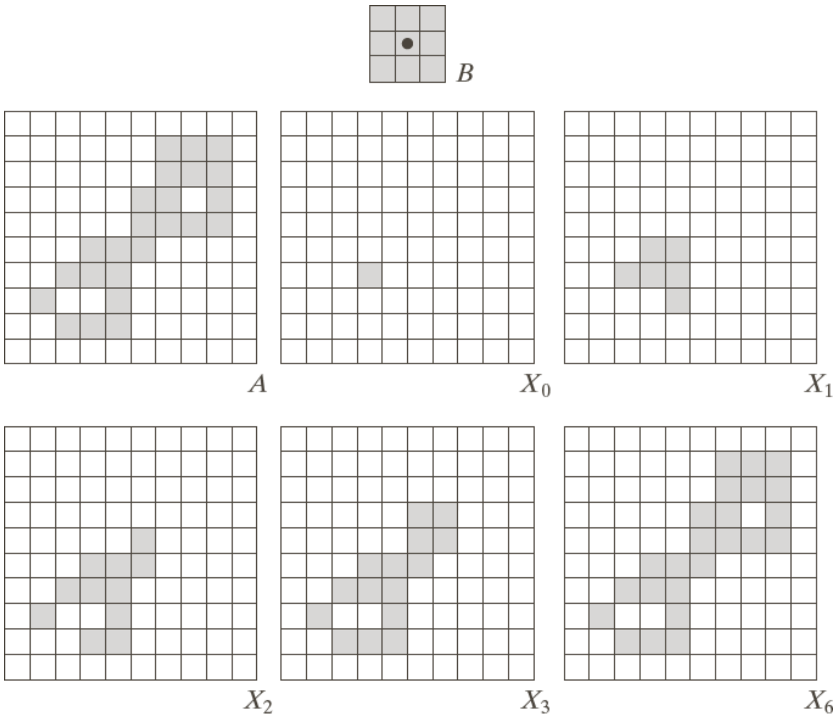
Connected components extraction

- The extraction of the connected components of a binary image is a fundamental process for the automatic digital image processing.
- Let A a set containing one or more connected components, X_0 a set containing a point for each connected components of A and X_k is a set defined as follows:

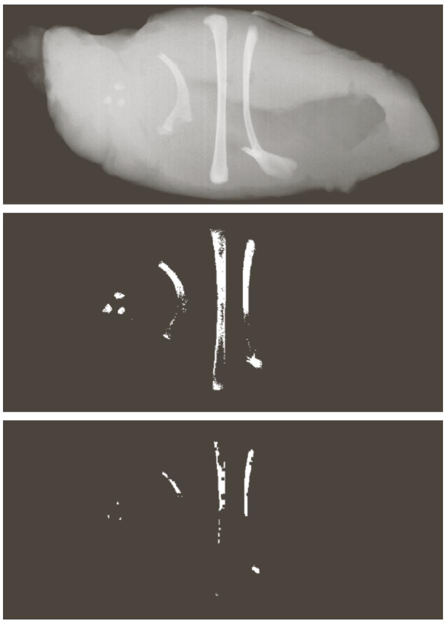
$$X_k = (X_{k-1} \oplus B) \cap A$$

  where B is a structuring element.
- For $X_k = X_{k-1}$, the set X_k contains all the connected components of A.
- Note: the operation is similar to that of hole filling, but it make use of A (instead of A^c) for masking the dilation.

Connected components extraction (2)



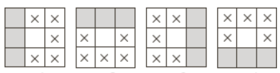
Connected components extraction: an example



- The presence of bones inside the chicken breast can be detected through an X-ray image.
- After a suitable thresholding, the erosion with a appropriate structuring element left only the objects that are not due to noise.
- The count of the resulting connected components pixels allows to estimate the size of the remaining bones.

Convex hull

- The (*convex hull*), H, of a set A is the smallest convex set containing A.
  - A region is convex if every segment joining two points belonging to the considered region is in the region.
- Let B^1, B^2, B^3 and B^4 the structuring elements:



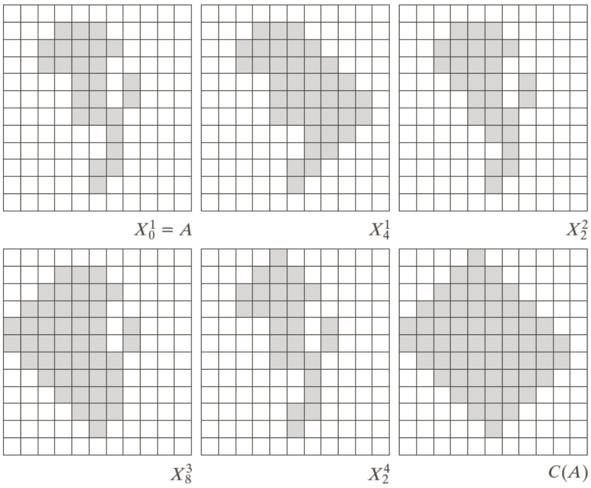
B^1 B^2 B^3 B^4

and $X_k^i = (X_{k-1}^i \circledast B^i) \cup X_{k-1}^i$, with $X_0^i = A$.
- Let $D^i = X_k^i$, for k such that $X_k^i = X_{k-1}^i$, for every i.
- The convex hull of A, $C(A)$, can be computed as:

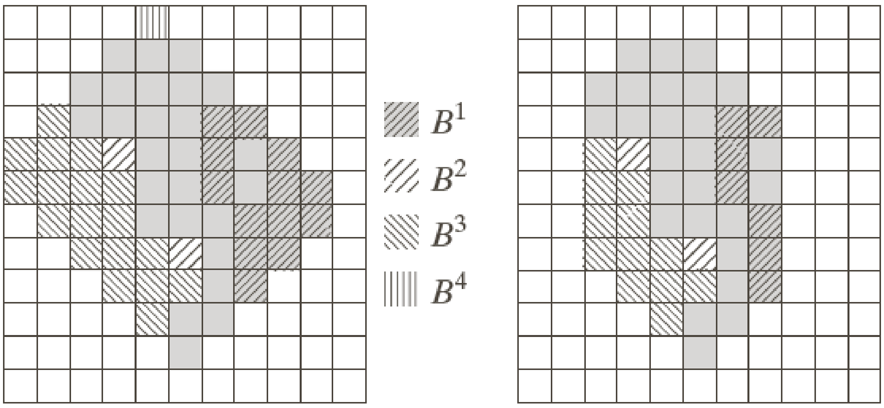
$$C(A) = \bigcup_i D^i$$

Convex hull: the algorithm

- In the practice, starting from A, the hit-or-miss transformation with B^1 is iterated until it reach a stable state.
- Then, the process is repeated with B^2, B^3, and B^4.
- The union of the so obtained four sets with A provides $C(A)$.



Convex hull: notes



- The hit-or-miss transformation here used does not require the local background of the structuring element.
- Every B^i add elements in a given direction.
- A useful technique consists in limiting the growing process only inside the smallest rectangle that contains A (*bounding box*).

Thinning

- The *thinning* of a set A through B, $A \otimes B$, can be defined as:

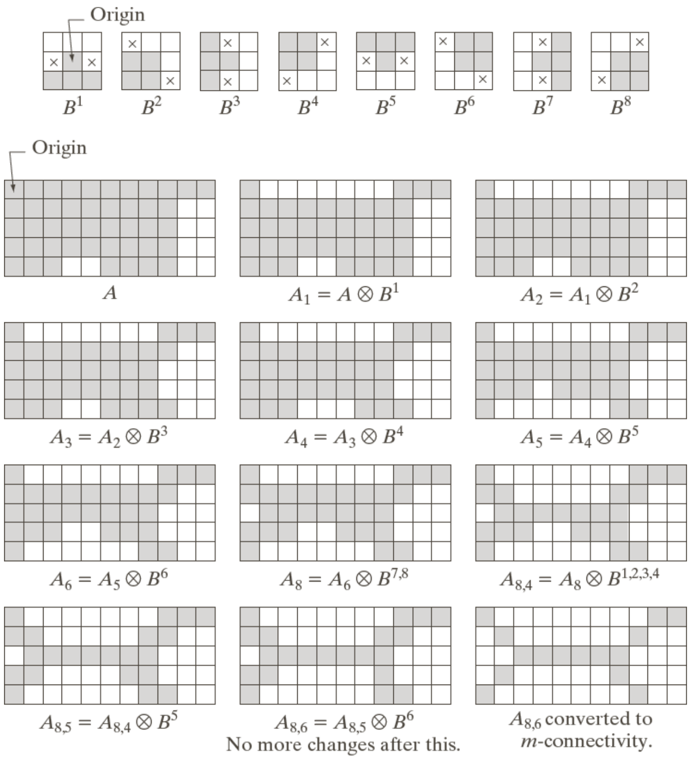
$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

  - The hit-or-miss transformation here used does not require the local background.
- Sometimes, defining different structuring elements for different directions, $\{B\} = \{B^1, \ldots, B^n\}$, can simplify the procedure. They are applied in sequence:

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

- Then, the results can be further processed for avoiding multiple paths (m-connectivity).

Thinning: an example



Thickening

- The *thickening* of a set A through B, $A \otimes B$, can be defined as:

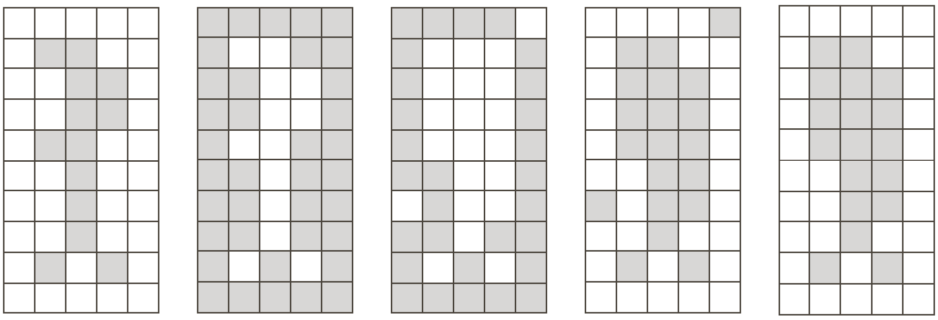
$$A \odot B = A \cup (A \circledast B)$$

  - The hit-or-miss transformation does not require the local background.
- It is the dual transformation of the thinning.
- It can be defined using a sequence of structuring elements:

$$A \odot \{B\} = (\cdots((A \odot B^1) \odot B^2) \cdots) \odot B^n$$

  where the structuring elements are the complements of those used for the thinning.

Thickening: an example



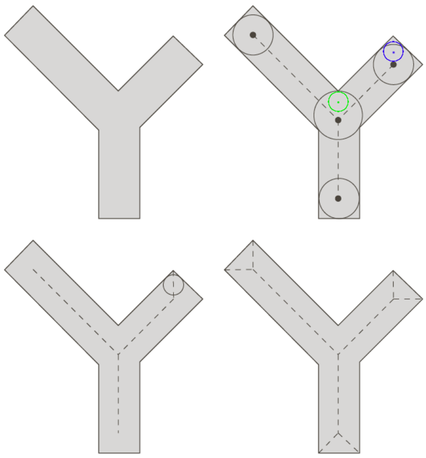
- The thickening is often realized by thinning the background.
- This method can produce disconnected points, which have to be removed, but the thinned background limits the thickening and the result is generally better than that obtained through the direct application of the thickening algorithm.

Skeletonization

- The *skeleton*, $S(A)$, of a set A can be intuitively defined as the centers of the minimum collection of circular disks that covers A.
- More formally, the concept of *maximum disk* has to be defined:
  - a disk $(D)_z$, positioned in $z \in A$, is said *maximum* if no other disk completely in A that contains $(D)_z$ can be positioned;
- and the skeleton of A, $S(A)$, can be defined as:

$$S(A) = \{z \in A \,|\, (D)_z \text{ is a maximum disk in } A\}$$

Skeleton



a	b
c	d

(a) The considered set, A.

(b) The black disks are maximum disks in A. The green disk is not maximum because it is included in another disk in A. The dashed lines are the centers of maximum disks: they are in $S(A)$. The blue disk is positioned in a point that does not belong to the dashed lines and is not completely included in any black disk.

(c) New points of $S(A)$ are found.

(d) The skeleton of A, $S(A)$.

Morphological definition of skeleton

- The *skeleton*, $S(A)$, of a set A can be defined in terms of morphological operations.
- It can be shown that:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

with

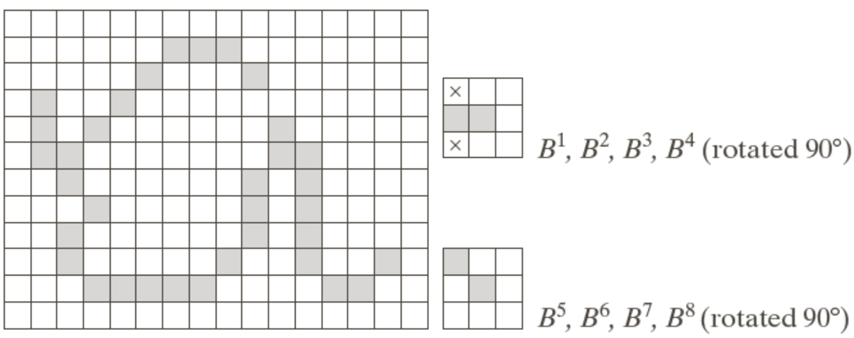
$$S_k(A) = (A \ominus^k B) - (A \ominus^k B) \circ B$$

where B is a structuring element, $(A \ominus^k B)$ means k successive erosions, and K is the last iteration before the empty set is obtained: $K = \max\{k \mid A \ominus^k B \neq \emptyset\}$.

Pruning

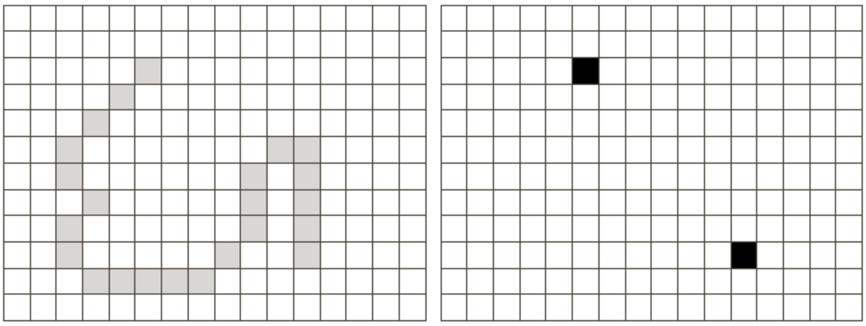
- The *pruning* is the typical post-processing step of the skelotonization algorithm;
  - usually some spurious branches are produced.

$$X_1 = A \otimes \{B\}$$



- For instance, every branch with less than three pixels can be considered spurious.
- $\{B\}$: three times the sequence B^1–B^8

Pruning (2)



X_1 X_2

- The terminal points set, X_2, can be obtained:

$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$

Pruning (3)

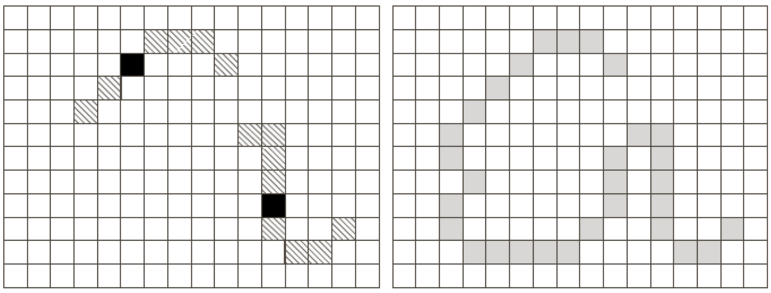
- The terminal points are dilated, constraining the result in A:

$$X_3 = (X_2 \oplus H) \cap A \text{ (three times)}$$

with $H = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$

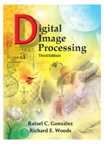
- The pruning ends with the union of the two intermediate sets:

$$X_4 = X_1 \cup X_3$$



X_3 X_4

Homeworks and suggested readings



DIP, Sections 9.1–9.5.8

- pp. 627–656

GIMP

- Filter
  - Generic
    - Dilate
    - Erode