7-9 Monday – 309-GD2

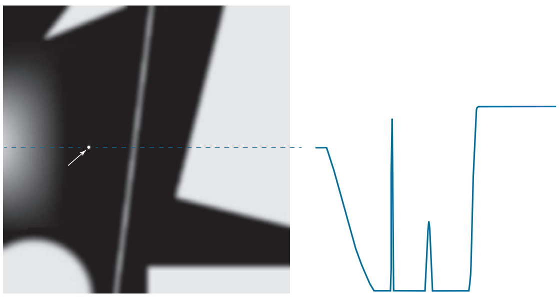
Xử lý ảnh
INT3404 1

Giảng viên: TS. Nguyễn Thị Ngọc Diệp

Email: ngocdiep@vnu.edu.vn

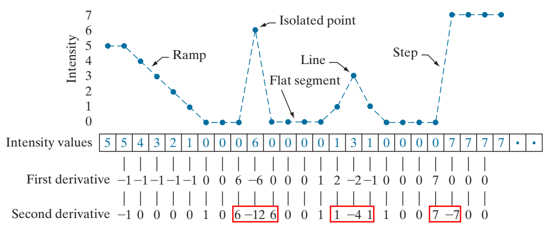
Slide & code: https://github.com/chupibk/INT3404_1

Ôn lại tuần 5: Tìm cạnh (edge detection)



(1) Detecting the local maxima or minima of the first derivative

(2) Detecting the zero-crossings of the second derivative



Lịch trình

Tuần	Nội dung	Yêu cầu đối với sinh viên
1	Giới thiệu môn học Làm quen với OpenCV + Python	Cài đặt môi trường: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Phép toán điểm (Point operations) – Điều chỉnh độ tương phản – Ghép ảnh	Làm bài tập 1: điều chỉnh gamma tìm contrast hợp lý
3	Histogram - Histogram equalization - Phân loại ảnh dùng so sánh histogram	Thực hành ở nhà
4	Phép lọc trong không gian điểm ảnh (linear processing filtering) - làm mịn, làm sắc ảnh	Thực hành ở nhà Tìm hiểu thêm các phép lọc
5	Tìm cạnh (edge detection)	Thực hành ở nhà
6	Các phép toán hình thái (Erosion, Dilation, Opening, Closing) - tìm biển số	Làm bài tập 2: tìm barcode
7	Chuyển đổi không gian - miền tần số (Fourier) - Hough transform	Thực hành ở nhà
8	Phân vùng (segmentation) - depth estimation - threshold-based - watershed/grabcut	Đăng ký thực hiện bài tập lớn
9	Mô hình màu Chuyển đổi giữa các mô hình màu	Làm bài tập 3: Chuyển đổi mô hình màu và thực hiện phân vùng
10	Mô hình nhiễu -Giảm nhiễu -Khôi phục ảnh -Giảm nhiễu chu kỳ - Ước lượng hàm Degration -Hàm lọc ngược, hàm lọc Wiener	Thực hành ở nhà
11	Template matching – Image Matching	Làm bài tập 4: puzzle
12	Nén ảnh	Thực hành ở nhà
13	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
14	Hướng dẫn thực hiện đồ án môn học	Trình bày đồ án môn học
15	Tổng kết cuối kỳ	Ôn tập

Morphology

- In biology: morphology = form + structure of animals and plants
- Mathematical morphology = a tool for extracting region shape
  - Boundaries, connected components

Set theory of morphology

- Binary image = a set of 2-D integer space Z^2
- Grayscale digital image = a set of 3-D elements Z^3
- Morphological operations are defined in terms of sets
- 2 sets of pixels:
  - Objects → foreground pixels
  - Structuring elements

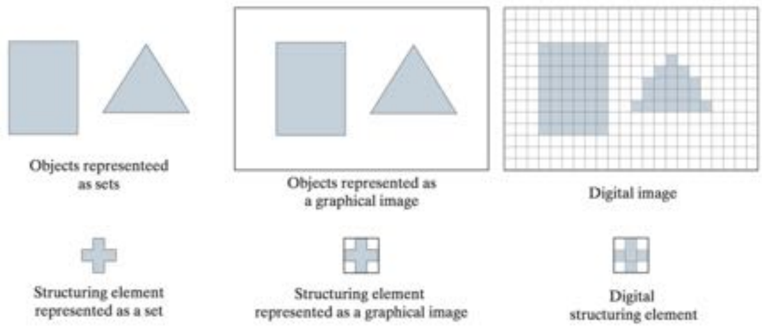


FIGURE 9.1 Top row. *Left:* Objects represented as graphical sets. *Center:* Objects embedded in a background to form a graphical image. *Right:* Object and background are digitized to form a digital image (note the grid). Second row: Example of a structuring element represented as a set, a graphical image, and finally as a digital SE.

Reflection and translation

The *reflection* of a set (structuring element) B about its origin, denoted by $\hat{B}$, is defined as

$$\hat{B} = \{ w \mid w = -b, \text{ for } b \in B \}$$

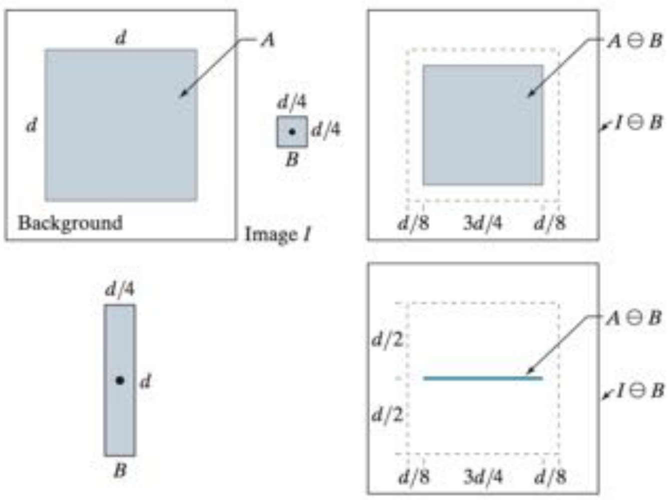
The *translation* of a set B by point $z = (z_1, z_2)$, denoted $(B)_z$, is defined as

$$(B)_z = \{ c \mid c = b + z, \text{ for } b \in B \}$$

Example of erosion

a b c
d e

FIGURE 9.4
(a) Image I, consisting of a set (object) A, and background.
(b) Square SE, B (the dot is the origin).
(c) Erosion of A by B (shown shaded in the resulting image).
(d) Elongated SE.
(e) Erosion of A by B. (The erosion is a line.) The dotted border in (c) and (e) is the boundary of A, shown for reference.



Erosion – formal definition

With A and B as sets in Z^2, the *erosion* of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \qquad (9\text{-}3)$$

→ B has to be contained in A
→ Equivalent to B not sharing any common elements with the background (i.e., the set complement of A)

$$A \ominus B = \{z \mid (B)_z \cap A^c = \varnothing\}$$

For the whole image

$$I \ominus B = \{z \mid (B)_z \subseteq A \text{ and } A \subseteq I\} \cup \{A^c \mid A^c \subseteq I\} \qquad (9\text{-}4)$$

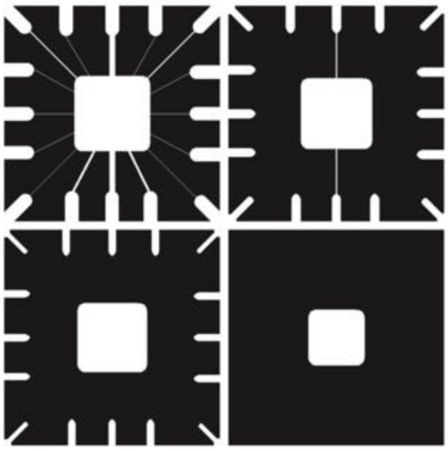
where I is a rectangular array of foreground and background pixels. The contents of

Example: Using erosion to remove image components

Erosion shrinks or thins objects in a binary image
→ Erosion as a morphological filtering operation in which image details smaller than the structuring element are filtered (removed) from the image

a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486×486 binary image of a wire-bond mask in which foreground pixels are shown in white. (b)–(d) Image eroded using square structuring elements of sizes 11×11, 15×15, and 45×45 elements, respectively, all valued 1.

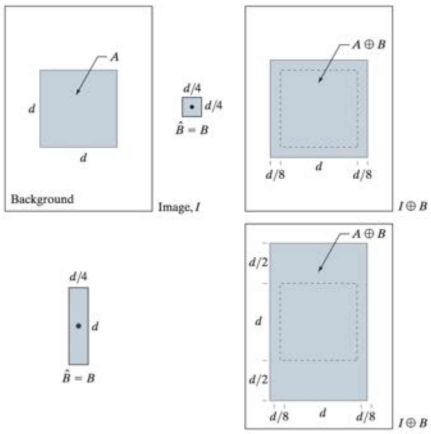


Dilation example

Erosion = shrinking or thinning operation
Dilation ”grows” or “thickens” objects

a b c
d e

FIGURE 9.6
(a) Image I, composed of set (object) A and background.
(b) Square SE (the dot is the origin).
(c) Dilation of A by B (shown shaded).
(d) Elongated SE.
(e) Dilation of A by this element. The dotted line in (c) and (e) is the boundary of A, shown for reference.



Dilation – formal definition

With A and B as sets in Z^2, the dilation of A by B, denoted as $A \oplus B$, is defined as

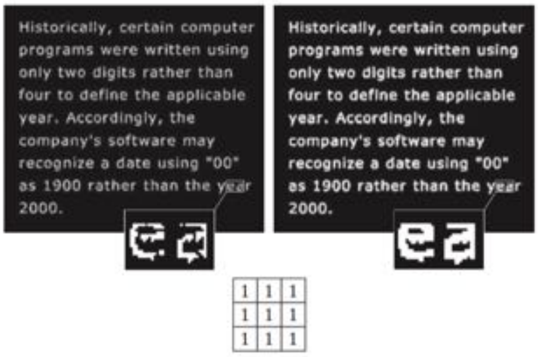
$$A \oplus B = \left\{ z \,\middle|\, (\hat{B})_z \cap A \neq \varnothing \right\} \qquad (9\text{-}6)$$

→ The dilation of A by B is the set of all displacements, z, such that the foreground elements of B-hat overlap at least one element of A

$$A \oplus B = \left\{ z \,\middle|\, [(\hat{B})_z \cap A] \subseteq A \right\}$$

Example: Using dilation to repair broken characters

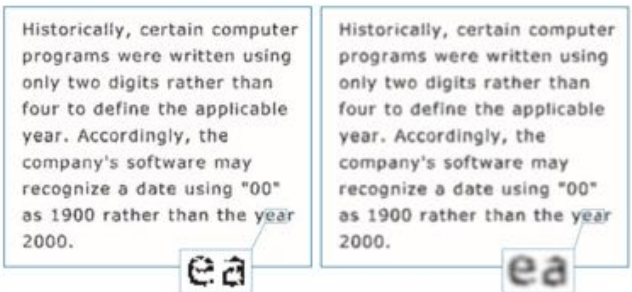
a c
b
FIGURE 9.7
(a) Low-resolution text showing broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.



Bridging gaps comparing with spatial filtering

Lowpass filtering: starts with a binary image and produces a grayscale image
Dilation: results directly in a binary image

a b
FIGURE 4.48
(a) Sample text of low resolution (note the broken characters in the magnified view).
(b) Result of filtering with a GLPF, showing that gaps in the broken characters were joined.



Duality of Erosion and Dilation

Erosion and dilation are *duals* of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B} \qquad (9\text{-}8)$$

and

$$(A \oplus B)^c = A^c \ominus \hat{B} \qquad (9\text{-}9)$$

Opening

- Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions

The *opening* of set A by structuring element B, denoted by $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B \qquad (9\text{-}10)$$

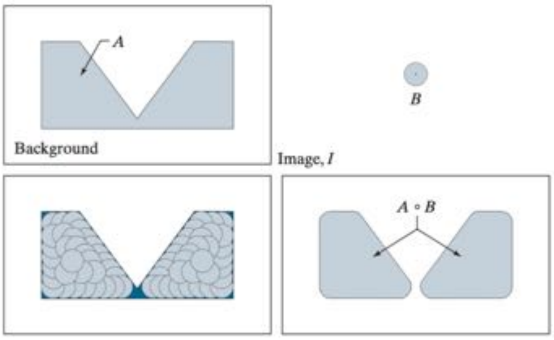
→ The opening of A by B is the union of all the translations of B so that B fits entirely in A

$$A \circ B = \bigcup \left\{ (B)_z \middle| (B)_z \subseteq A \right\}$$

Geometric interpretation of Opening

a b
c d

FIGURE 9.8
(a) Image I, composed of set (object) A and background.
(b) Structuring element, B.
(c) Translations of B while being contained in A. (A is shown dark for clarity.)
(d) Opening of A by B.



Opening → eliminate regions narrower than the structuring element

Closing

- Closing tends to smooth sections of contours, but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour

Similarly, the *closing* of set A by structuring element B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B \qquad (9\text{-}11)$$

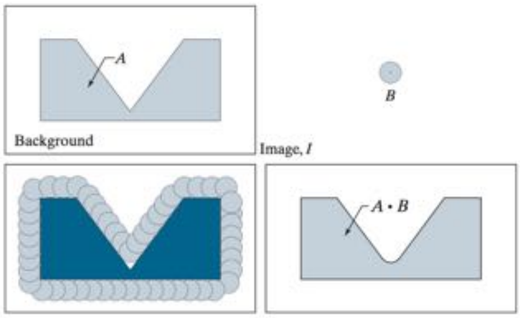
→ Complement of the union of all translations of B that do not overlap A

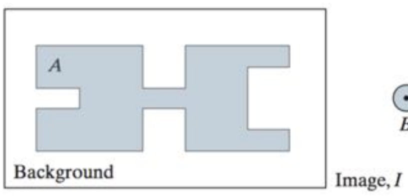
$$A \bullet B = \left[\bigcup \left\{ (B)_z \,\middle|\, (B)_z \cap A = \varnothing \right\} \right]^c$$

Geometric interpretation of Closing

a b
c d

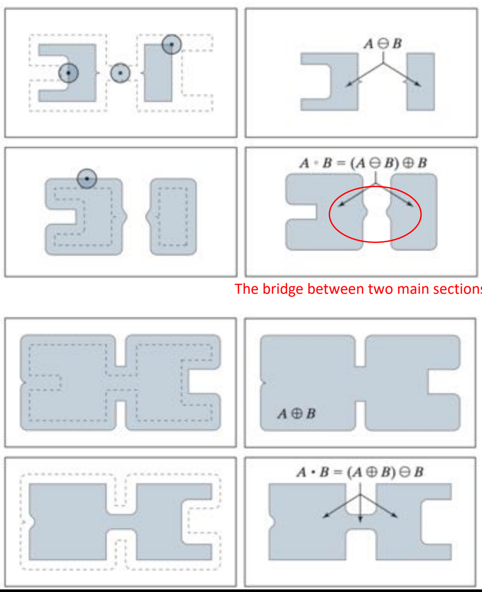
FIGURE 9.9
(a) Image I, composed of set (object) A, and background.
(b) Structuring element B.
(c) Translations of B such that B does not overlap any part of A. (A is shown dark for clarity.)
(d) Closing of A by B.





a
b c
d e
f g
h i

FIGURE 9.10
Morphological opening and closing.
(a) Image I, composed of a set (object) A and background; a solid, circular structuring element is shown also. (The dot is the origin.)
(b) Structuring element in various positions.
(c)-(i) The morphological operations used to obtain the opening and closing.



The bridge between two main sections was eliminated

Opening & Closing duality

As with erosion and dilation, opening and closing are duals of each other with respect to set complementation and reflection:

$$(A \circ B)^c = \left(A^c \bullet \hat{B}\right) \quad (9\text{-}14)$$

and

$$(A \bullet B)^c = \left(A^c \circ \hat{B}\right) \quad (9\text{-}15)$$

Properties of Opening and closing

Morphological opening

(a) $A \circ B$ is a subset of A.
(b) If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.
(c) $(A \circ B) \circ B = A \circ B$.

Multiple openings or closings of a set have no effect after the operation has been applied once.

Similarly, closing satisfies the following properties:

(a) A is a subset of $A \bullet B$.
(b) If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.
(c) $(A \bullet B) \bullet B = A \bullet B$.

Opening and closing as morphological filtering

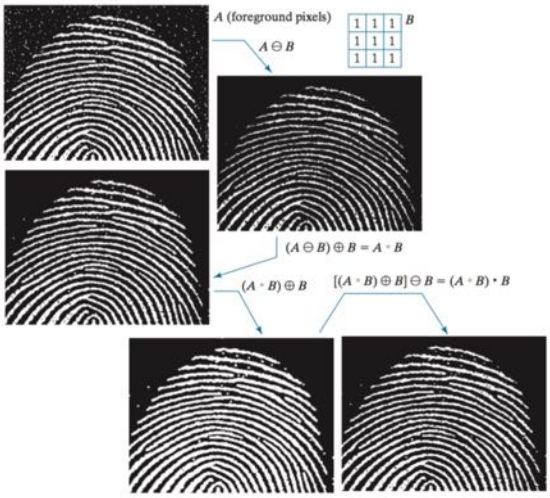
Eroding → remove white specks, but increase size of dark spots

Dilation → new gaps between the fingerprint ridges

Dilation → store breaks but ridges are thickened

Erosion → remedy thickened ridges

Overall: only some white specks



Hit-or-miss transform (HMT)

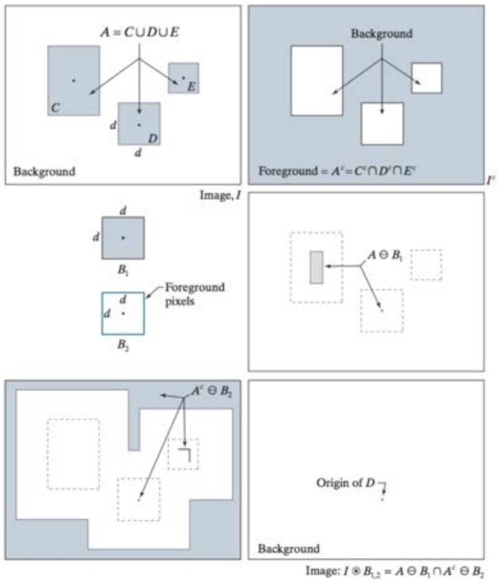
- Basic tool for shape detection
- Using two structuring elements

B_1, for detecting shapes in the foreground
B_2, for detecting shapes in the background

$$I \circledast B_{1,2} = \left\{ z \,\middle|\, (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c \right\}$$
$$= (A \ominus B_1) \cap (A^c \ominus B_2)$$

→ *Simultaneously* B_1 find a match in the foreground and B2 find a match in the background

HMT example



a b
c d
e f

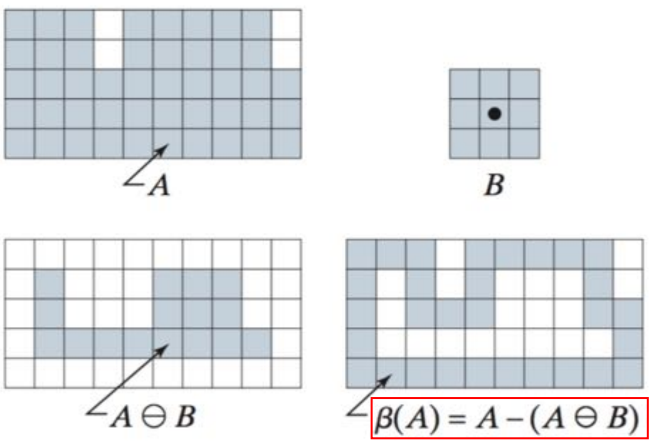
FIGURE 9.12 (a) Image consisting of a foreground (1's) equal to the union, A, of set of objects, and a background of 0's. (b) Image with its foreground defined as A^c. (c) Structuring elements designed to detect object D. (d) Erosion of A by B_1. (e) Erosion of A^c by B_2. (f) Intersection of (d) and (e), showing the location of the origin of D, as desired. The dots indicate the origin of their respective components. Each dot is a single pixel.

Morphological algorithms

- Extract boundaries
- Fill holes
- Find connected components

Foreground boundary

Boundary extraction algorithm

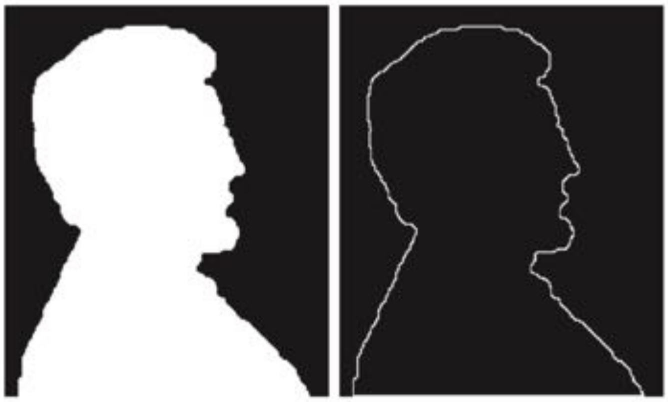


Reference: Gonzalez et. al., Fig. 9.15

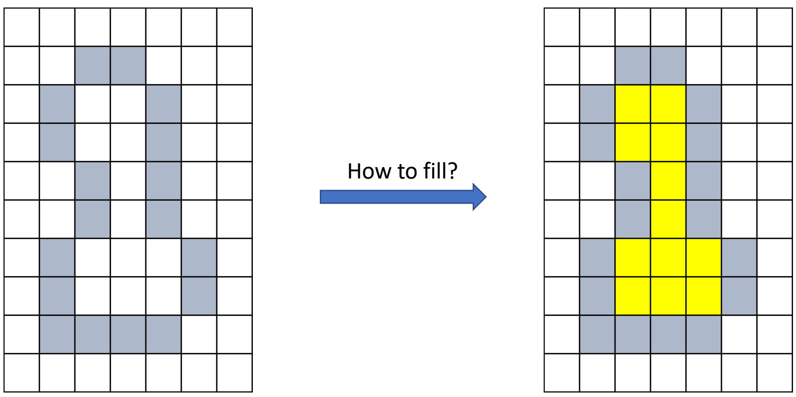
Boundary extraction example

a b

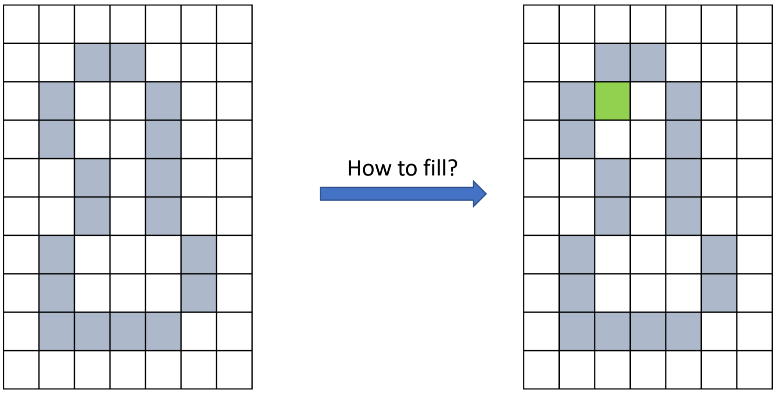
FIGURE 9.16
(a) A binary image.
(b) Result of using Eq. (9-18) with the structuring element in Fig. 9.15(b).



Hole filling

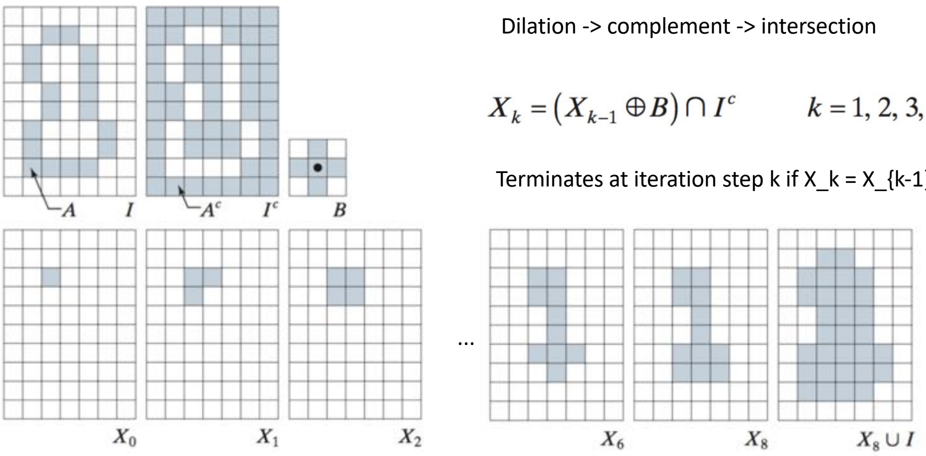


Hole filling



Given that we know a point in the hole

Hole filling



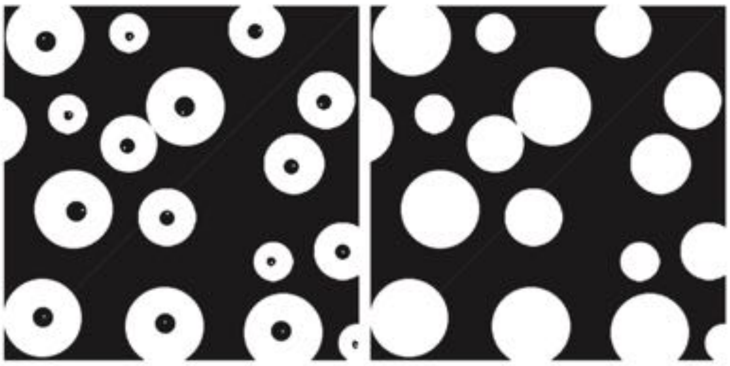
Dilation -> complement -> intersection

$$X_k = (X_{k-1} \oplus B) \cap I^c \qquad k = 1, 2, 3, \ldots$$

Terminates at iteration step k if X_k = X_{k-1}

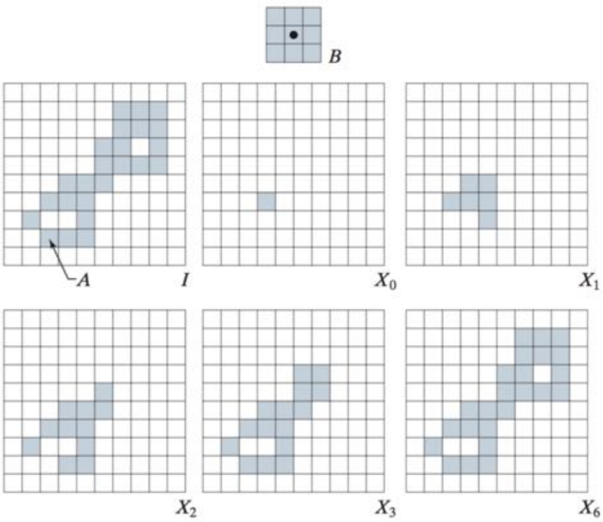
Ref: Gonzalez et.al., Fig. 9.17

Hole filling example



a b
FIGURE 9.18
(a) Binary image. The white dots inside the regions (shown enlarged for clarity) are the starting points for the hole-filling algorithm.
(b) Result of filling all holes.

Connected components



Dilation -> intersection

$$X_k = \left(X_{k-1} \oplus B\right) \cap I$$

Ref: Gonzalez et.al., Fig. 9.19

Convex hull

Let B^i, $i = 1, 2, 3, 4$, denote the four structuring elements in Fig. 9.21(a). The procedure consists of implementing the morphological equation

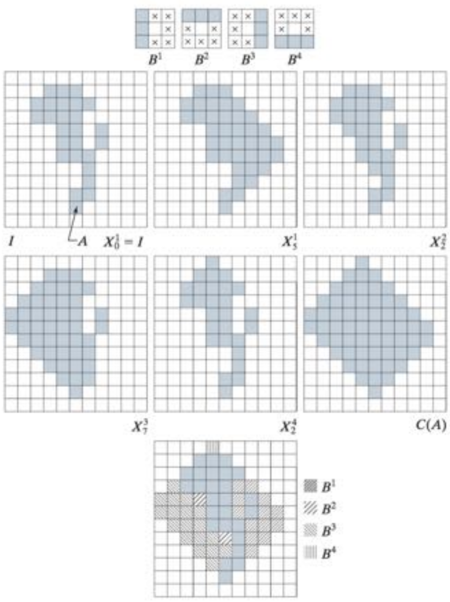
$$X_k^i = \left(X_{k-1}^i \circledast B^i\right) \cup X_{k-1}^i \quad i = 1, 2, 3, 4 \quad \text{and} \quad k = 1, 2, 3, \ldots \qquad (9\text{-}21)$$

with $X_0^i = I$. When the procedure converges using the ith structuring element (i.e., when $X_k^i = X_{k-1}^i$), we let $D^i = X_k^i$. Then, the convex hull of A is the union of the four results:

$$C(A) = \bigcup_{i=1}^{4} D^i \qquad (9\text{-}22)$$

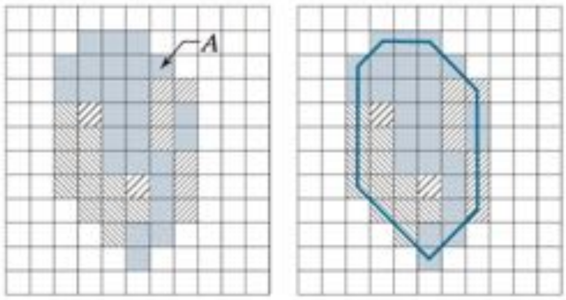
a
b c d
e f g
h

FIGURE 9.21
(a) Structuring elements.
(b) Set A.
(c)–(f) Results of convergence with the structuring elements shown in (a).
(g) Convex hull.
(h) Convex hull showing the contribution of each structuring element.



a b

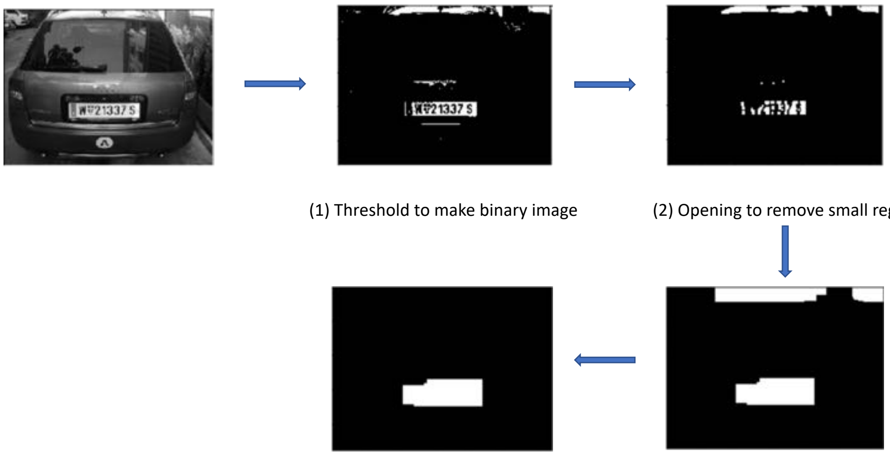
FIGURE 9.22
(a) Result of limiting growth of the convex hull algorithm.
(b) Straight lines connecting the boundary points show that the new set is convex also.



Other algorithms

- Top hat
- Black hat
- Morphological reconstruction
  - Geodesic dilation
  - Geodesic erosion
- Morphology for grayscale images

License plate detection using morphology



(1) Threshold to make binary image

(2) Opening to remove small regions

(3) Dilation to connect close regions

(4) Connected components to detect highest region

Homework 2

Homework 2: Detect barcode

- Description: Detect barcode region in an image

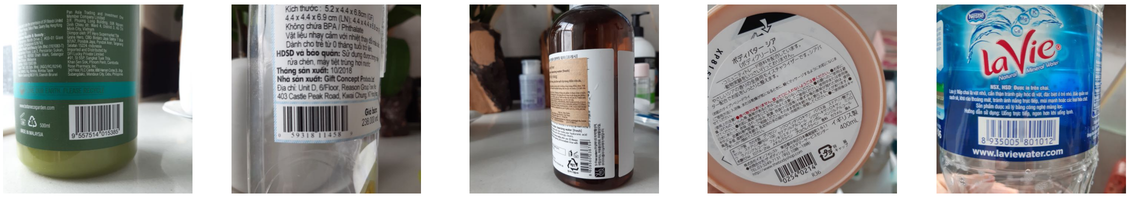
  There're 5 images that have barcode in each image.

  The barcodes vary in size, color, orientation, and slightly, shape.

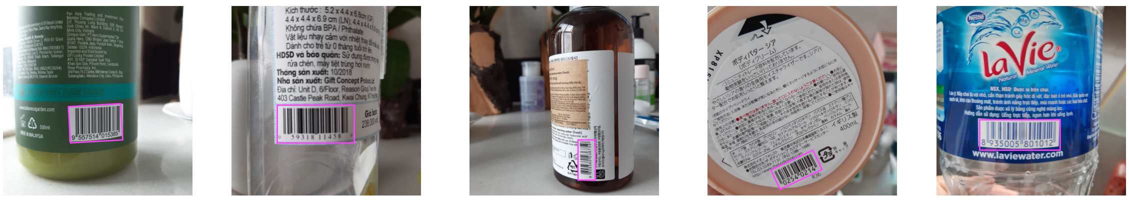
  Build a script using OpenCV and Python to detect the barcode region.
- Requirements:
  - Submit the source code that includes detecting + displaying the region in the original data

Data

Input:



Expected output:



Note: Overlapping with the groundtruth about 80% is acceptable!

Submission

- Form: https://forms.gle/ftoUvATYeEGj55KY7
- Deadline: Oct 13, 2019, 23:59 (Hanoi time)