

Image Processing

INT3404 20

Week 6: Morphological Operations

Lecturer: Nguyen Thi Ngoc Diep, Ph.D.

Email: ngocdiep@vnu.edu.vn

Slide & code: https://github.com/chupibk/INT3404_20

1

Schedule

Week	Content	Homework
1	Introduction	Set up environments: Python 3, OpenCV 3, Numpy, Jupyter Notebook
2	Digital image – Point operations Contrast adjust – Combining images	HW1: adjust gamma to find the best contrast
3	Histogram - Histogram equalization – Histogram-based image classification	Self-study
4	Spatial filtering - Template matching	Self-study
5	Feature extraction Edge, Line, and Texture	Self-study
6	Morphological operations	HW2: Barcode detection → Require submission as mid-term test
7	Filtering in the Frequency domain Announcement of Final project topics	Final project registration
8	Color image processing	HW3: Conversion between color spaces, color image segmentation
9	Geometric transformations	Self-study
10	Noise and restoration	Self-study
11	Compression	Self-study
12	Final project presentation	Self-study
13	Final project presentation Class summarization	Self-study

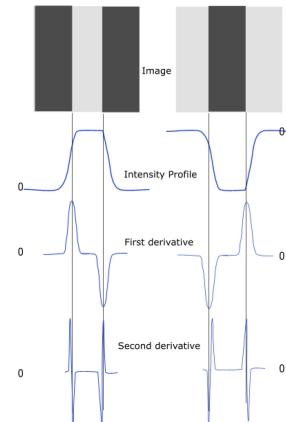
2

2

Recall week 5: Edge detection using derivatives

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

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



3

Recall week 5: Edge detection filters

	Column mask	Row mask
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Robert	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Laplace	$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix}$

4

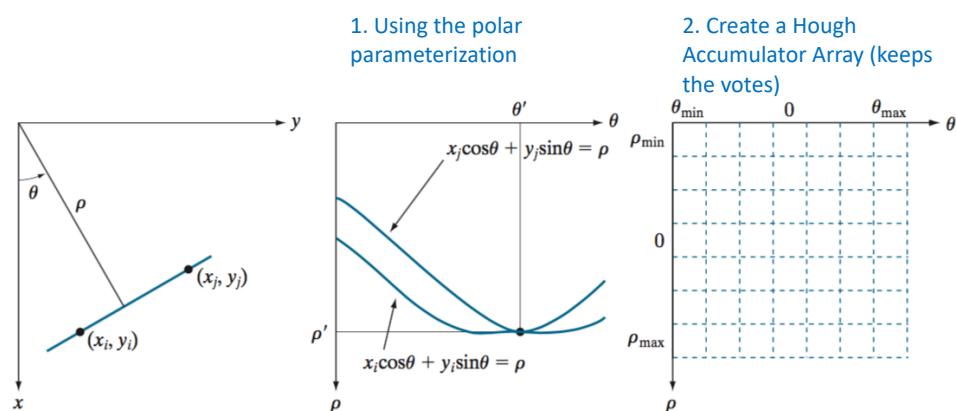
Recall week 5: Canny edge detection

1. Noise reduction;
2. Gradient calculation;
3. Non-maximum suppression;
4. Double threshold;
5. Edge Tracking by Hysteresis.



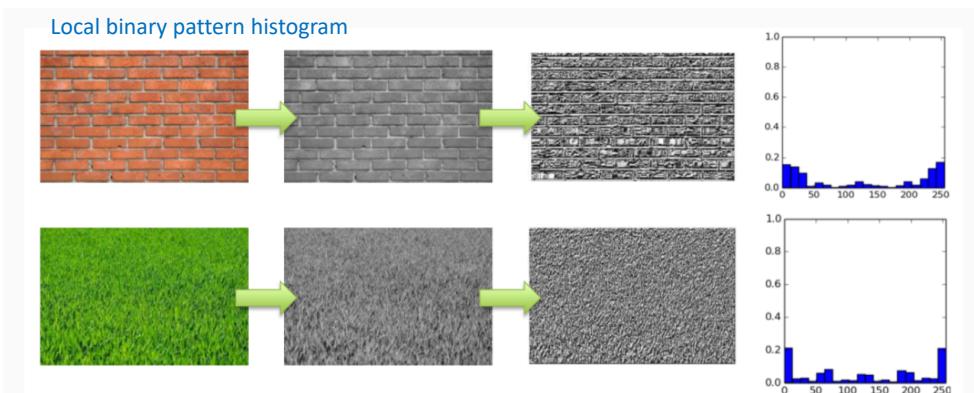
5

Recall week 5: Line detection with Hough transform



6

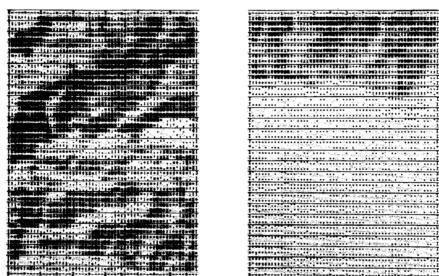
Recall week 5: Texture analysis



7

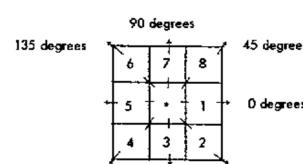
Recall week 5: Texture analysis

Gray Level Co-occurrence Matrix (GLCM)



Grassland			Water Body			
Angle	ASM	Contrast	ASM	Contrast	Correlation	
0°	.0128	3.048	.8075	.1016	2.153	.7254
45°	.0080	4.011	.6366	.0771	3.057	.4768
90°	.0077	4.014	.5987	.0762	3.113	.4646
135°	.0064	4.707	.4610	.0741	3.129	.4650
Avg.	.0087	3.945	.6297	.0822	2.863	.5327

Fig. 4. Textural features for two different land-use category images.



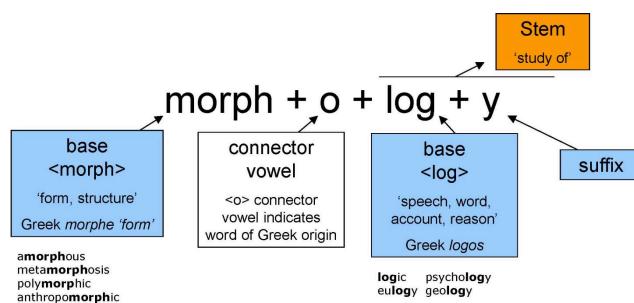
8

Week 6: Morphological operations

Erosion, Dilation, Opening, Closing

9

Morphology



→ Study of form and structures

- In image processing: Study of image components that are useful in the representation and description of region shape
- Eg: boundaries, skeletons, convex hull
- Operations: filtering, thinning, pruning

Illustration credit: <https://educ.queensu.ca/research/spotlights/morphology>

10

Set theory for morphology

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

11

Represent image as a set

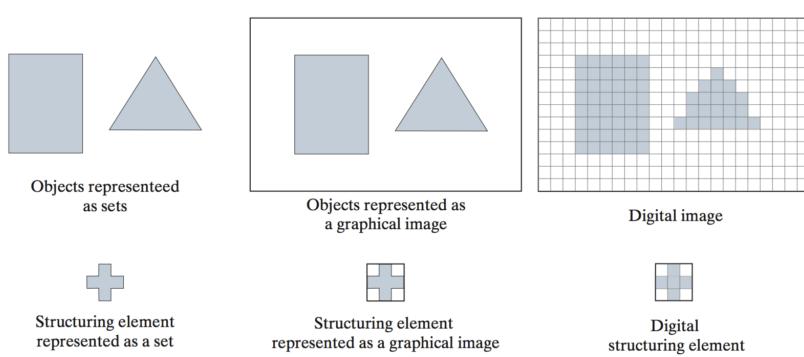


FIGURE 9.1 Top row. *Left:* Objects represented as graphical sets. *Center:* Objects embedded in a background to form a graphical image. *Right:* Object and background are digitized to form a digital image (note the grid). Second row: Example of a structuring element represented as a set, a graphical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

12

Reflection

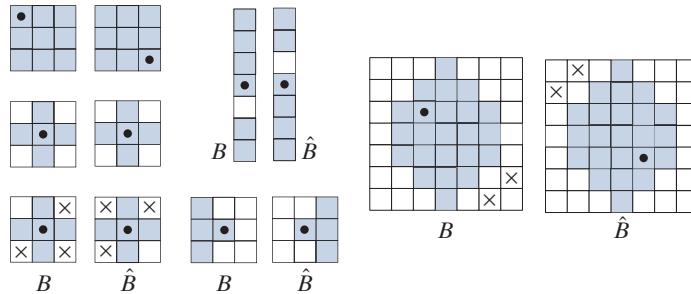
The *reflection* of a set (structuring element) B about its origin, denoted by \hat{B} , is defined as

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

→ (x, y) is replaced by $(-x, -y)$

FIGURE 9.2

Structuring elements and their reflections about the origin (\times 's are don't care elements, and the dots denote the origin). Reflection is rotation by 180° of an SE about its origin.



13

Translation

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

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

(x, y) is replaced by $(x+z_1, y+z_2)$

→ Is used to translate (slide) a structuring element over an image, at each location perform a set operation between the structuring element and the area of the image directly under it

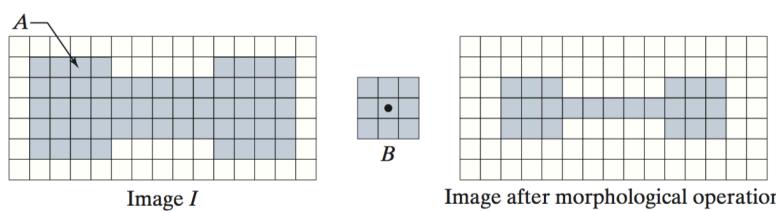


Fig. 9.3

14

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

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

→ 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)

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

For the whole image

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

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

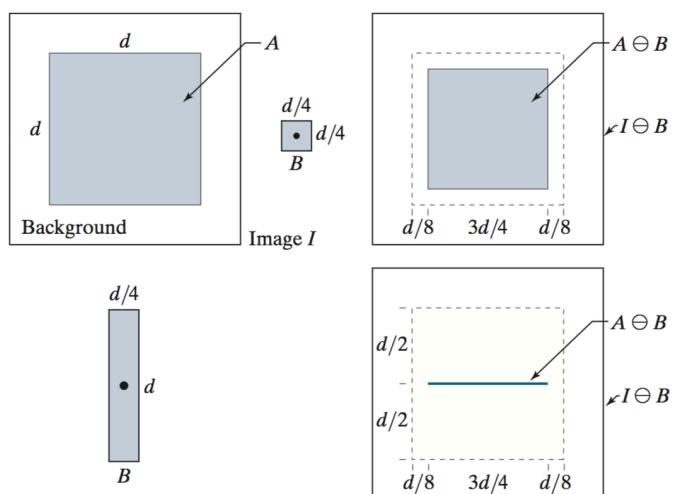
15

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.



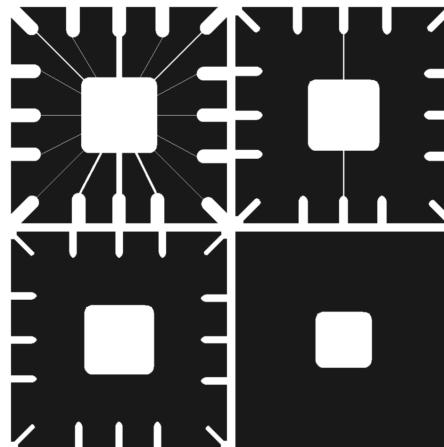
16

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.



17

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

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (9-6)$$

- Reflecting B about its origin and translating the reflection by z
- The set of all displacements, z , such that the foreground elements of \hat{B} overlap at least one element of A

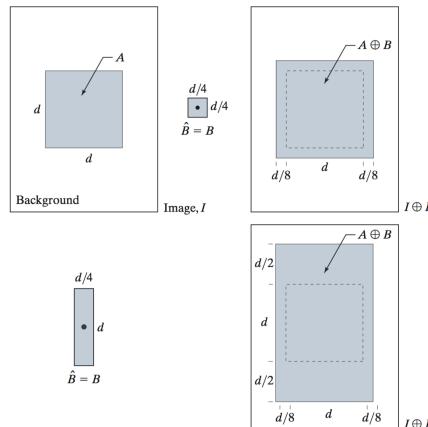
$$A \oplus B = \left\{ z \mid [(\hat{B})_z \cap A] \subseteq A \right\}$$

18

Dilation example

Erosion = shrinking or thinning operation
 Dilation "grows" or "thickens" objects

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.



19

Example: Using dilation to repair broken characters

One of the simplest applications of dilation is for bridging gaps

a b c

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.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



1	1	1
1	1	1
1	1	1

20

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.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



21

Duality of Erosion and Dilation

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

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

and

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

22

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

$$A \circ B = (A \ominus B) \oplus B \quad (9-10)$$

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

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

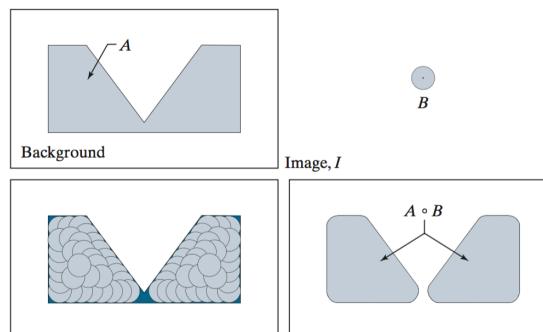
24

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

25

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

$$A \bullet B = (A \oplus B) \ominus B \quad (9-11)$$

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

$$A \bullet B = \left[\bigcup \{(B)_z \mid (B)_z \cap A = \emptyset\} \right]^c$$

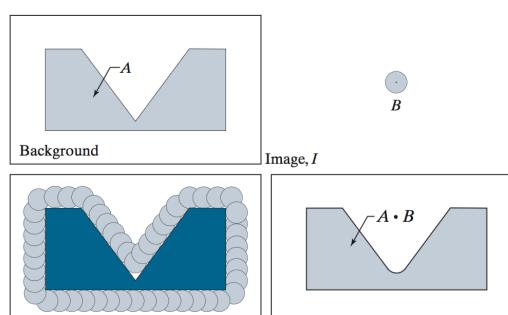
26

Geometric interpretation of Closing

a	b
c	d

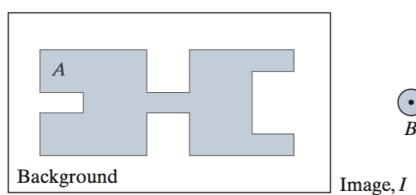
FIGURE 9.9

- (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 .



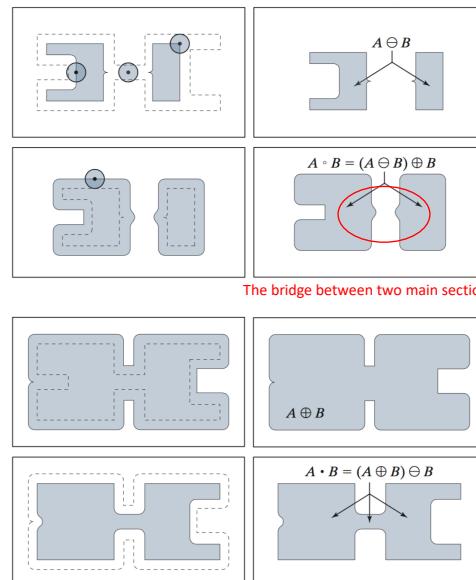
27

Opening vs closing



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.



28

Opening & Closing duality

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

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

and

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

29

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.

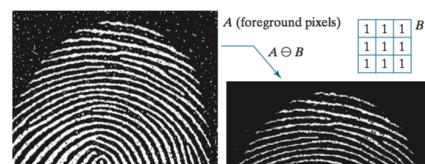
Similarly, closing satisfies the following properties:

- (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) $(A \bullet B) \bullet B = A \bullet B$.

30

Opening and closing as morphological filtering

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



Dilation → new gaps between the fingerprint ridges



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

Dilation → store breaks but ridges are thickened



Erosion → remedy thickened ridges

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

Overall: only some white specks

31

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

$$\begin{aligned} I \circledast B_{1,2} &= \left\{ z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c \right\} \\ &= (A \ominus B_1) \cap (A^c \ominus B_2) \end{aligned}$$

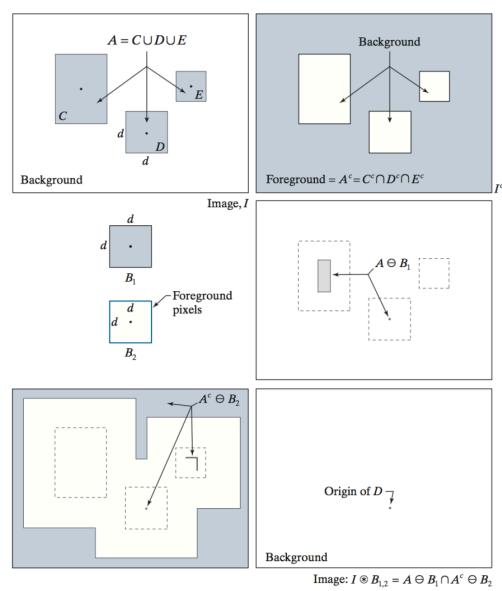
→ Simultaneously B_1 find a match in the foreground and B_2 find a match in the background

32

HMT example

$$\begin{aligned} I \circledast B_{1,2} &= \left\{ z \mid (B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c \right\} \\ &= (A \ominus B_1) \cap (A^c \ominus B_2) \end{aligned}$$

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.



33

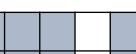
Morphological algorithms

- Extract boundaries
 - Fill holes
 - Find connected components

34

Foreground boundary

Binary object

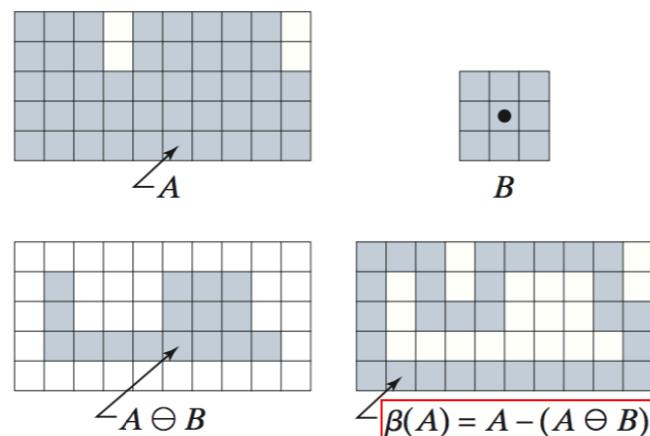


A 5x5 grid of cells. All cells are gray except for one, which is white. The white cell is located at the third row and second column from the top-left corner.

How to extract?

35

Boundary extraction algorithm



Reference: Gonzalez et. al., Fig. 9.15

36

Boundary extraction example

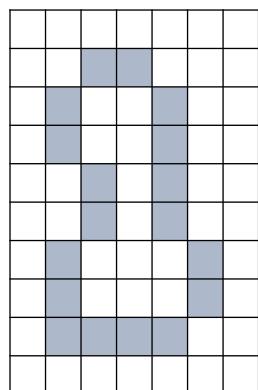
a b

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

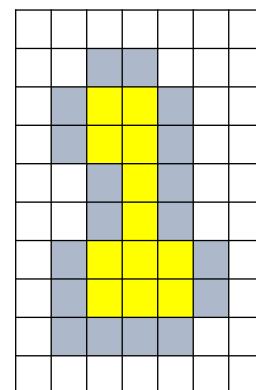


37

Hole filling

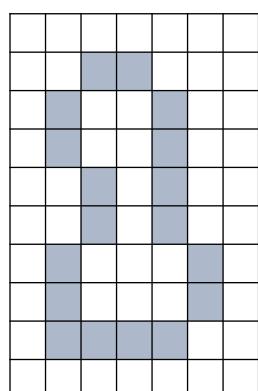


How to fill?

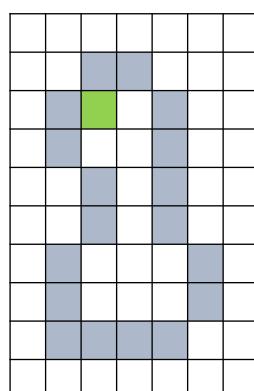


38

Hole filling



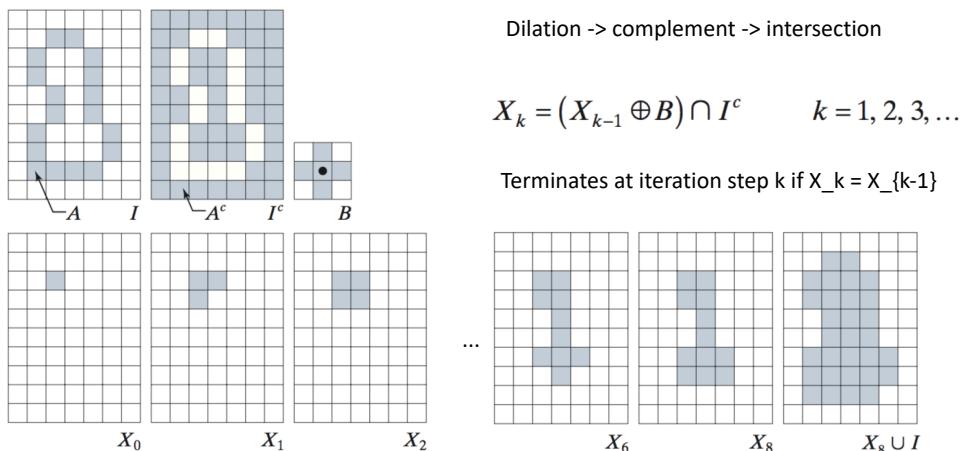
How to fill?



Given that we
know a point in
the hole

39

Hole filling



Ref: Gonzalez et.al., Fig. 9.17

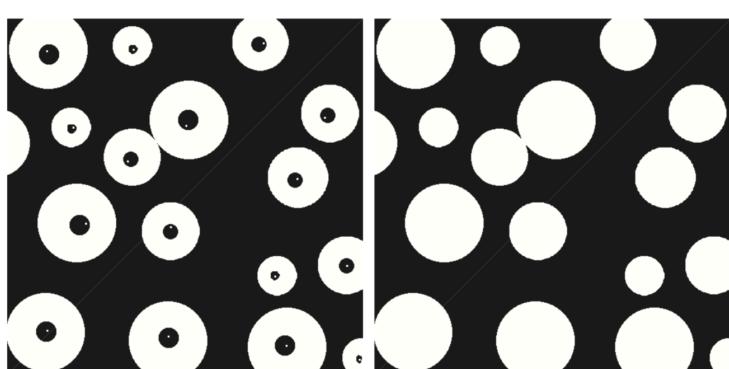
40

Hole filling example

a b

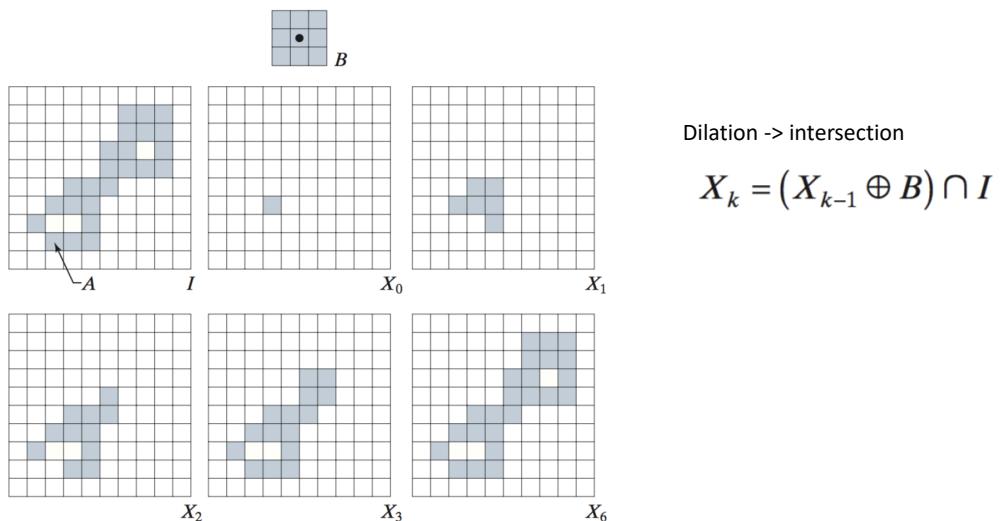
FIGURE 9.18

(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.



41

Connected components



Ref: Gonzalez et.al., Fig. 9.19

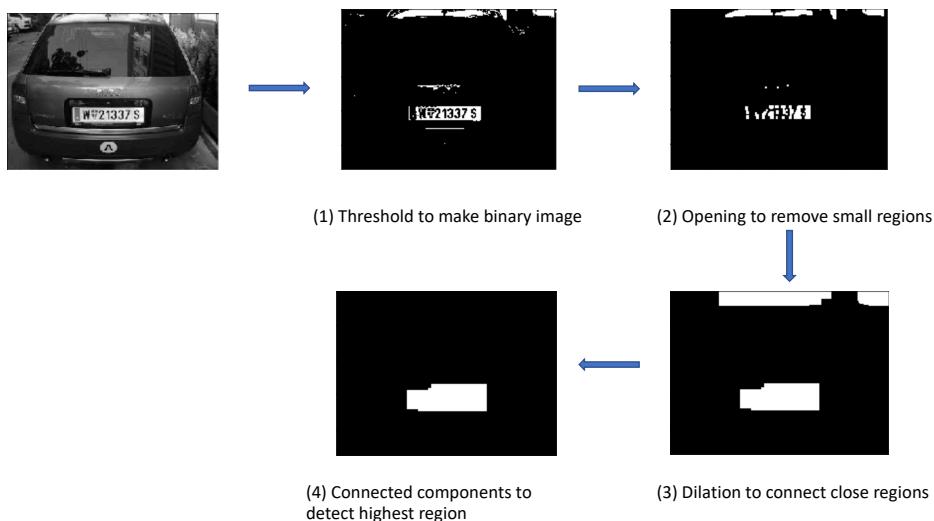
42

Other algorithms

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

43

License plate detection using morphology



47

Homework : Detect QR code

- Description: Detect QR code region in an image and build a heatmap for detected region

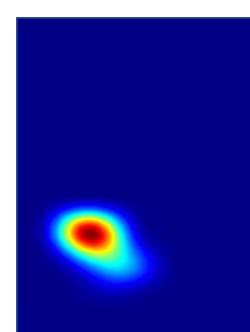
Input:



Detected QR code



QR heatmap



Note: Overlapping with the groundtruth about 80% is acceptable! IOU>0.8

48

Submission requirements

- Requirements:
 - Submit the source code, output image of detected rectangle, heatmap image for the detected region
- Submission:
 - Via Teams (or Google Drive) → announce later
 - Deadline: Sunday April 19, 2020, 23:59 (Hanoi time)
- Note: difference between *boundingRect()* and *minAreaRect()*