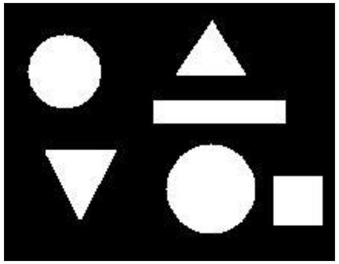
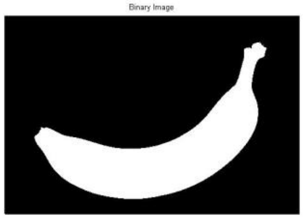
ẢNH NHỊ PHÂN

BINARY IMAGE PROCESSING





Nội dung

Giới thiệu ảnh nhị phân

Các phép Morphology cơ bản

Nhắc lại một số phép toán trên tập hợp

Các phép Morphology tổng quát

- Erosion and dilation
- Opening and closing
- Hit-or-miss, boundary extraction, …

Skeleton via distance transform

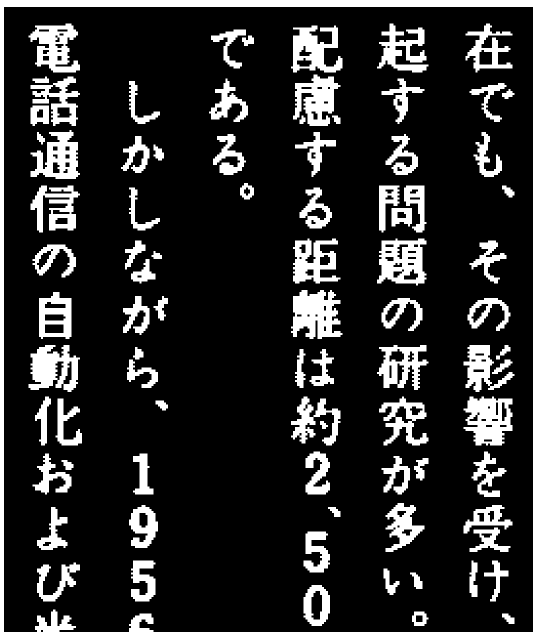
Binary Images

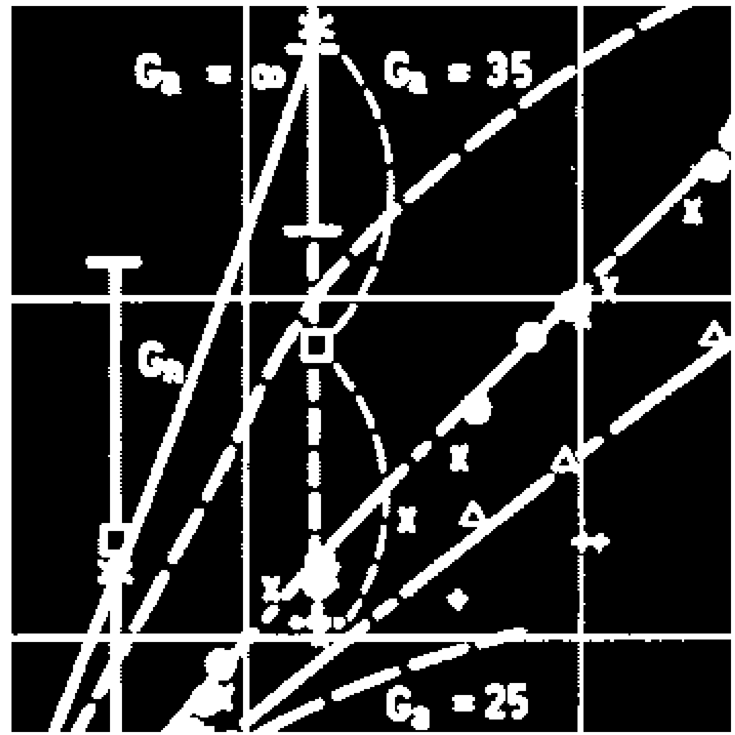
Images only consist of two colors (tones): white or black

Numerical example (image of a square block)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	255	255	255	255	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Binary Images


在でも、その影響を受け、
起する問題の研究が多い。
配慮する距離は約2、50
である。
しかしながら、195
電話通信の自動化および



Why are binary images special?

Since pixels are either white or black, the locations of white (black) pixels carry ALL information of binary images

Example

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{matrix}$$

f(m,n)

$\Longleftrightarrow$

L={(3,3),(3,4),(4,3),(4,4)}

$\downarrow$

location of white pixels

matrix representation

set representation

It is often more convenient to consider the set representation than the matrix representation for binary images

Why are binary images special?

Representation of individual pixels as 0 or 1, convention:

- foreground, object = 1 (white)
- background = 0 (black)

Processing by logical functions is fast and simple

Nội dung

- Giới thiệu ảnh nhị phân
- Các phép Morphology cơ bản
- Nhắc lại một số phép toán trên tập hợp
- Các phép Morphology tổng quát
  - Erosion and dilation
  - Opening and closing
  - Hit-or-miss, boundary extraction, …
- Skeleton via distance transform

Phép biến đổi hit-or-miss (trúng hay trật)

Có 3 thông số:

- Kích thước cửa sổ
- hit table: lưu các mẫu so khớp
- Quy tắc, ví dụ như quy tắc chuyển trạng thái của 1 pixel

Ý tưởng : cho cửa sổ dịch chuyển trên ảnh, nếu giá trị của nó thuộc hit table thì áp dụng “quy tắc” lên pixel trung tâm.

Cho trước vị trí (x,y) trong ảnh nguồn, ta nói một hit xảy ra nếu cửa sổ có tâm (x,y) khớp với 1 trong các mẫu của hit table.

Phép biến đổi hit-or-miss (trúng hay trật)

Ví dụ: xác định 3 thông số:

- Cửa sổ 3x3
- Hit table: có 2 mẫu sau

1	0	0
0	1	0
0	0	0

0	0	1
0	1	0
0	0	0

- Quy tắc: nếu 1 hit nảy sinh thì gán p(x,y) về zero (0)

Phép biến đổi hit-or-miss (trúng hay trật)

Với ảnh sau:

		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		
	0					0	

Phép co (Erosion)

Dùng để gọt bớt lớp pixel ngoài cùng của đối tượng có trong ảnh.

*Quy tắc: nếu pixel trung tâm là 1 và có ít nhất 1 pixel lận cận là 0 (nhưng không đồng thời bằng 0, điều này để đảm bảo không xóa mất đối tượng khi cần) thì gán pixel trung tâm về 0.

Ví dụ: với ảnh cho trên, các pixel 1 ngoài biên được gán về 0.

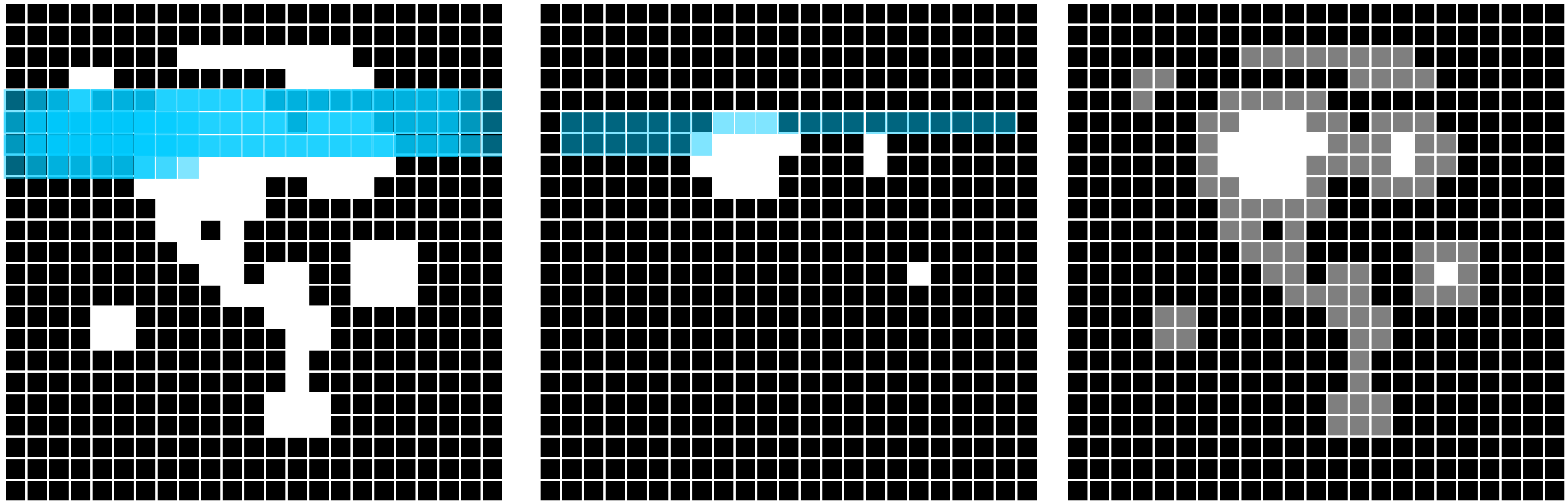
Phép co (Erosion)

Với ảnh sau:

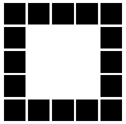
		1	1	1	1		
		1	1	1	1		
		1	1	1	1		
		1	1	1	1		
	1					1	

		0	0	0	0		
		0	1	1	0		
		0	1	1	0		
		0	0	0	0		
	0					0	

Phép co (Erosion)



difference

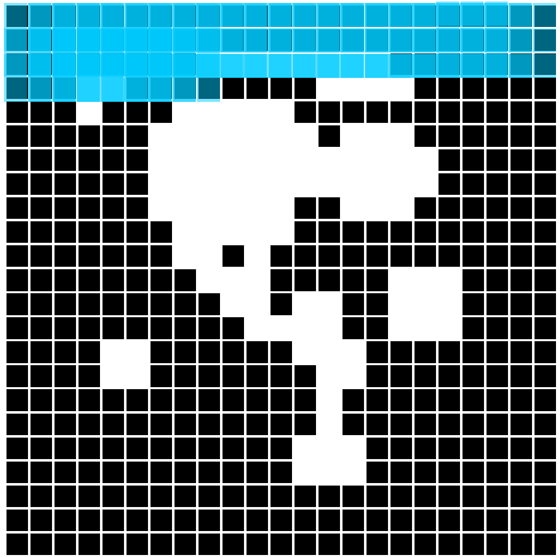


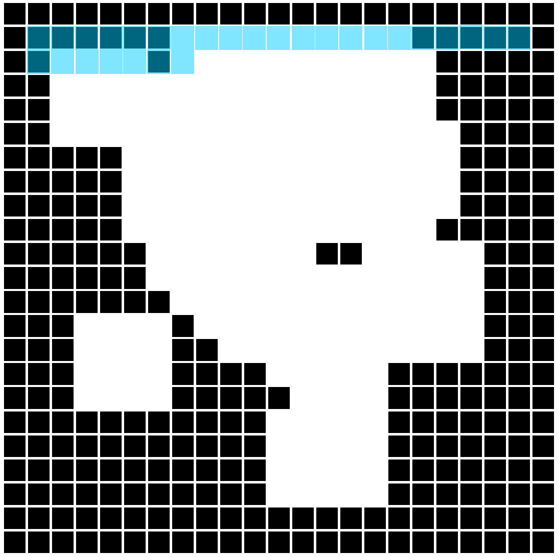
Phép giãn (Dilation)

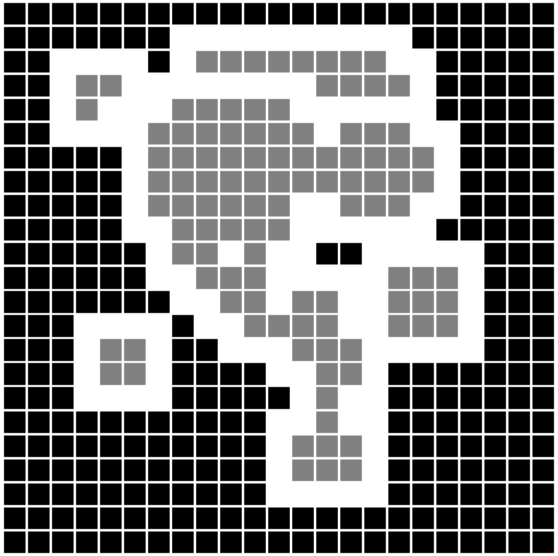
* Là thao tác ngược với phép co: khi thêm 1 lớp pixel có chiều rộng là 1 vào biên đối tượng.

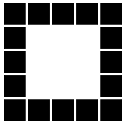
* Quy tắc: nếu pixel trung tâm là 0 và có ít nhất 1 pixel lận cận là 1 thì pixel trung tâm được gán về 1.

Phép giãn (Dilation)









Phép mở (Opening

* Là phép co sau đó là phép giãn

* Có tác dụng làm trơn ảnh và loại nhiễu.

*Lưu ý: dù 2 phép biến đổi là trái ngược nhau nhưng nó không trở về trạng thái ban đầu, có khả năng xảy ra đứt nét.

Phép đóng (Closing)

* Là phép giãn sau đó là phép co.

* Có thể dùng để lấp lổ hỏng trong ảnh.

Phép loại bỏ pixel nội

Gán pixel trung tâm về 0 nếu cửa sổ có dạng sau :

X	1	X
1	1	1
X	1	X

trong đó x là vị trí của pixel mà ta không cần quan tâm.

Cài đặt

1. Phương pháp thông thường:

Quét ảnh và so khớp từng lân cận 3x3 với tất cả các mẫu trong hit table.

X_3	X_2	X_1
X_4	X	X_0
X_5	X_6	X_7

2. Phương pháp dồn pixel:

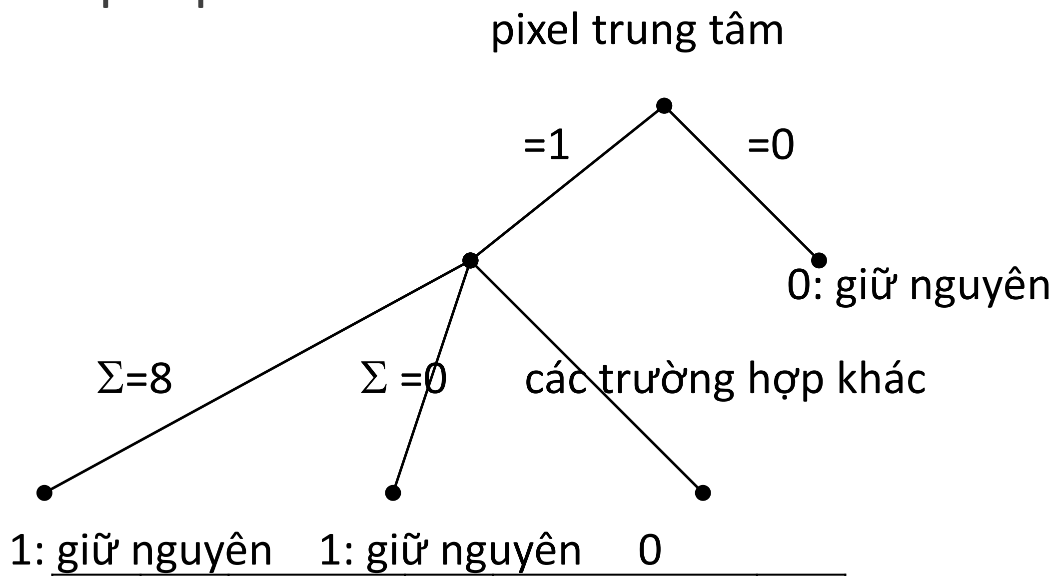
- Dồn pixel 3x3 về 1 số 9 bits.

X	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7

- Sau đó tạo bảng look-up sao cho mọi phần tử so khớp với mẫu sẽ thay đổi theo qui tắc tương ứng.

Cài đặt

Ví dụ: Tạo bảng look-up cho phép co

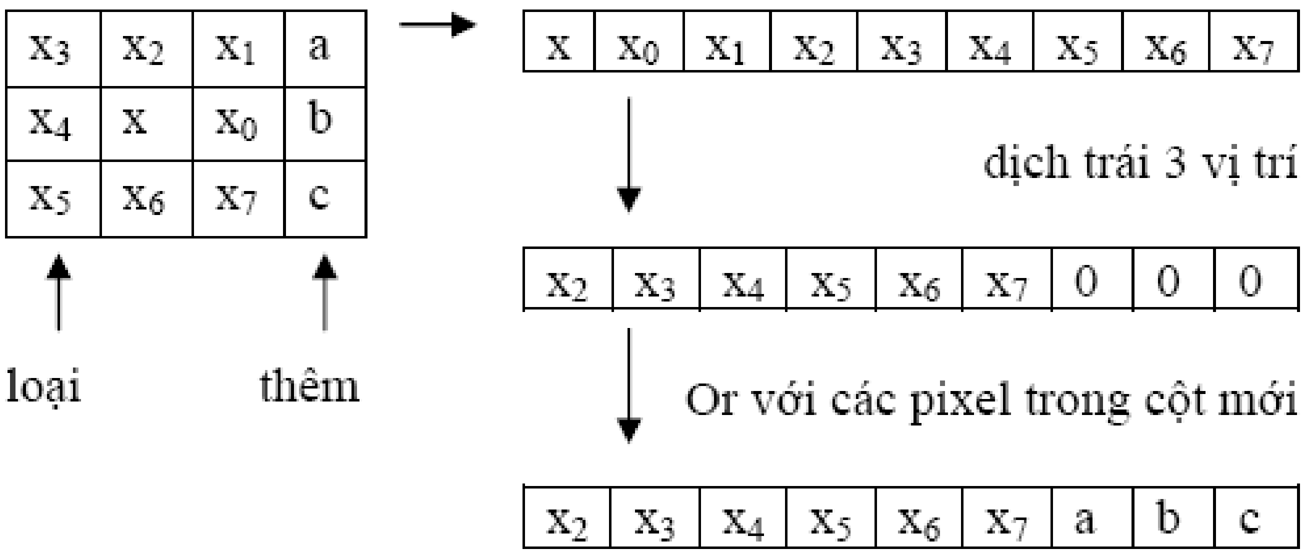


0	1		256		511
			1		1

Nhận xét: Vì bảng look-up trên chỉ có 2 phần tử bằng 1, các phần tử khác còn lại bằng 0 nên không cần dùng bảng.

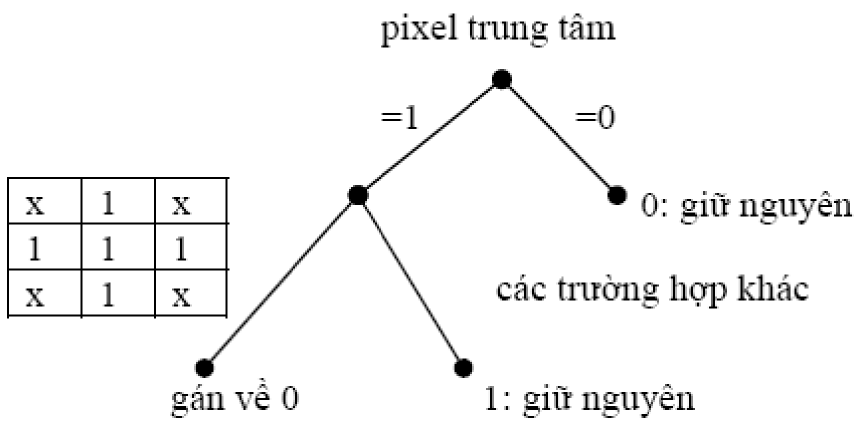
Cài đặt

* Để tăng tốc độ thi hành, giá trị 9 bits được bố trí lại như sau:



Cài đặt

Ví dụ: Tạo bảng look-up cho phép loại bỏ pixel nội.



Cài đặt

Hàm minh họa:

```
//Internal Pixel Removing
void genIprLut(unsigned char* table)
{
   int i;
   for(i=0;i<512;i++)             // 2^9=512
   if((i&0x10)==0)            //pixel trung tâm bằng 0
       table[i]=0;
   else if((i&0x20)&&(i&0x8)&&(i&0x2)&&(i&0x80))
       table[i]=1;
   else table[i]=0;
}
```

Nội dung

Giới thiệu ảnh nhị phân

Các phép Morphology cơ bản

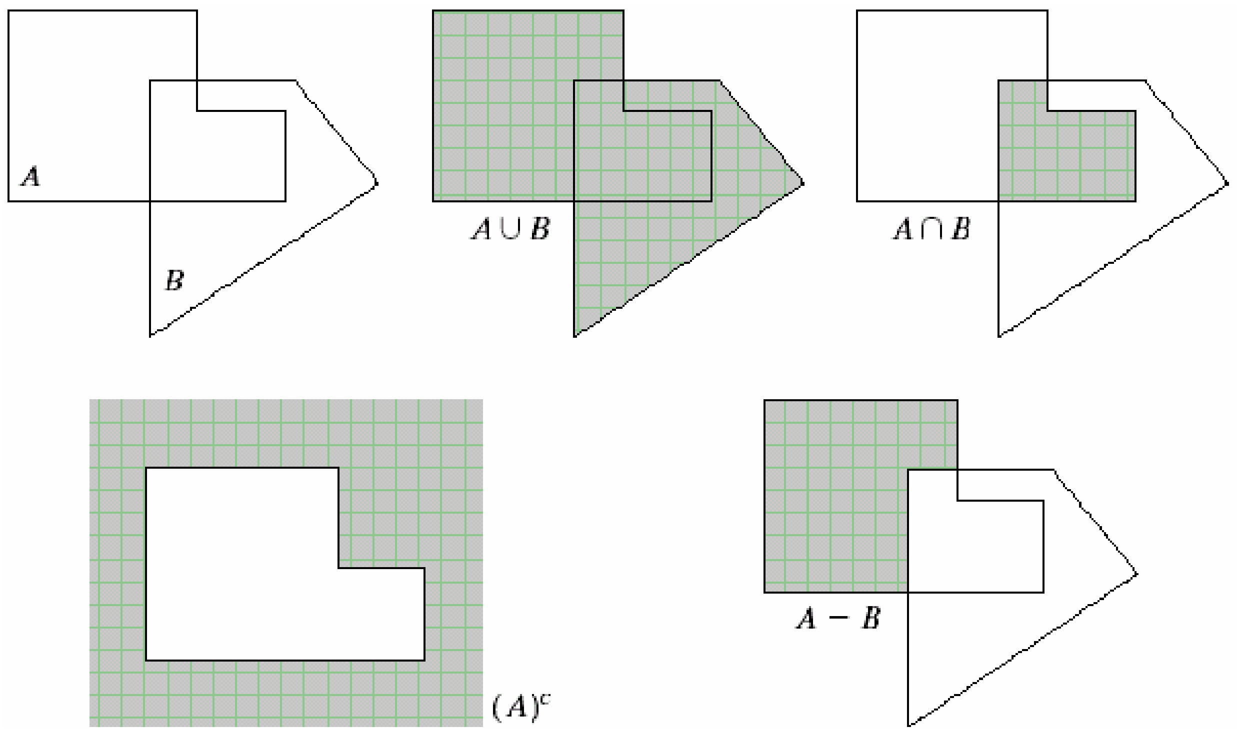
Nhắc lại một số phép toán trên tập hợp

Các phép Morphology tổng quát

- Erosion and dilation
- Opening and closing
- Hit-or-miss, boundary extraction, …

Skeleton via distance transform

Một số phép toán trên tập hợp



$$A^c = \{w \mid w \notin A\}$$

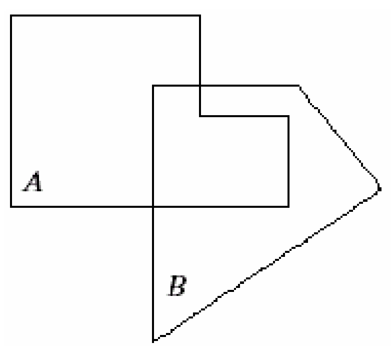
$$A - B = \{w \mid w \in A, w \notin B\}$$

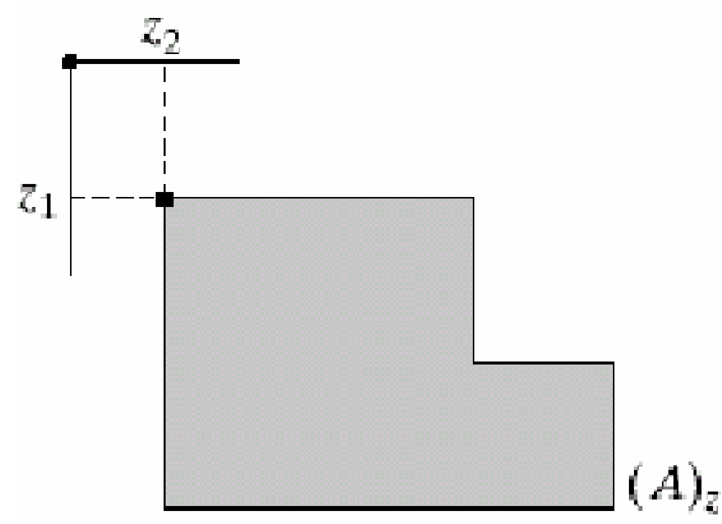
Think of sets A and B as the collections of spatial coordinates

Phép tịnh tiến (Translation Operator)

$$(A)_z = \{w \mid w = a + z, a \in A\}$$

<u>Example</u>



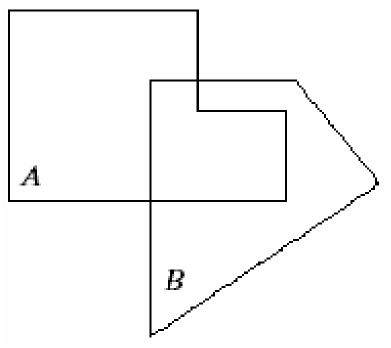


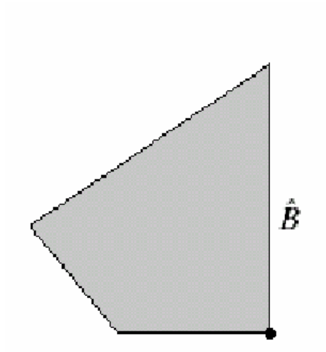
$$z = (z_1, z_2)$$

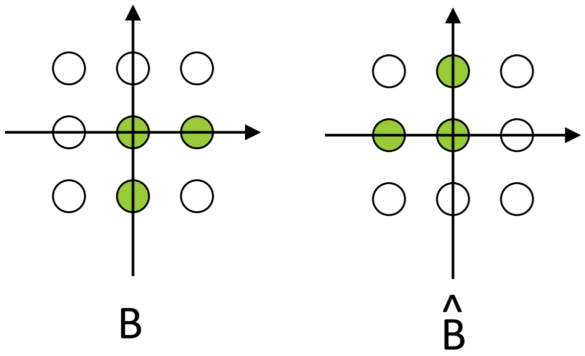
Phép phản chiếu (Reflection Operator)

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

<u>Example</u>







Nội dung

Giới thiệu ảnh nhị phân

Các phép Morphology cơ bản

Nhắc lại một số phép toán trên tập hợp

Các phép Morphology tổng quát

- Erosion and dilation
- Opening and closing
- Hit-or-miss, boundary extraction, …

Skeleton via distance transform

Biểu diễn ảnh nhị phân

Đối tượng được xem là một tập hợp toán học gồm xác pixel đen.

Một pixel là một điểm trong không gian 2 chiều

Ví dụ:

A:

	0	1	2	3	4	5
0						
1						
2						
3				1	1	
4				1	1	
5						

A={(3,3),(4,3),(3,4),(4,4)}

Thành phần cấu trúc B

Definition: là một tập các pixel lân cận với pixel gốc (origin)

<u>Examples</u>

Lưu ý: các thành phần cấu trúc khác nhau sẽ tạo ra kết quả khác nhau

Phép giãn (Dilation)

Cho A là một ảnh nguồn

Cho B là mộ thành phần cấu trúc

Ta định nghĩa:

$$A \oplus B = \{p \mid p = a + b, a \in A, b \in B\}$$
$$= \bigcup_{b \in B} (A)_b$$

*Lưu ý: Khác với phép giãn trong phần 1, phép giãn này được định nghĩa với thành phần cấu trúc B (khi định nghĩa thành phần cấu trúc thì chúng ta phải định nghĩa) gốc toạ độ của nó)

Phép giãn (Dilation)

Ví dụ:

$$B: \quad \begin{array}{c|c|c|} 0 & 1 & 1 \\ \hline & 0 & 1 \end{array} \qquad B=\{(0,0),(1,0)\}$$

$$\begin{aligned} A \oplus B &= (A)_{(0,0)} \cup (A)_{(1,0)} \\ &= A \cup \{(4,3),(5,3),(4,4),(5,4)\} \\ &= \{(3,3),(4,3),(3,4),(4,4),(5,3),(5,4)\} \end{aligned}$$

→

<u>Definition</u> $$Y = X \oplus B = \{z \mid (\hat{B})_z \cap X \neq \phi\}$$

or

<u>Example</u> $$\boxed{Y = X \oplus B = \bigcup_{b \in B} X_b = \bigcup_{x \in X} B_x = B \oplus X}$$

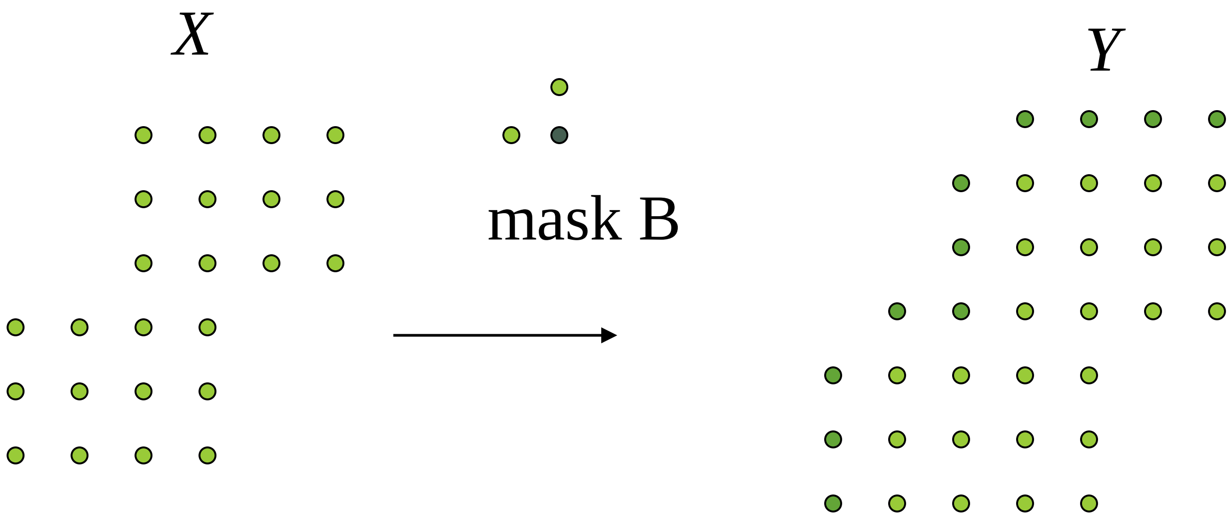
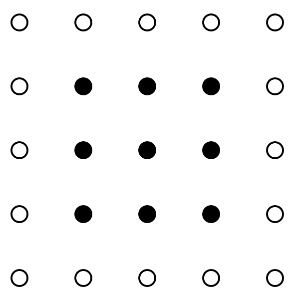
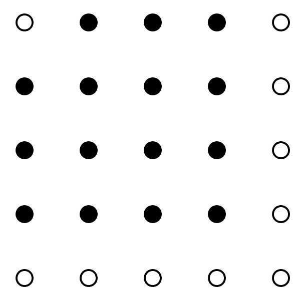


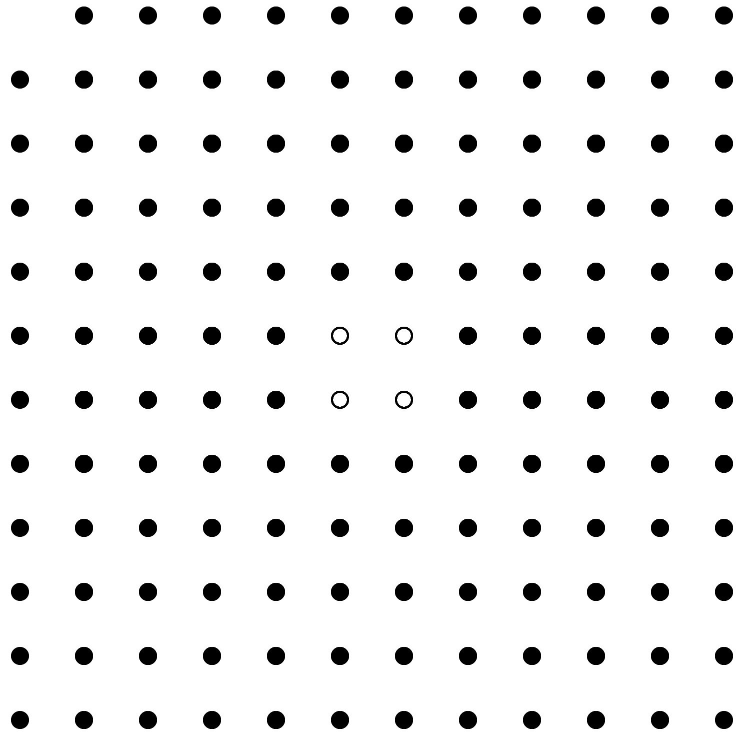
Illustration by Animation



X

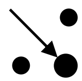


$X \oplus B$



X^c $X^c \oplus B$

origin



B

B={a,b,c}

a=(0,0),b=(0,-1),c=(-1,0)

Phép giãn (Dilation)

* Lưu ý: vị trí gốc của B là rất quan trọng

Ví dụ:

ảnh được giãn

$$B: \quad \begin{array}{c|c|c|} \cline{2-3} 0 & 1 & 1 \\ \cline{2-3} \multicolumn{1}{c}{} & \multicolumn{1}{c}{-1} & \multicolumn{1}{c}{0} \end{array} \qquad B=\{(-1,0),(0,0)\}$$

*Muốn giãn ra mọi phía:

Tuy nhiên không phải lúc nào cũng dễ dàng đc cấu trúc

$$B: \quad \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Phép giãn (Dilation)

Ví dụ:

A:

	0	1	2	3	4	5
0						
1		1				
2			1	1		
3			1	1		
4					1	
5						

A={(1,1),(2,2),(3,2),(2,3),(3,3),(4,4)}

B:

	-1	0	1
0	1	0	1

A:

	0	1	2	3	4	5
0						
1	1		1			
2		1	1	1	1	
3		1	1	1	1	
4				1		1
5						

$A \oplus B = (A)_{(-1,0)} \cup (A)_{(1,0)}$

Nhận xét: ảnh đích nhận được bằng cách trượt thành phần cấu trúc lên ảnh. Khi gốc của thành phần cấu trúc trùng với pixel đen thì các vị trí tương ứng với các pixel đen trong thành phần cấu trúc được đánh dấu để sau này bậc lên 1.

Phép co (**Erosion)**

$A \Theta B = \{p \mid (B)_p \subseteq A\}$

= "tập các pixel p sao cho dời thành phần cấu trúc B bởi p sẽ nằm trong A"

Ví dụ

A:

	0	1	2	3	4	5
0						
1						
2						
3				1	1	
4				1	1	
5						

$A=\{(3,3),(4,3),(3,4),(4,4)\}$

B:

	0	1
0	1	1

$B:\{(0,0),(1,0)\}$

$(B)_{(3,3)} = \{(3,3),(4,3)\} \subseteq A$

$(B)_{(4,3)} = \{(4,3),(5,3)\} \not\subseteq A$

$(B)_{(3,4)} = \{(3,4),(4,4)\} \subseteq A$

$(B)_{(4,4)} = \{(4,4),(5,4)\} \not\subseteq A$

$\Rightarrow A \Theta B=\{(3,3),(3,4)\} \subseteq A$

Phép co (**Erosion)**

* Lưu ý: AΘB không phải lúc nào cũng là tập con của A, nói cách khác là AΘB$\subseteq$A không phải lúc nào cũng đúng.

Ví dụ:

B:

0	0	1
	0	1

$A\Theta B\ )=\{(2,3),(3,3),(2,4),(3,4)\}\not\subset A$

* Nói chung, co và giãn không phải là hai thao tác ngược nhau.

Erosion

Definition

$$\boxed{Y = X \ominus B = \{x : B_x \subset X\}}$$

Example

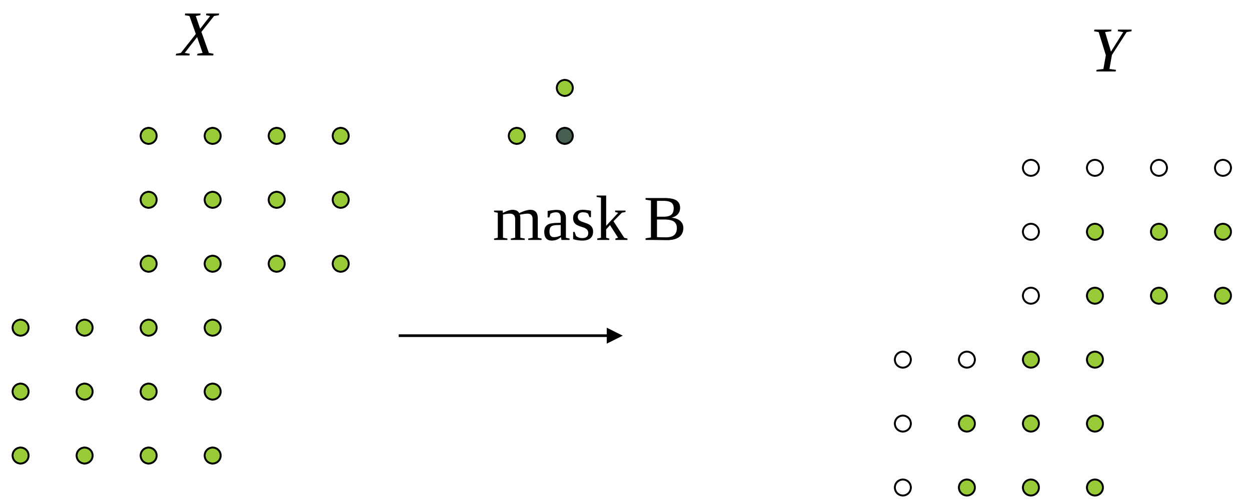
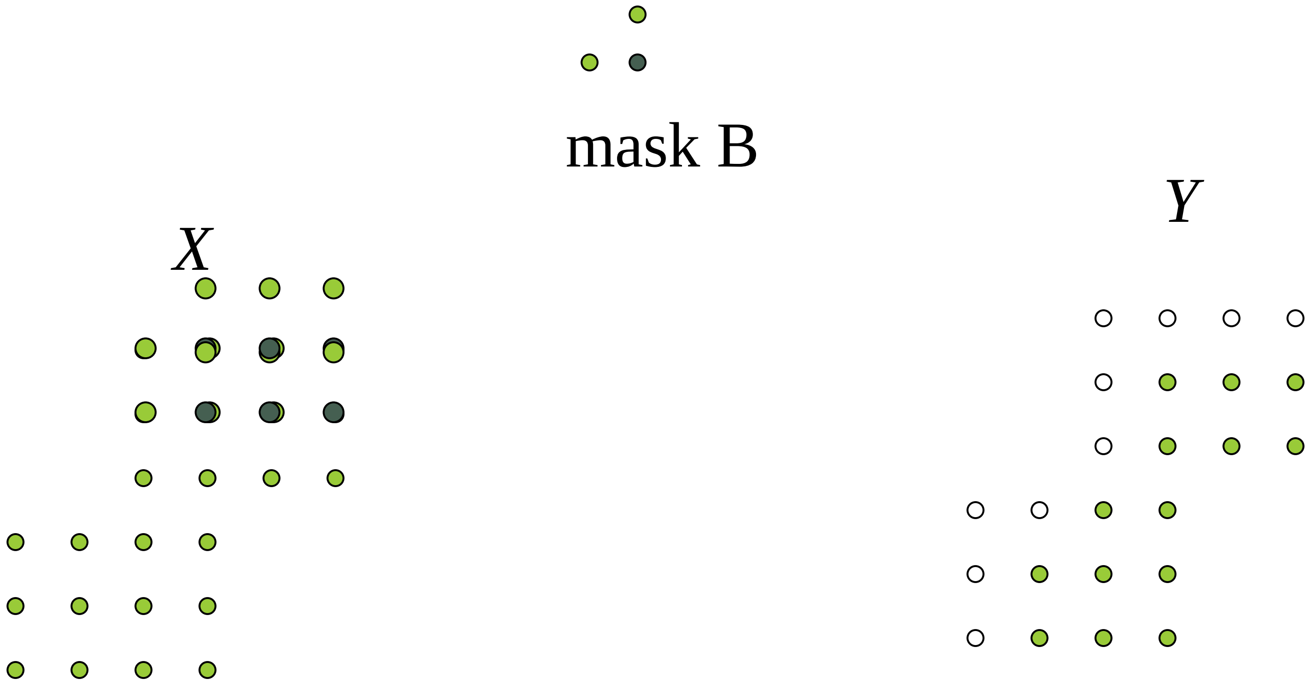


Illustration By Animation



Phép co (**Erosion)**

*Mệnh đề: $(A \Theta G)^C = A^C \oplus \hat{G}$

Chứng minh:

Theo định nghĩa : $A \Theta G = \{p \mid (G)p \subseteq A\}$

Suy ra : $(A \Theta G)^C = \{p \mid (G)_p \subseteq A\}^C = \{p \mid (G)_p \not\subset A\}$

$= \{p \mid (G)_p \cap A \neq \emptyset\}$ hay $\{p \mid (G)_p \cap A^C = \emptyset\}$

Do $(G)_p = \{z \mid z=g+p, g \in G\}$

$= \{p \mid g+p=a, a \in A^C, g \in G\}$

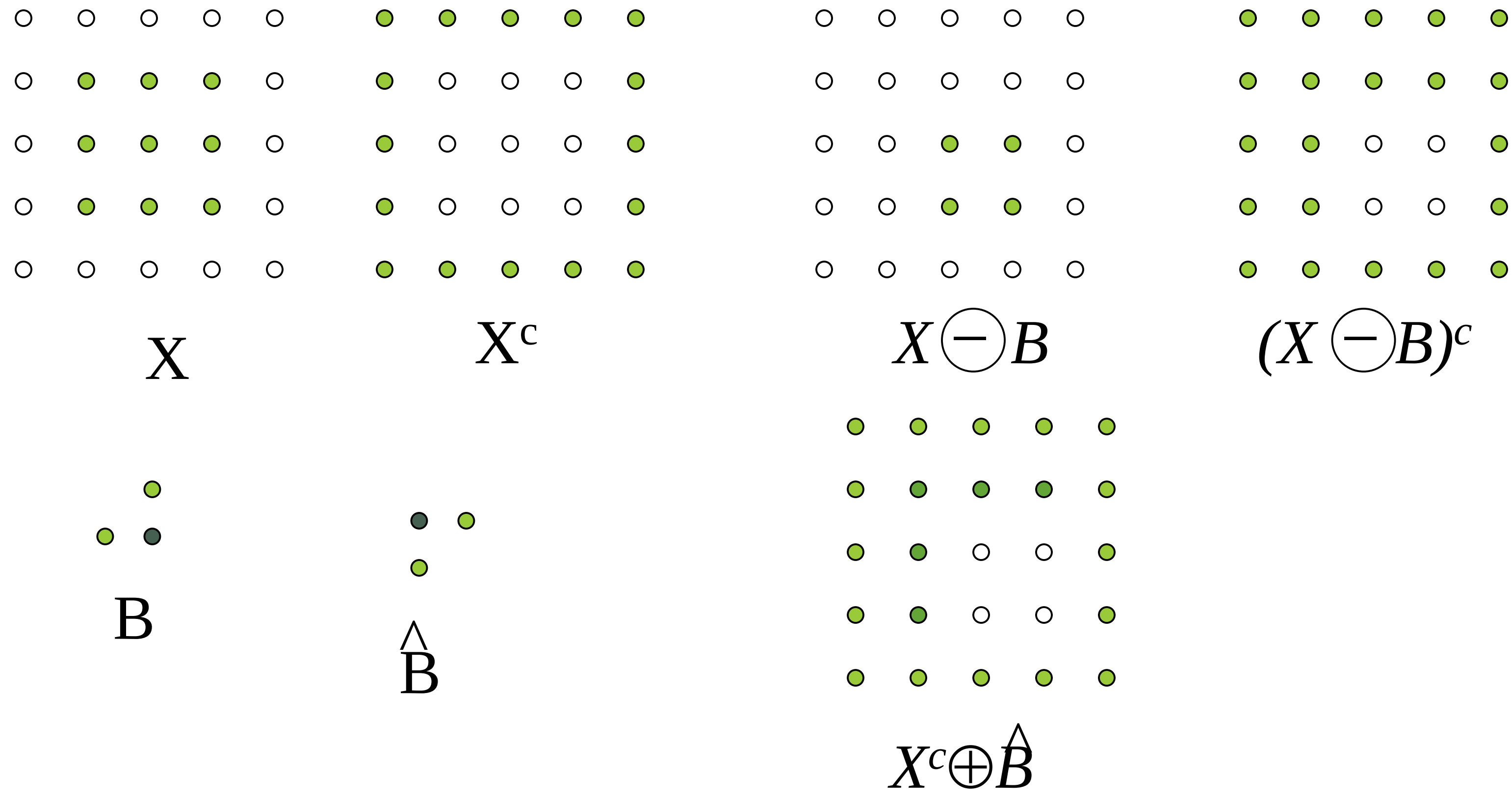
$= \{p \mid p=a-g, a \in A^C, g \in G$

Ta có $\hat{G} = \{-g, g \in G\}$

$= \{p \mid p=a+g, a \in A^C, g \in \hat{G}\}$

$= A^C \oplus \hat{G}$

Example

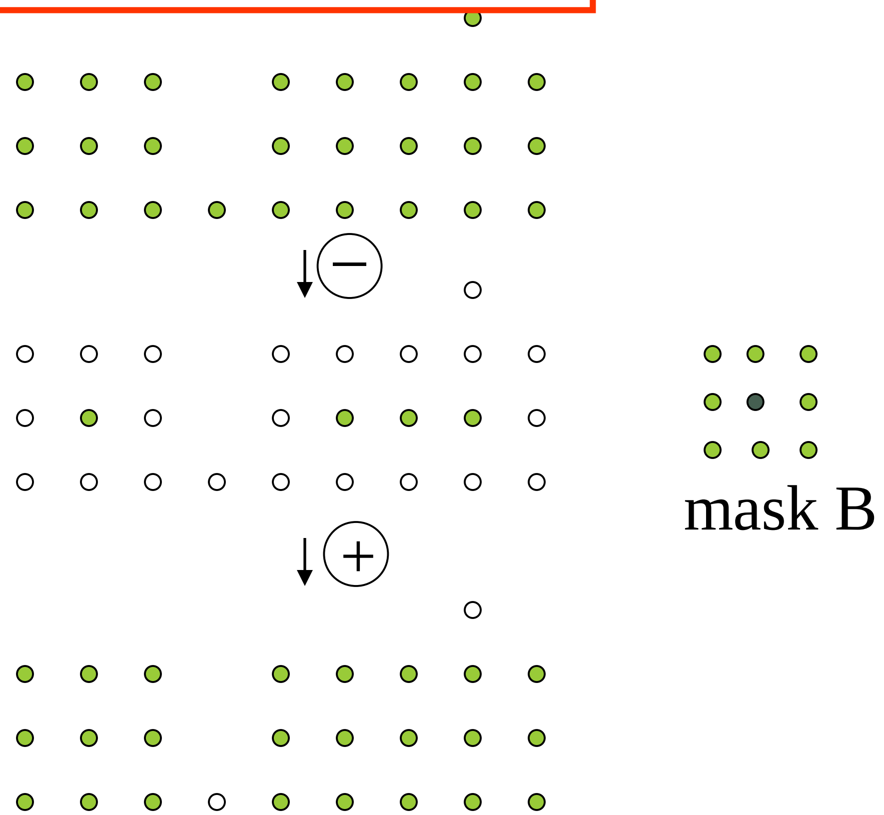


Opening Operator

Definition $X \circ B = (X \ominus B) \oplus B$

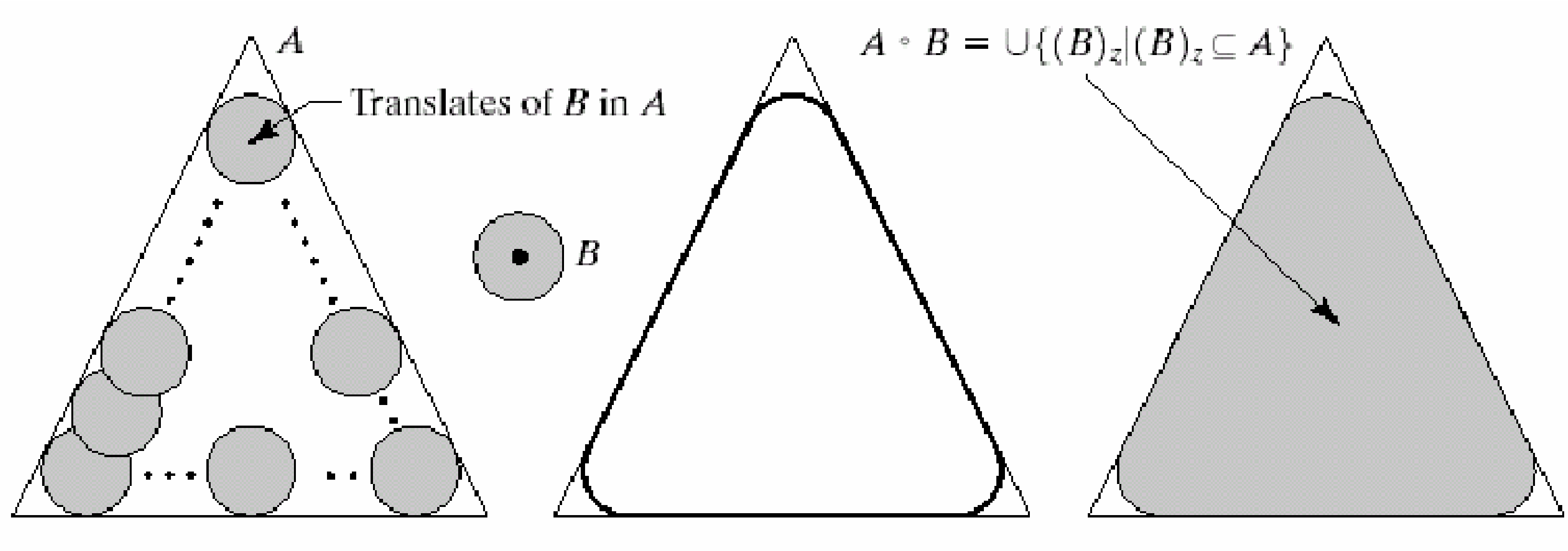
Example

X



$X \circ B$

Geometric Interpretation of Opening Operator

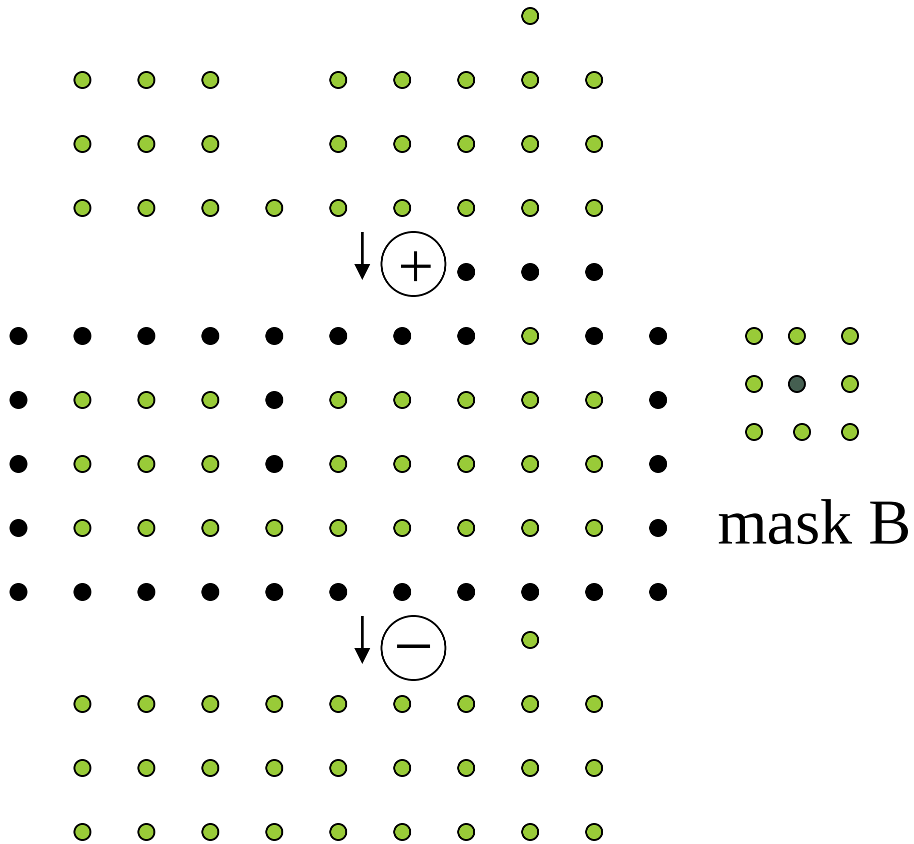


Closing Operator

Definition $\boxed{X \bullet B = (X \oplus B) \ominus B}$

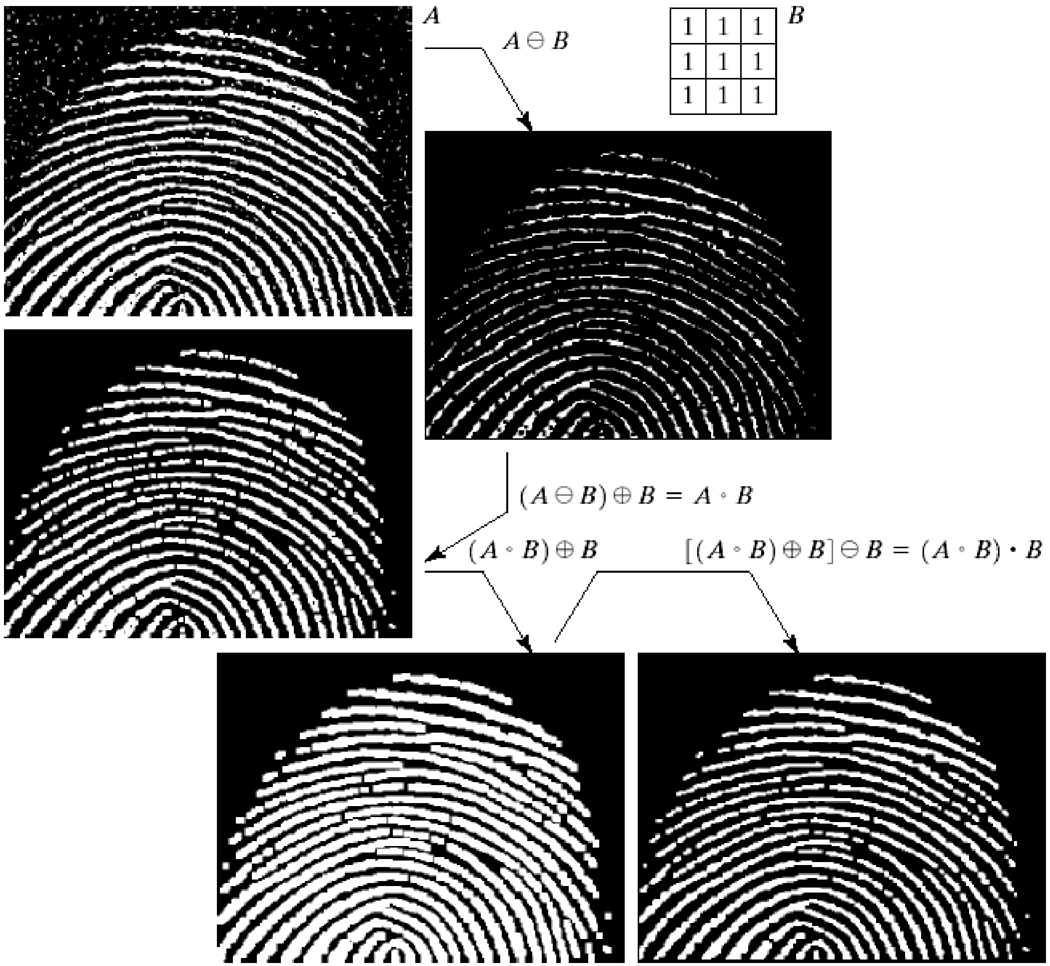
Example

X



$X \bullet B$

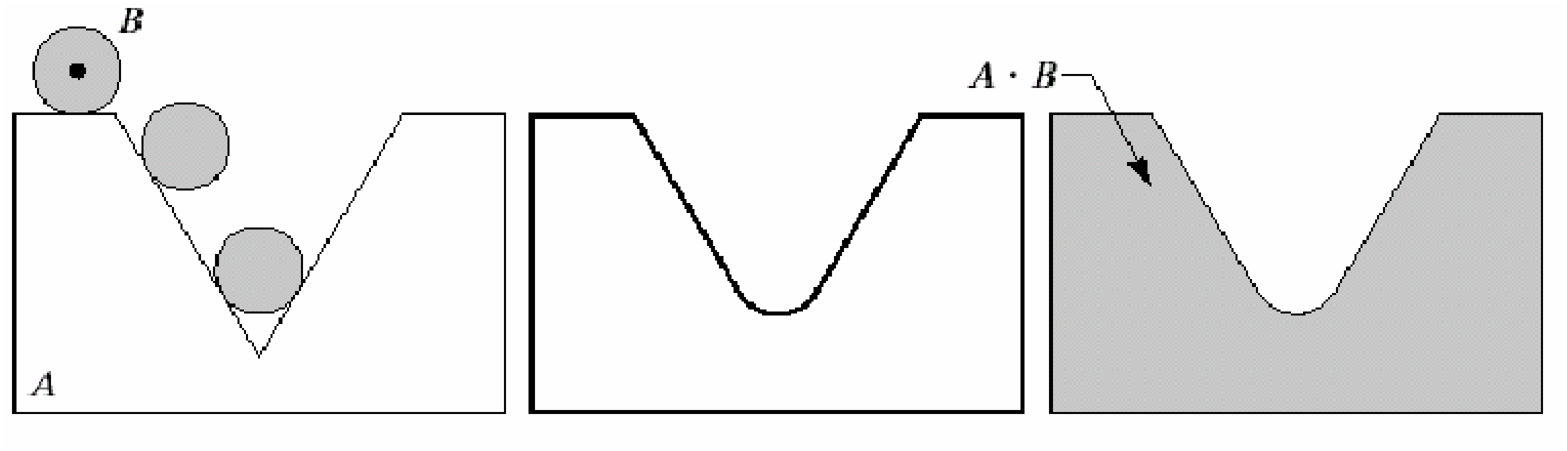
Open   Close



a b
d c
e f

FIGURE 9.11
(a) Noisy image.
(c) Eroded image.
(d) Opening of A.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

Geometric Interpretation of Closing Operator



Properties of Opening and Closing Operators*

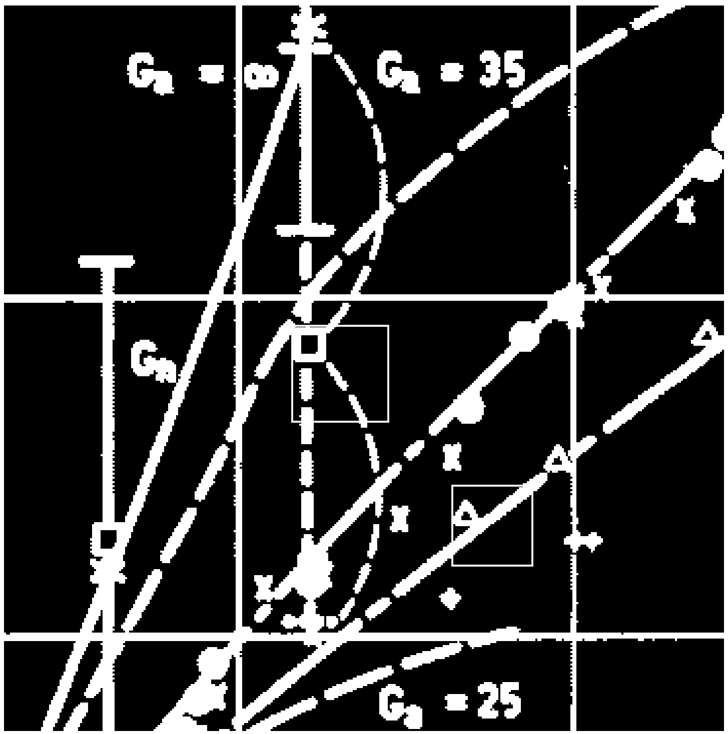
Opening

- $X \circ B \subseteq X$
- $X \subseteq Y \Rightarrow X \circ B \subseteq Y \circ B$
- $(X \circ B) \circ B = X \circ B$

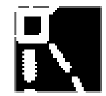
Closing

- $X \subseteq X \bullet B$
- $X \subseteq Y \Rightarrow X \bullet B \subseteq Y \bullet B$
- $(X \bullet B) \bullet B = X \bullet B$

$$(X \bullet B)^C = X^C \circ \hat{B} \qquad (X \circ B)^C = X^C \bullet \hat{B}$$



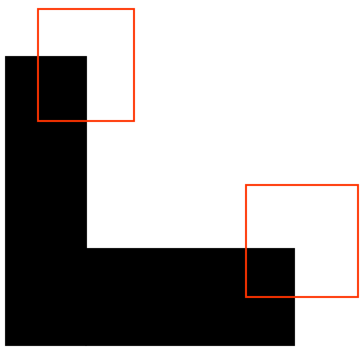
Templates



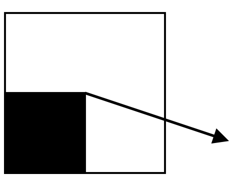
A

B

Illustration by a Simpler Case



X



origin

Template B

How to find the match of A in X
using a computer?

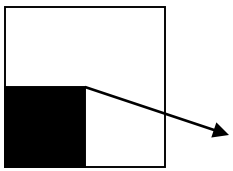
Hit: the southwest quadrant must be black in X

Hit: the other three quadrants must be white in X

Hit: the southwest quadrant must be black in X

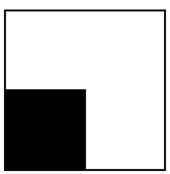
Hit: the other three quadrants must be black in X^c

Matching via Hit-or-Miss

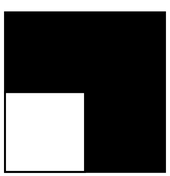


origin

Template B



Template B_1



Template B_2

Hit: the southwest quadrant must be black in X

$$X_1 = X \ominus B_1$$

Hit: the other three quadrants must be black in X^c

$$X_2 = X^c \ominus B_2$$

To satisfy both conditions, we need to take the Intersection of X_1 and X_2

Hit-or-Miss Operator

Definition $X \circledast B = \boxed{(X \ominus B_1) \cap (X^c \ominus B_2)}$

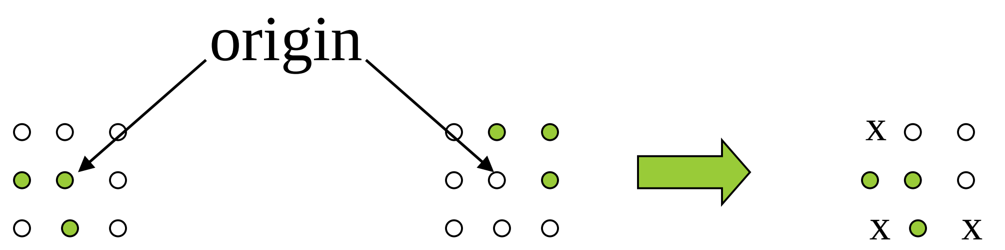
Hit Miss

Why $\ominus$?

Why complement?

Why intersection?

<u>Structuring element example</u>

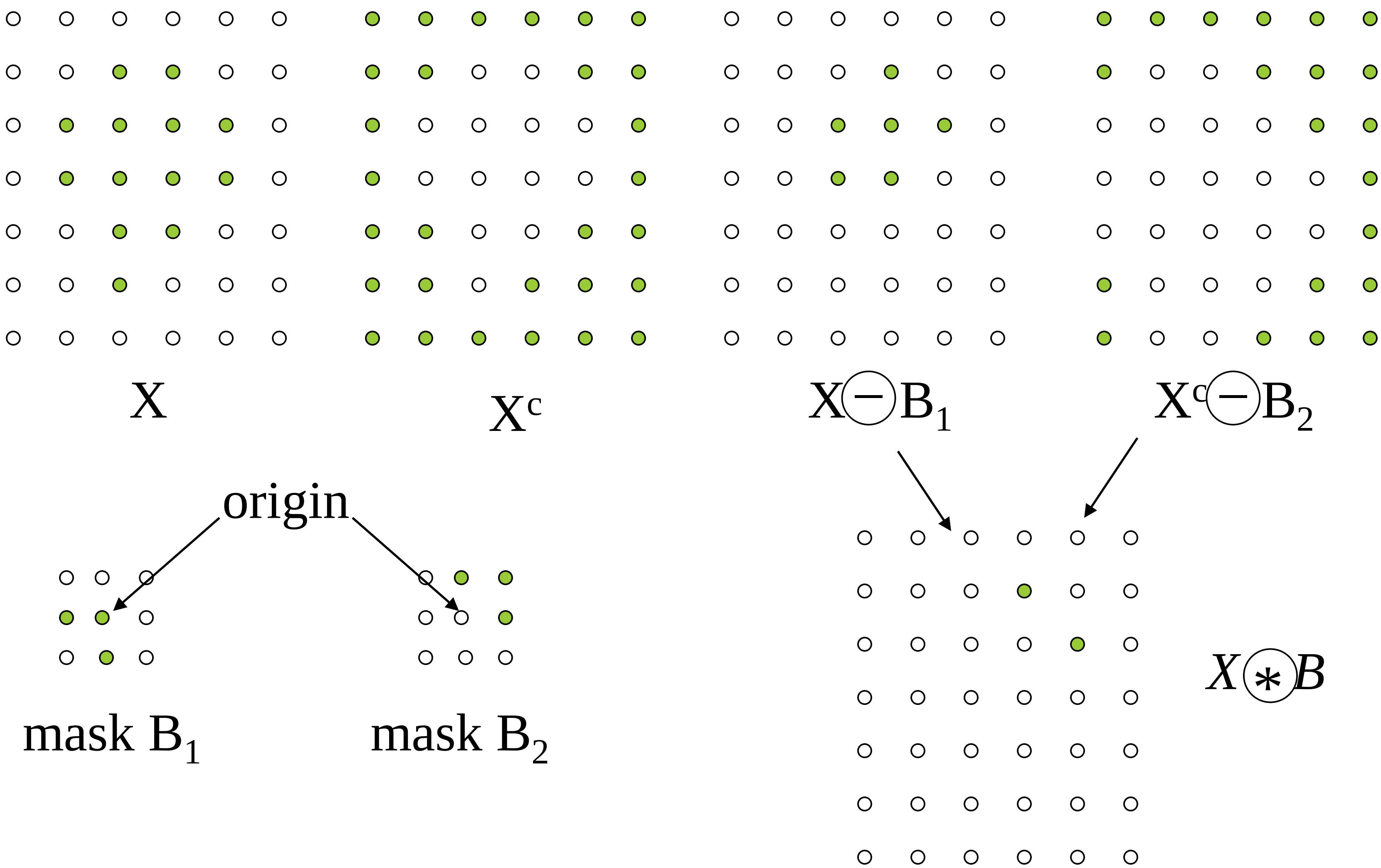


mask B_1 mask B_2 mask B

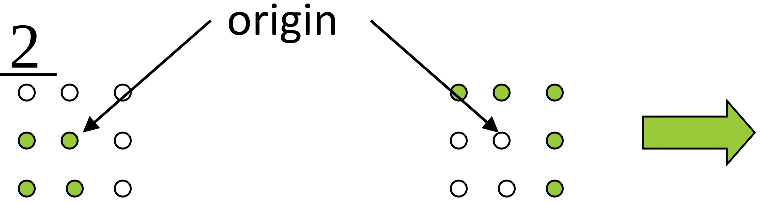
$$\begin{matrix} 0 & -1 & -1 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{matrix}$$

MATLAB

Example 1


X
X^c
$X \ominus B_1$
$X^c \ominus B_2$
origin
mask B_1
mask B_2
$X \circledast B$

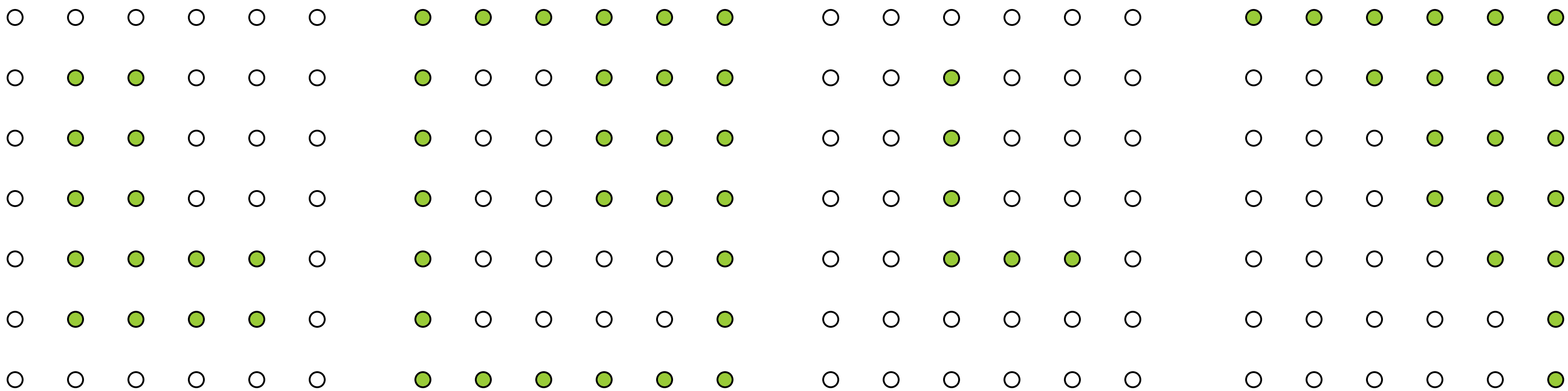
Example 2



mask B_1 mask B_2 mask B

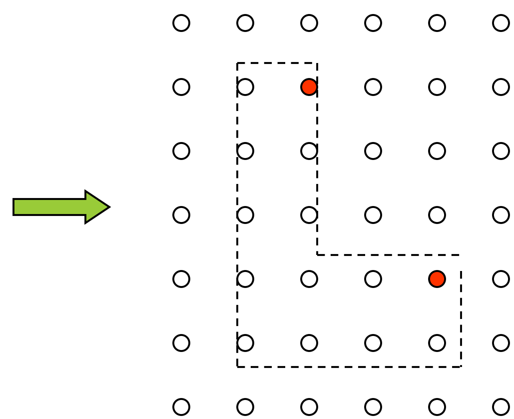
$$\begin{matrix} -1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \end{matrix}$$

MATLAB



X X^c $X \ominus B_1$ $X^c \ominus B_2$

Now, take the intersection
and note that only two points
Remain (highlighted by red)



$X \circledast B$

Roadmap of Morphological Filtering

basic set operators (complement, union, intersection, difference)
dilation/erosion operators

closing/opening, Hit-or-Miss operators

morphological filters/algorithms
(boundary, region filling, thinning)

1. Boundary Extraction

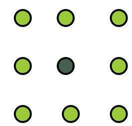
Definition

$$\partial X = X-(X \ominus B)$$

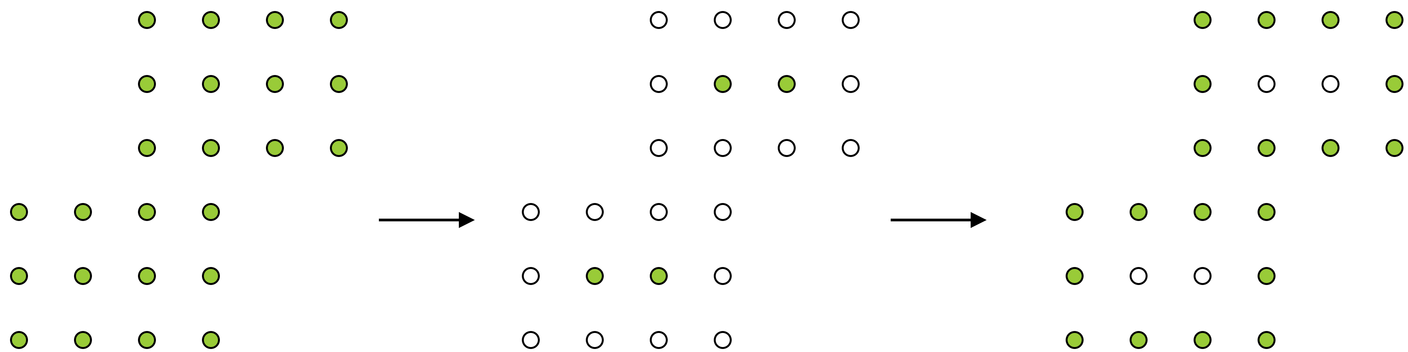
<u>Example</u>

How about

$$\partial X = (X \oplus B) - B?$$



mask B



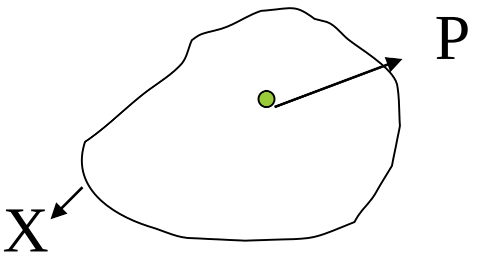
X $\quad$ $X \ominus B$ $\quad$ $X-(X \ominus B)$

Image Example

X

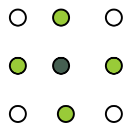
∂X

2. Region Filling



Idea: recursively expand the region around P but stop the expansion at the boundary of X

Iterations:



mask B

expansion   stop at the boundary

$Y_0 = P$

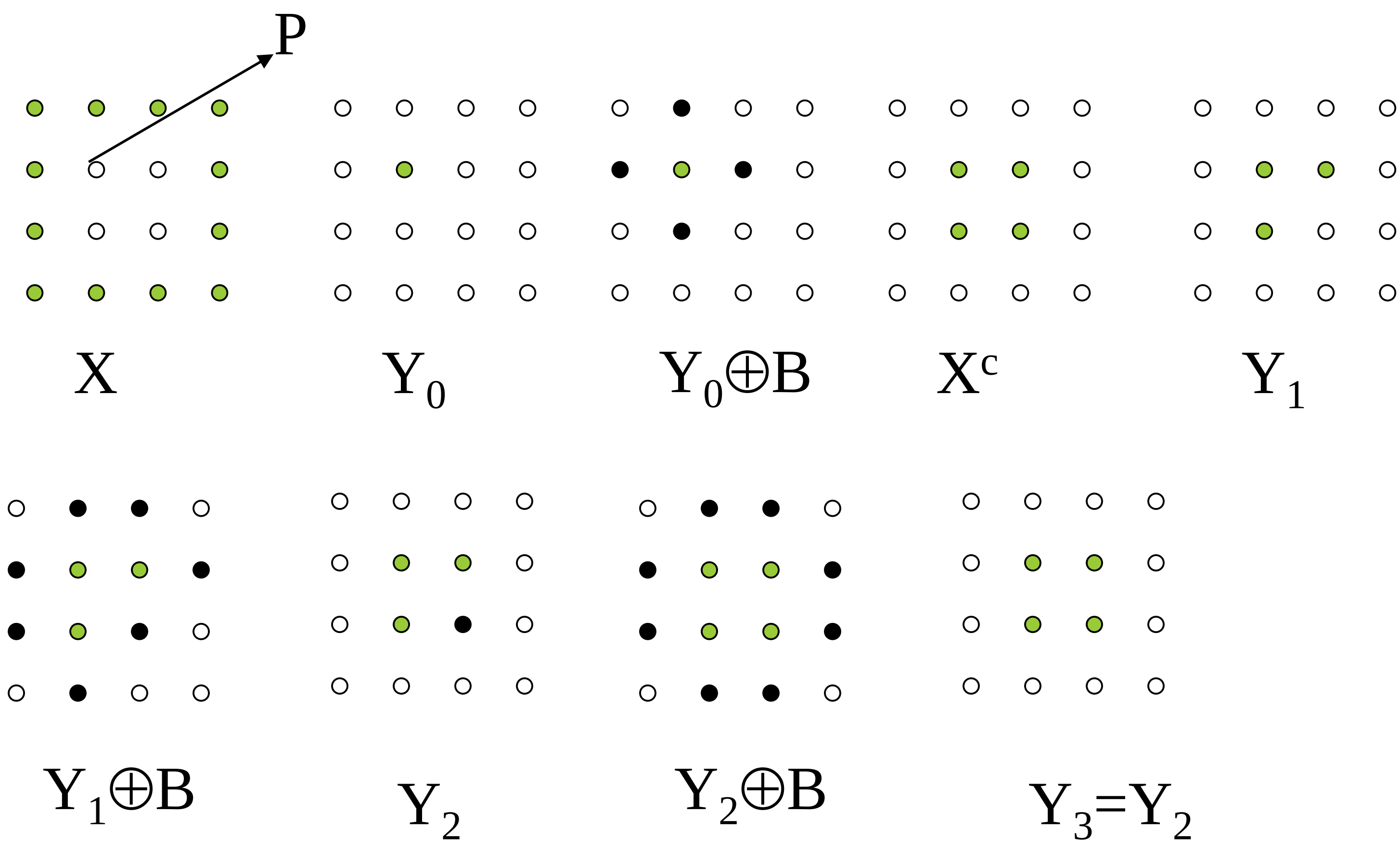
$$\boxed{Y_k = (Y_{k-1} \oplus B) \cap X^c,\ k = 1,2,3\ldots}$$

Why dilation?

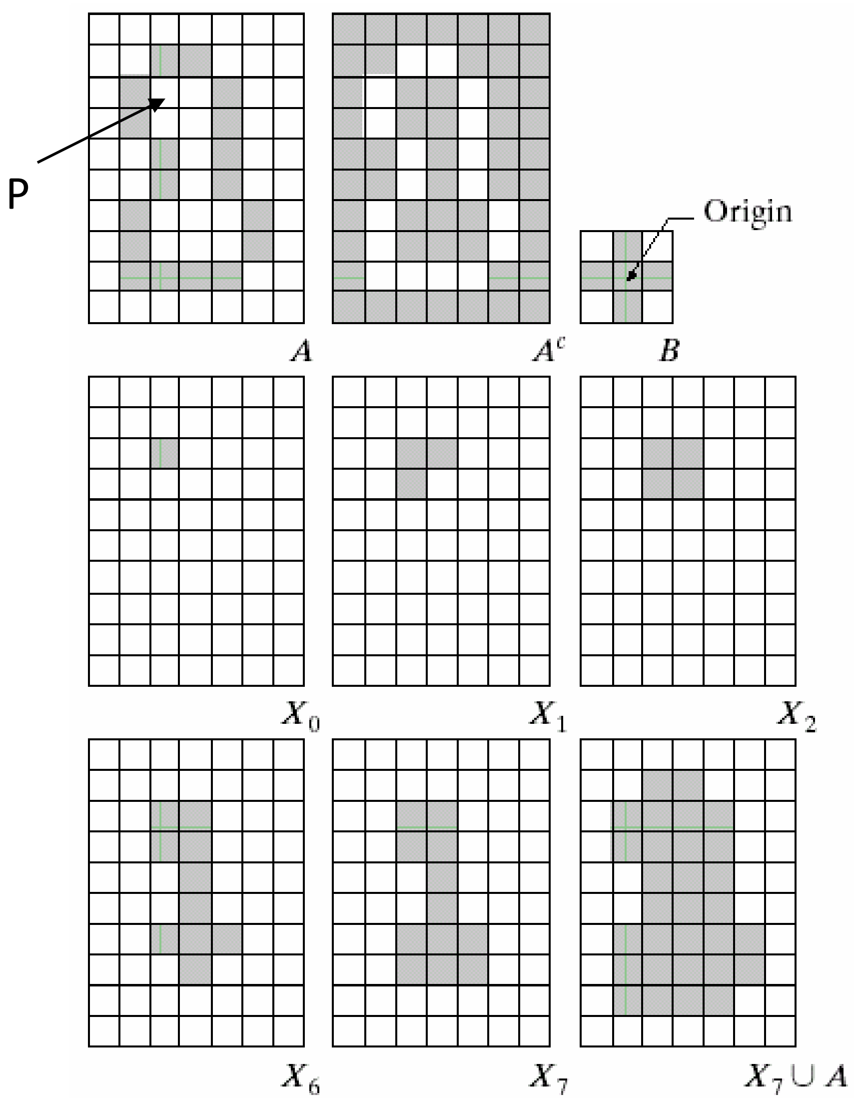
Why intersection with X^c?

Terminate when $Y_k = Y_{k-1}$, output $Y_k \cup X$

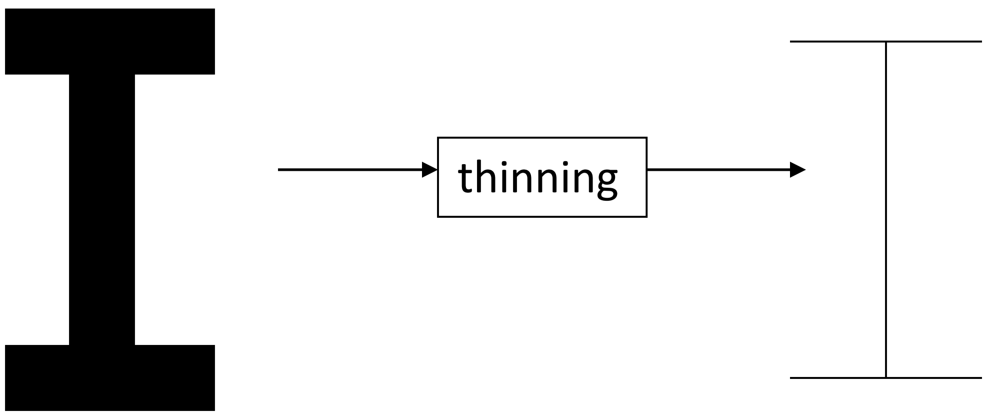
Image Example



Additional Example



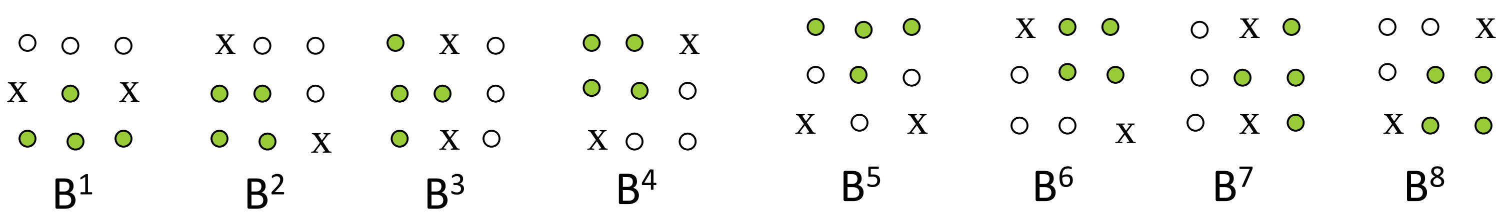
3. Thinning



Intuitively, thinning finds the skeleton of a binary image (you will learn a different way of finding skeleton by distance transform later)

Basic idea:

- Use Hit-or-Miss operator as a sifter
- Use multiple masks to characterize different patterns



$$X_0 = X$$

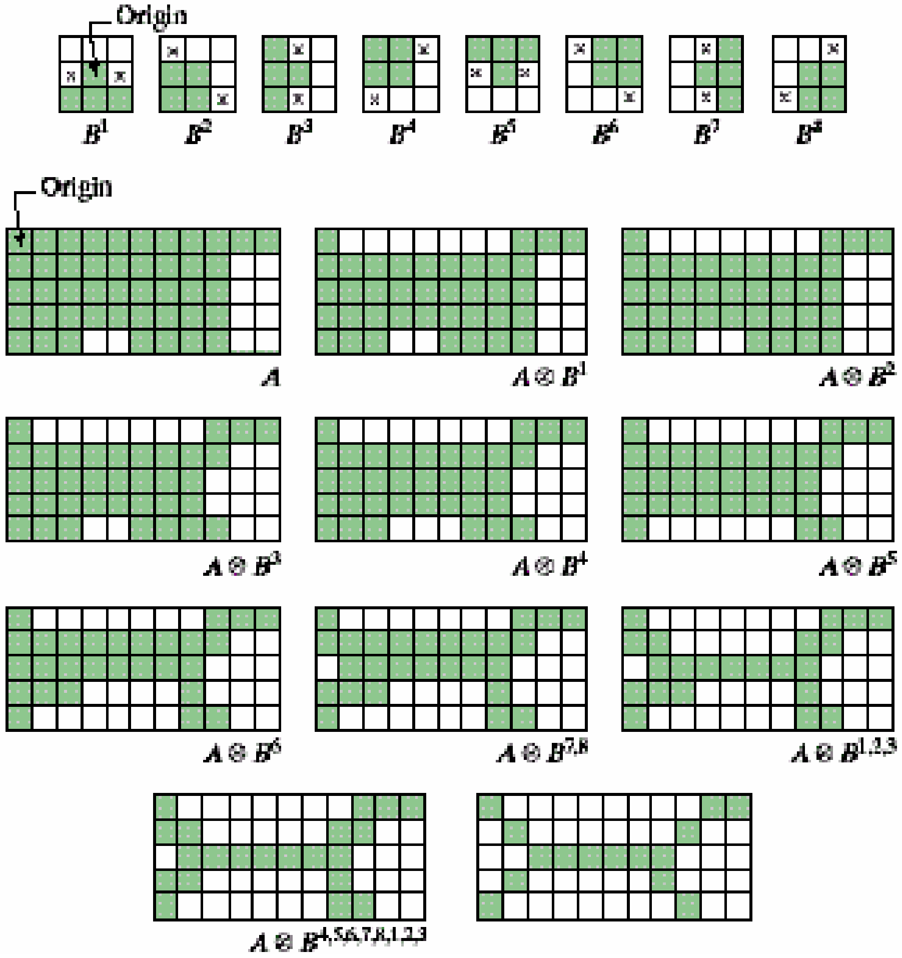
$$\boxed{X_k = (\ldots(\,(X_{k-1} \otimes B^1) \otimes B^2 \ \ldots \otimes B^8)}$$

where $X \otimes B = X - X \circledast B$

Stop the iteration when $X_k = X_{k-1}$

Why eight different B's?
Why hit-or-miss?

Image Example



Nội dung

Giới thiệu ảnh nhị phân

Các phép Morphology cơ bản

Nhắc lại một số phép toán trên tập hợp

Các phép Morphology tổng quát

- Erosion and dilation
- Opening and closing
- Hit-or-miss, boundary extraction, …

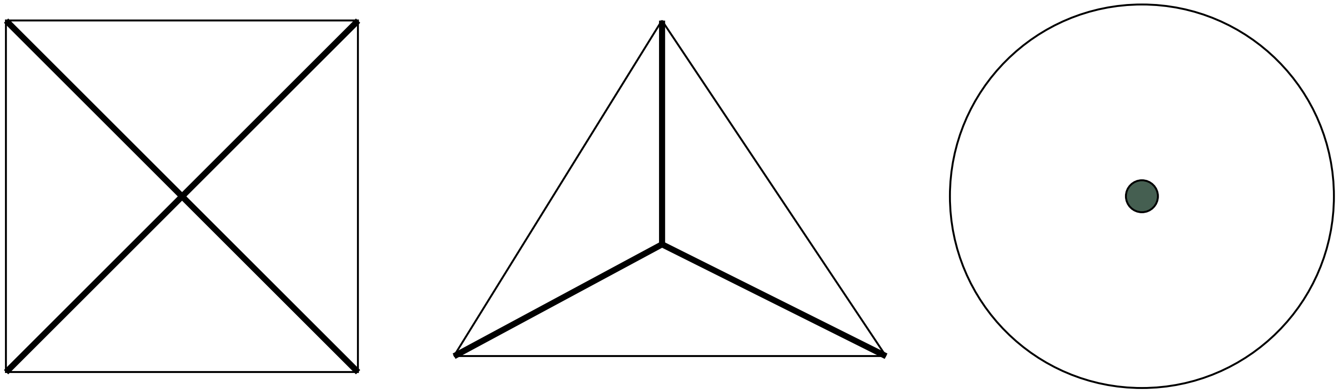
Skeleton via distance transform

Medial Axis (Skeleton)

- Definition

  Suppose that a fire line propagates with constant speed from the contour of a connected object towards its inside, then all those points lying in positions where at least two wave fronts of the fire line meet during the propagation will constitute a form of a **skeleton**

- Examples



Define $$X \ominus kB = (((X \underbrace{\ominus B) \ominus B) \ldots \ominus B)}_{\text{k-fold}}$$

and $$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

Then the skeleton of an image A is given by

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

Image Example

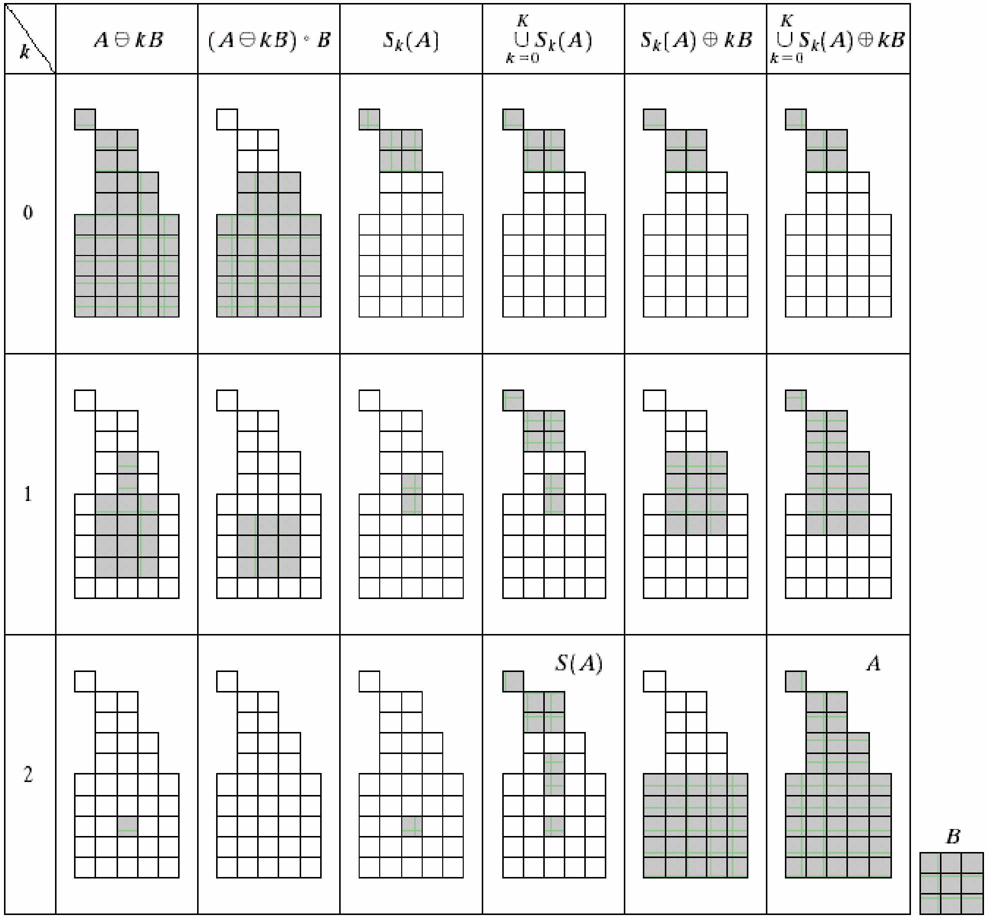
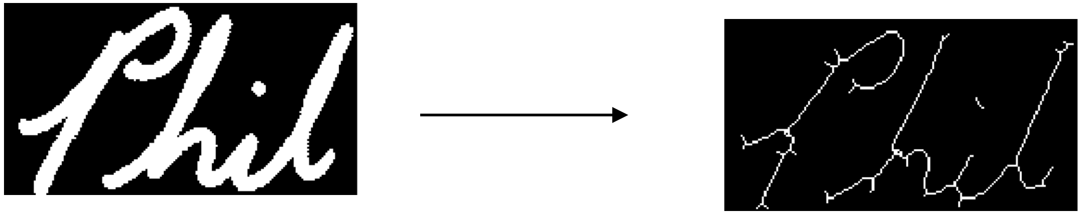


Image Example



original skeleton

Image Example (Con’t)

Binary fingerprint image

Skeleton image

Summary of Morphological Filtering

Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Translation	$(A)_z = \{w \mid w = a + z, \quad \text{for } a \in A\}$	Translates the origin of A to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)

Summary (Con'd)

Hit-or-miss transform	$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$ $= (A \ominus B_1) - (A \oplus \hat{B}_2)$	The set of points (coordinates) at which, simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c.
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Region filling	$X_k = (X_{k-1} \oplus B) \cap A^c$; $X_0 = p$ and $k = 1, 2, 3, \ldots$	Fills a region in A, given a point p in the region. (II)
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\ldots((A \otimes B^1) \otimes B^2) \ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)