Image Processing

Morphological Image Processing (Part I)

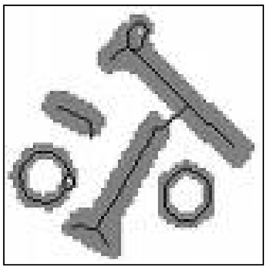
Pattern Recognition and Image Processing Laboratory (Since 2012)

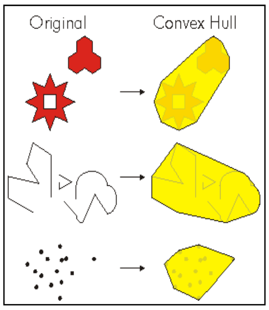
Introduction

หลักๆ ก็คือแกะภาพออกมา

Mathematical morphology is a tool for extracting image components, such as boundaries, skeletons, and convex hulls.

เชื่อมจุดพีคเข้าถึงกัน





Introduction

3 ตัวพื้นฐาน

Morphological techniques include morphological filtering, thinning, and pruning.

ทำให้บางลง ตัดออก (เล็มๆ)

Set Theory

คู่ของมัน (not กำลัง 2)

Let Z be a set of integers, and Z^2 be a pair of elements from the Cartesian product. If $w = (x, y)$ is an element of A, then we write

$$w \in A$$

W เป็นสมาชิกของ A

Similarly, if w is NOT an element of A, we write

$$w \notin A$$

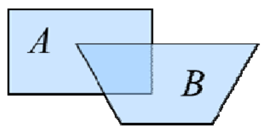
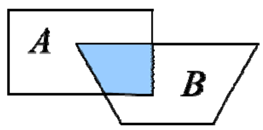
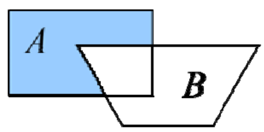
Set Theory

A set B of pixel coordinates that satisfy a particular condition is written as

$$B = \{\, \omega \mid condition \}$$

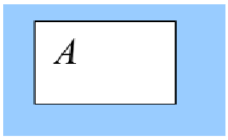
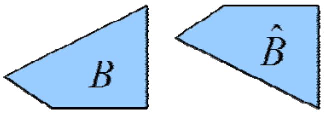
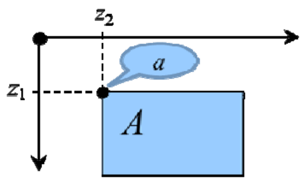
B ประกอบไปด้วยสมาชิก Omega โดยที่ ...Condition...

Set Theory

Logical Operators	Illustrations
$A \cup B$	
$A \cap B$	
$A - B = \{\omega \mid \omega \in A, \omega \notin B\}$	

compliment คือส่วนเติมเต็ม eg เต็ม 10 compliment ของ 3 คือ 7

Set Theory

Logical Operators	Illustrations
A^(compliment) $$A^c = \{\omega \mid \omega \notin A\}$$	 A^c ประกอบไปด้วย Omega โดยที่ Omega ไม่ได้เป็นสมาชิกของ A
ทุกตัวคือ -b โดยที่ b เป็นสมาชิกของ B (พับไปด้านขวา และพับขึ้น) $$\hat{B} = \{\omega \mid \omega = -b,\ for\ b \in B\}$$	
$$(A)_z = \{c \mid c = a + z,\ for\ a \in A\}$$ $$\|$$ $$(z_1 + z_2)$$	

Binary Images, Sets, and Logical Operators

MATLAB Expression for Binary Images

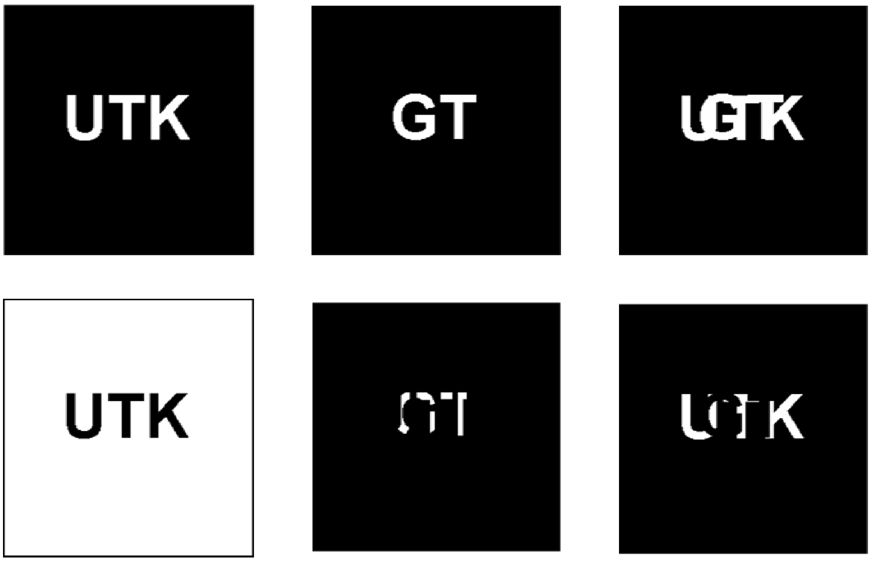
```
>> utk = imread('utk.tif');
>> gt = imread('gt.tif');
>> figure(1); imshow(utk);
>> figure(2); imshow(gt);
```

อาจจะออกสอบ (โค้ด union, intersect)

```
>> comp_utk = ~utk;
>> figure(3); imshow(comp_utk);

>> AorB = utk | gt;       % A union B
>> AandB = utk & gt;      % A intersection B
>> AanddifB = utk & ~gt;
>> figure(4); imshow(AorB);
>> figure(5); imshow(AandB);
>> figure(6); imshow(AanddifB);
```

Binary Images, Sets, and Logical Operators



การขาย (ทำให้binary image มัน growth(ขยาย))

Dilation and Erosion

Function: Dilation is an operator that "grows" or "thickens" objects in a binary image.

Definition: $A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\}$

Property: Commutation; $A \oplus B = B \oplus A$

ทุกตัวคือ -b โดยที่ b เป็นสมาชิกของ B (พับไปด้านขวา และพับขึ้น)

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

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

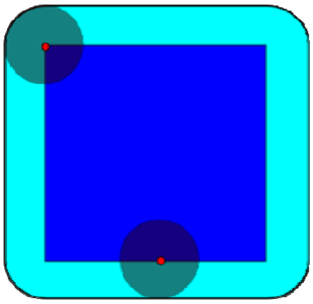
$$\| $$

$$(z_1 + z_2)$$

Dilation and Erosion

```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 1 1 1 0 0 0
0 0 0 0 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 1 0 0 0 0 0
```

- **Dilation**



```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    1
  [1]
1
```

The dilation of the dark-blue square by a disk, resulting in the light-blue square with rounded corners.

1. finding B

Dilation and Erosion

- **IPT function: imdilate**

```
>> A = imread('broken_text.tif');
>> B = [0 1 0;
        1 1 1;     B is a structuring element
        0 1 0];

>> A2 = imdilate(A, B);
>> figure(1); imshow(A);
>> figure(2); imshow(A2);
```

Dilation and Erosion

- **Structuring Element Decomposition**

As a Commutation property

Property: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

$\downarrow$ ← Decomposition

$$(B_1 \oplus B_2)$$

Dilation and Erosion

- **Structuring Element Decomposition**

but if you done this you would be faster (10 time per pix) way 2.5x faster

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & \boxed{1} & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \Rightarrow \begin{bmatrix} 1 & 1 & \boxed{1} & 1 & 1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 1 \\ \boxed{1} \\ 1 \\ 1 \end{bmatrix}$$

you use this thing in picture you would get 25 time per pixel in picture

Dilation and Erosion

- **Structuring Element Decomposition**

- IPT function: strel structuring element

  se = strel(shape, parameter)
  circle, square
  or whatever

object would thinner than it
was

Dilation and Erosion

Function: Erosion is an operator that "shrinks" or "thins" objects in a binary image.

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

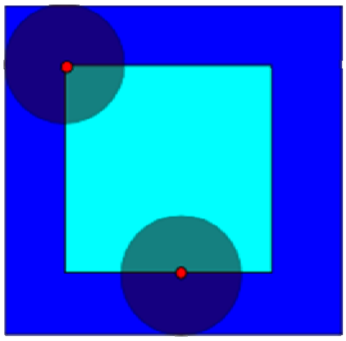
A^(compliment)

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

Dilation and Erosion

- **Erosion**

dark blue is original
sky blue is new (thinner)



```
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 1 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
    1
  [1]
 1
```

ถ้าไม่ตรงก็ไม่เอา (ประมาณเป็น and) ถ้า 1 and 0 จะเป็น 0

The erosion of the dark-blue square by a disk, resulting in the light-blue square.

```
0 0 0        0 0 1
0 1 0 eros 0 1 0
0 0 0        1 0 0

would be 0 0 0
         0 0 0
         0 0 0
```

Dilation and Erosion

- **IPT function: imerode**

```
>> A = imread('wirebond_mask.tif');
>> se1 = strel('disk', 10);
>> A1 = imerode(A, se1);
>> se2 = strel('disk', 5);
>> A2 = imerode(A, se2);
>> se3 = strel('disk', 20);
>> A3 = imerode(A, se3);
>> figure(1);
>> subplot(2, 2, 1); imshow(A);
>> subplot(2, 2, 2); imshow(A1);
>> subplot(2, 2, 3); imshow(A2);
>> subplot(2, 2, 4); imshow(A3);
```

Combining Dilation and Erosion

- **Opening and Closing**

combination of dilation and erosion

Function: Morphological opening is a operator that smoothes object contours, breaks thin connections, and removes thin protrusion.

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

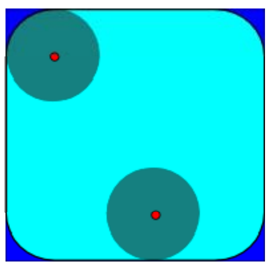
(A Erosion B) And then Dilation B

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

contour = ขอบ
so opening mean make contour smooth and remove a noise(ติ่งๆ ที่โผล่มานิดนึง)

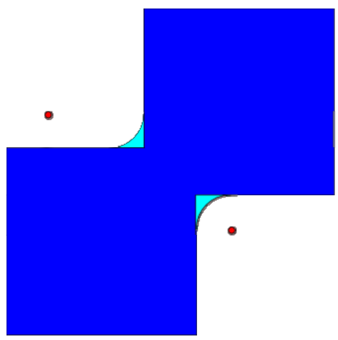
Combining Dilation and Erosion

- **Opening and Closing**



The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners.

หดก่อน แล้วค่อยขยายขึ้น



The closing of the dark-blue shape (union of two squares) by a disk, resulting in the union of the dark-blue shape and the light-blue areas.

ขยายก่อน แล้วหดทีหลัง

Combining Dilation and Erosion

- **Opening and Closing**

Function: Morphological closing is a operator that joints narrow breaks, fills long thin gulfs, and fills holes smaller than the structuring element.

ตรงไหนที่จะขาดไม่ขาดแหล่ closing จะไปเติมเต็มให้ พวกเว้าๆแหว่งจะไปเติมให้ หรือจะไป fill บริเวณไหนก็ตามที่เป็นรู มันจะไปเติมเต็มให้

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

$$A \bullet B = \bigcup \left\{ (B_z) \mid (B_z) \subseteq A^c \right\}$$

opposite of opening

Combining Dilation and Erosion

- **Opening and Closing**

ITP function: opening and closing

```
>> f = imread('noisy_fingerprint.tif');
>> se = strel('square', 3);

>> fo = imopen(f, se);
>> foc = imclose(fo, se);

>> figure(1);
>> subplot(1, 3, 1); imshow(f);
>> subplot(1, 3, 2); imshow(fo);
>> subplot(1, 3, 3); imshow(foc);
```

Combining Dilation and Erosion

or Hit-and-Miss I don't know

- **The Hit-or-Miss Transformation**

Function: It is useful to identify specified configurations of pixels, such as isolated foreground pixels, or pixels that are end points of line segments.

isolate pixel from background

Definition: $A \otimes B = (A \ominus B_1) \cap (A^c \ominus B_2)$

different **Structuring elements**

A^c mean A-inverse in case of binary

Combining Dilation and Erosion

- **The Hit-or-Miss Transformation**

เราอยาก detect อะไรเราก็กำหนด structural element เป็นอันนั้น

ต้องการ + เท่านั้น

$$
\begin{array}{cccccccccccccccc}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&1&1&1&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&1&1&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{array}
$$

A

$$
\begin{array}{ccc}
 & 1 & \\
1 & \boxed{1} & 1\\
 & 1 &
\end{array}
$$

B1 erode with A

$$
\begin{array}{ccc}
1 & & 1\\
 & \boxed{} & \\
1 & & 1
\end{array}
$$

B2 erode with A^c

ทำให้ว่างไว้จะได้ให้เป็ฯจุดอ้างอิง

Combining Dilation and Erosion

- **The Hit-or-Miss Transformation**

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

$$\left(A \ominus B_1\right)$$

```
1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1
0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1
1 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 1 0 1 0 1 1 1 1 1 0 1 0 1
1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1
1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1
```

$$\left(A^C \ominus B_2\right)$$

Combining Dilation and Erosion

- **The Hit-or-Miss Transformation**

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

$$A \otimes B = \left(A \ominus B_1\right) \cap \left(A^c \ominus B_2\right)$$

Combining Dilation and Erosion

- **The Hit-or-Miss Transformation**

ITP function: bwhitmiss

```
>> f = imread('small_squares.tif');
>> figure(1); imshow(f);

>> B1 = strel([0 0 0; 0 1 1; 0 1 0]);
>> B2 = strel([1 1 1; 1 0 0; 1 0 0]);
>> g = bwhitmiss(f, B1, B2);
>> figure(2); imshow(g, []);
```

Combining Dilation and Erosion

- **The Hit-or-Miss Transformation**

ITP function: bwmorph

```
>> f = imread('noisy_fingerprint.tif');
>> se = strel('square', 3);
>> fo = imopen(f, se);
>> foc = imclose(fo, se);
>> g1 = bwmorph(foc, 'thin', 1);
>> g2 = bwmorph(foc, 'thin', 2);
>> ginf = bwmorph(foc, 'thin', Inf);
>> figure(1);
>> subplot(2, 2, 1); imshow(f);
>> subplot(2, 2, 2); imshow(foc);
>> subplot(2, 2, 3); imshow(g1);
>> subplot(2, 2, 4); imshow(g2);
>> figure(2); imshow(ginf);
```

The end of part I