

機器視覺

講座集 - 03
二值圖像處理
惠永林

二值圖像處理

n圖像在量化以獲得數字圖像之前包含連續的強度值n常用的量化級別 :256

(8 位) n還使用 :32、64、128、512 和

4096 (12 位 ,通常用於醫學

圖片)

n更多量化級別 :更好的表示 ,更多存儲n二值圖像 :2 級灰度級量
化 (1 位)

n為什麼是二值圖像 ?

n早期內存和計算能力有限n算法很好理解n需要更少的內存
和快速的執行時間n對象和背景通過
掩碼分離n易於分析

二值圖像

二值圖像對於識別具有獨特輪廓的
物體特別有用

n例如製造廠傳送帶上的組件 n 識別文本和符號

n例如文件處理或解釋路標

n確定物體的方向

n缺點： n需要適當

的照明控制以獲得良好的對比度 n 對於某些應用，僅使用兩
個強度級別無法恢復信息

二值圖像處理

n二值圖像的表示n具有 $m \times n$ 個像素大
小的圖像n 1 (白色)用於對象像素，
0 (黑色)用於背景
像素

n關於二值圖像處理的主題n二值圖像
的形成n幾何性質n拓撲性
質n二值圖像中的對
象識別

圖像分割

n圖像分割n將圖像劃分為

多個區域n識別表示對象的子圖像

n視覺系統中最重要的問題之一n它可以定義為

一種將圖像 $F[i, j]$ 劃分為子圖像的方法，稱為regions,

P₁, …, P_k，使得每個子圖像是一個候選對象n Region n圖像的一個子集

二值圖像分割

n分割是將像素分組到區域中，例如
那

$\bigcup_{i=1}^n P_i = \text{整個圖像}$ ($\{P_i\}$ 是窮舉劃分) $\bigcap_{i,j} P_i \cap P_j = \emptyset$, $i \neq j$ ($\{P_i\}$ 是排他劃分)即，所有點的

分區有一些共同的屬性 n 屬於相鄰區域的
像素，當聯合使用時，不滿足謂詞 n 謂詞可以是“具有均勻強
度”等 n 使用灰度圖像的適當

分割獲得二值圖像

閾值和分割

n如果物體的強度值在某個區間內，而背景的強度值在該區間外

n Thresholding可以用來設置一個區間為1，另一個區間為0來分割物體和背景區域

n對於二元視覺，分割和閾值化是同義詞

二值圖像的閾值

n閾值化是一種轉換灰度的方法

將圖像轉換為二值圖像，以便將感興趣的對象與背景
分開n對象和背景有足夠的對比度是

進行閾值處理所必需的（為什麼？）

二值圖像的閾值

n如果我們對原始圖像 $F[i, j]$ 使用一個閾值 T 來獲得
二值圖像 $B[i, j] = FT[i, j]$ ，然後

$$F_{ij}[,] = \begin{cases} 1 & \text{如果 } T, \neq F[0] \\ 0 & \text{否則} \end{cases}$$

n如果物體強度值在 $[T_1, T_2]$ 范圍內，那麼我們可以使用以下等式進行閾
值化

$$F_{ij}[,] = \begin{cases} 1 & \text{如果 } T_1 \leq F[i,j] \leq T_2 \\ 0 & \text{否則} \end{cases}$$

二值圖像的閾值 n 一個通用的閾值方案，其中一個對象的強度水平

可能來自幾個不相交的區間，表示為 $i \in [,] F_{ij}$

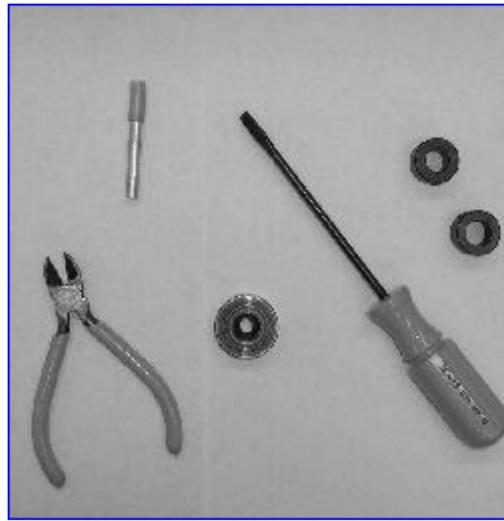
$$F_{ij}[,] = Z_{ij} \text{ if } i \in [,] \\ F_{ij}[,] = 0 \text{ 否則}$$

其中 Z 是對象成分的一組強度值 n 如果有兩個或更多閾

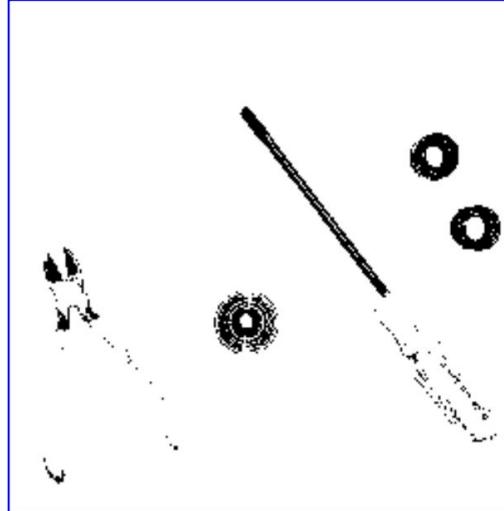
值，則通常根據應用領域的經驗選擇閾值 n 圖像的自動閾值化通常是分析圖像的第一步機器視覺系統中的圖像

例子

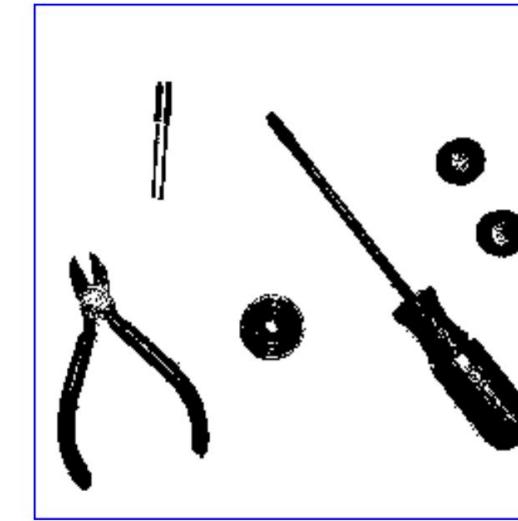
原圖



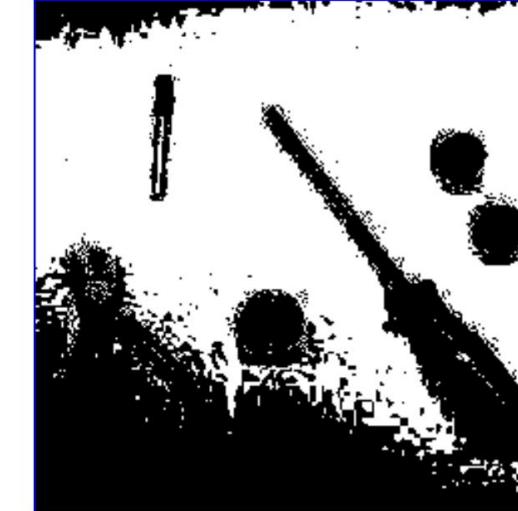
臨界點
太低



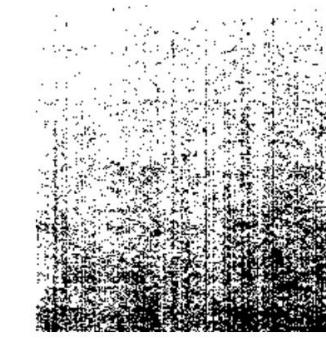
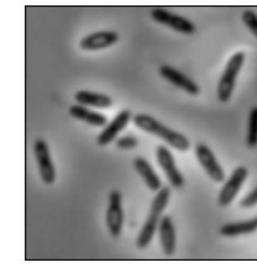
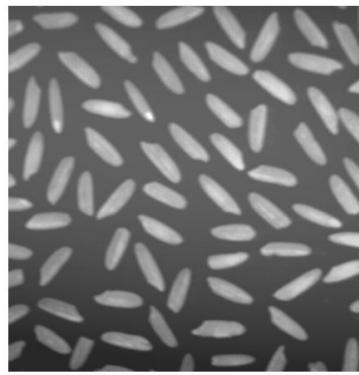
臨界點
分割



臨界點
太高



示例 應用程序



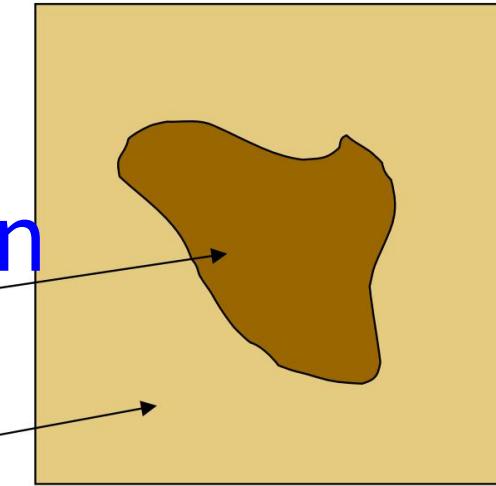
Geometric Properties n

二值圖像中物體的幾何屬性包括 n Size n
Position n

Perimeter n Orientation

目的

背景



n 它們將通過對象的力矩來定義

物體的時刻

n可以從對象的矩中獲得關於二值對象的許多有用信息
n定義對象的二值圖像B為

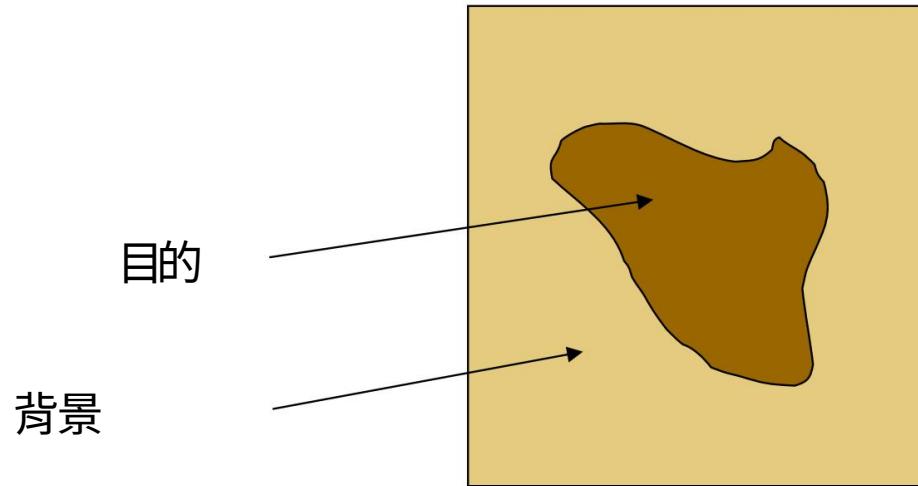
$$B(i, j) = \begin{cases} 1 & \text{代表物體上的點} \\ 0 & \text{否則} \end{cases}$$

Size of an Object n

物體 (面積)的大小由第0 個給定

片刻

$$A = \sum_{j=1}^n B_{ij}(,)$$



物體的位置

n 物體的位置可以表示為 n 封閉矩形 邊界框 n 區域

中心 對噪聲相對不敏感 n 對於二
值圖像，區域中心與質心（強度值）相同 n 中
心質量由第一時刻給出

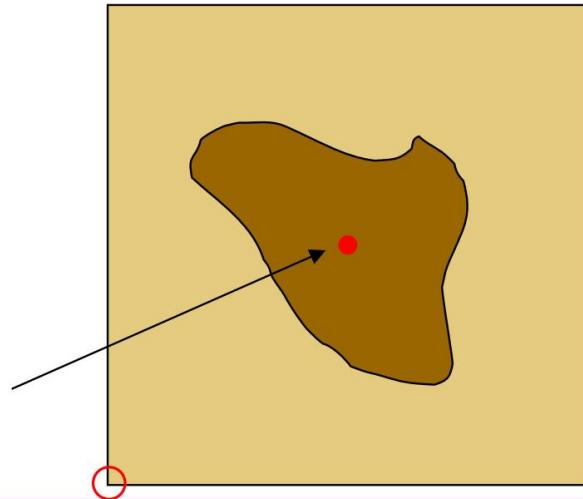
$$\bar{x} = \frac{\sum_{j=1}^n j}{\sum_{j=1}^n 1}, \quad \text{是} = \frac{\sum_{j=1}^n j}{\sum_{j=1}^n 1}$$

啊啊 $iB(j,)$

啊啊 (B, jj)

$$\oint x f(x) dx = x \bar{f}(x) dx$$

質心



對象的方向

n 對象的方向不一定是唯一的（例如圓圈）

n 物體的方向定義為最小慣性

n 這是最小二階矩，其方向是

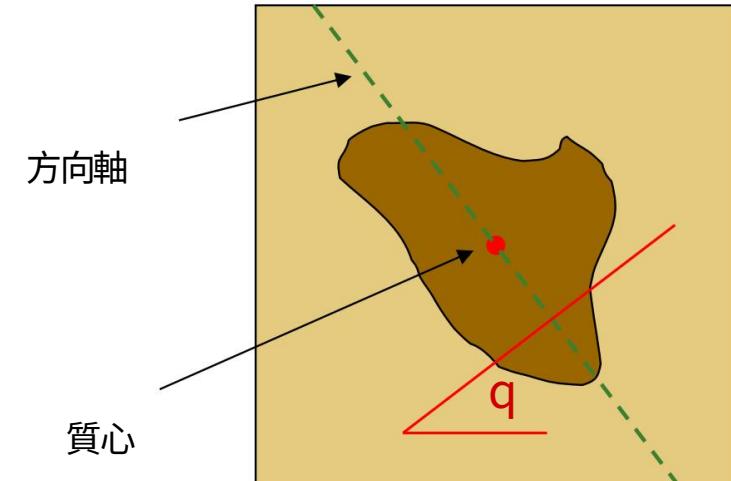
$$= \frac{1}{2} \overline{\text{曬黑 !}} \quad \frac{\#\$}{\#\#} \quad \frac{-}{\$\$}$$

第二時刻在哪裡

$$\bar{m}_{xx} = \frac{1}{2} \overline{(x^2 + y^2)} - \bar{x}\bar{y}$$

$$\bar{m}_{xy} = \frac{1}{2} \overline{(x^2 + y^2)} - \bar{x}\bar{y}$$

$$\bar{m}_{yy} = \frac{1}{2} \overline{(x^2 + y^2)} - \bar{x}\bar{y}$$



方向推導

n二階矩軸

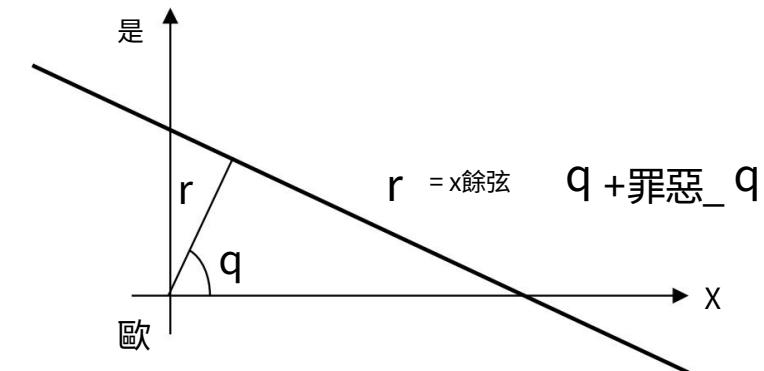
n物點與直線距離平方和最小的直線

n計算一條線到目標點的最小二乘法擬合n令rij為所有目標點與該線的距離，然後

$$C = \sum_{j=1}^n B_{ij}[,] r$$

n使用極坐標系，令 $r = x \cos q + y \sin q$ ，然後對於每個對像到直線

$$2 \times \text{ij} (\text{餘弦 } q + \sin y \text{ ij } q - r)$$



方向推導

因此， $n C = SS (\text{因} q_{ij} \sin + q_{-r})^2 B[j,]$

對 r 取導數，設置為零，求解 r :

$$\frac{\partial}{\partial r} = - \frac{n}{SS} (x_{ij} \cos q_{ij} + qr) \bar{Z}_{ij}$$

$r = x_{ij} \cos q_{ij} + qr$

讓 $x = x - x$ with $y = y -$ ，然後 $C = a \cos q + b \sin q + c \cos q + d \sin q$

$$A = SS (b_{ij} [b], x) = SS [ij], xy b_{ij} c = SS (B_{ij} [,])$$

因此， $b_{ij} (+ b \sin 2)$ 對 $c = a + q$ 取導數，設為零，求解 q

棕色 $2q = \frac{\text{——}}{\text{交流電}}$

投影n投影是二進制

的緊湊表示
圖像n

投影不是唯一的n不止一
張圖像可能具有相同的投影

n將二值圖像投影到一條直線上

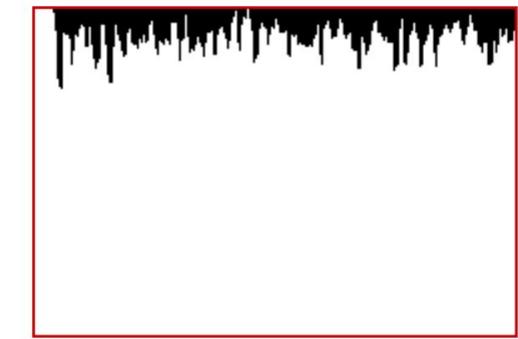
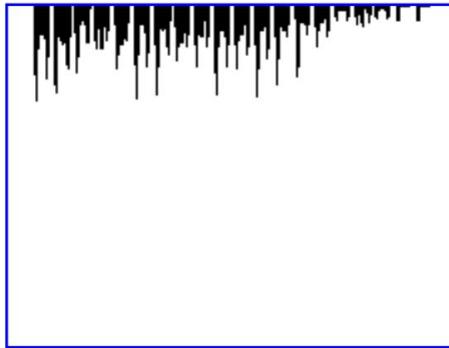
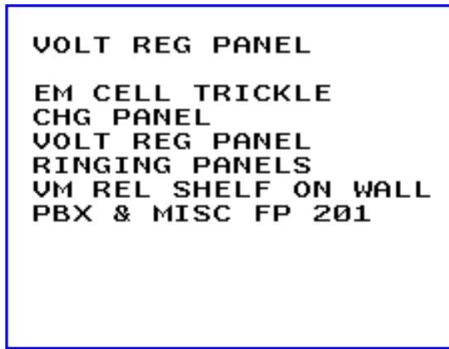
n將線劃分為 bins 並找到垂直於每個 bin 的線上的 1 像
素的數量n沿行的投影H[i]和投影

沿二值圖像列的V[j]由下式給出

$$\text{嗨}[j] = \sum_{i=1}^m A(i, j), \quad V[j] = \sum_{i=1}^n A(i, j)$$

預測

n雙向投影圖 (圖2.4、2.5、2.6)



following:

$$X \triangleq \{x_B, y_B\}$$

where an edge between
indicate the length of

預測和位置

n可以定義在任何直線上的一般投影 **n** 圖像的一階矩等於其投影的一階矩（為什麼？） **n** 物體位置的計算只需要一階矩

n可以根據水平和垂直投影計算位置

$$AVj \frac{\text{米}}{j = 1\text{個}} \quad \text{啊} \quad \text{啊} [] \quad n \quad [], \quad - = \frac{jVj []}{A}, \quad - = \frac{\text{å iH } []}{A}$$

遊程編碼

n用於圖像傳輸n使用數字表示

圖像中 1 像素的遊程長度n兩種常用方法： n每行 1
的起始位置和遊程

長度

被使用

n僅使用試驗的長度，從試驗的長度開始

1 次跑步

遊程編碼

n二值圖像：

1個	1個	1 0 0 0 1			1 0 0 0 1				1個	1個	1 0 1			1 0 1			1個	1個
0 0 0 0 1				1個	1個	1個	1個	1個	1個	1個	1個	1個	1個	1 0 1			1個	1個
1個	1個	1 0 0 0 0 0 0 0 0 0 0 0 0 0 1												1個	1個	1個	1個	1個

n 1 次運行的開始和長度：

n (1,3) (7,2) (12,4) (17,2) (20,3)

n (5,13) (19,4)

n (1,3) (17,6)

n長度1 和 0 次運行： n

3,3,2,3,4,1,2,1,3

n 0,4,13,1,4

n 3,13,6

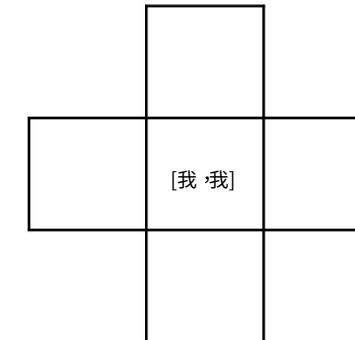
鄰居

n在數字圖像中，一個像素與四個像素有一個公共邊界，並與另外四個像素共享一個角**n**如果兩個像素共享一個公共邊界，則它們是4個相鄰像素**n**如果兩個像素共享至少一個角，則它們是8個相鄰像素**n**位置[i, j]的像素有4個鄰居[i+1, j], [i-1, j], [i, j+1],

[我, j-1]

n像素的8-neighbors包括4-neighbors加上[i+1, j+1], [i+1, j-1], [i-1, j+1], [i-1, j-1] **n**一個像素被稱為與

其4個鄰域4連接和與其8個鄰域8連接

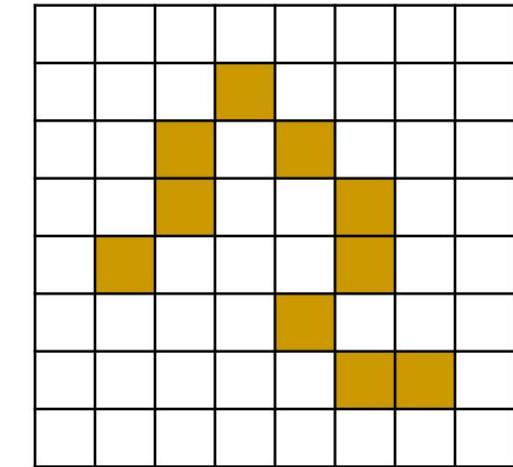
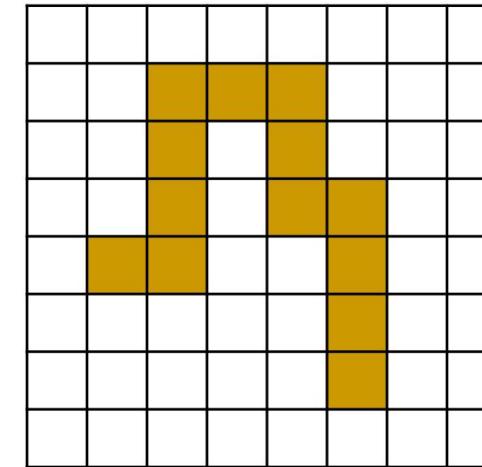


小路

n從 $[i_0, j_0]$ 處的像素到 $[i_n, j_n]$ 處的像素的路徑

是一系列像素索引 $[i_0, j_0], [i_1, j_1], \dots, [i_n, j_n]$ ，使得 $[i_k, j_k]$ 處的像素是 $[i_{k+1}, j_{k+1}]$ 處像素的鄰居] for all k with $0 \leq k \leq n - 1$ **n**如果鄰居關係使用4-connection，路

徑是4-path；如果鄰居關係使用8-connection，則路徑為8-path



周長

n一個區域的周長P的長度是一個全局屬性**n**一個沒有孔洞的區域的周長的定義是它的內部邊界像素的集合**n Perimeter**：

$$P4 = \{(r,c) \in R \mid N8(r,c) - R^1 \neq\}$$

$$P8 = \{(r,c) \in R \mid N4(r,c) - R^1 \neq\}$$

(檢查一個簡單的圖像。) **n**計

算長度|P|對於周長P， P中的像素必須按順序排列 $P = \{ (r_0, c_0), \dots, (r_{k-1}, c_{k-1}) \}$ **n**周

長： $|P| = |\{k \mid (r_{k+1}, c_{k+1}) \in N4(r_k, c_k)\}| + 21/2 |\{k \mid$

$(r_{k+1}, c_{k+1}) \in (N8(r_k, c_k)$

$- N4(r_k, c_k))\}|$

圓度n圓度 (或

緊湊度)可以定義為周長平方的長度除以面積

$2 / AC1 = |P| n$ 在此定義中，它具有數字八邊
或菱形形

的最小值，具體取決於使用的是 4 鄣域還是 8 鄣
域n越小越好！



n圓度也可以定義為 $C_2 = \mu R / sR$, 其中 μR 和 sR 是距形狀質心的距離的平均值和標準偏差

n平均徑向距離 :

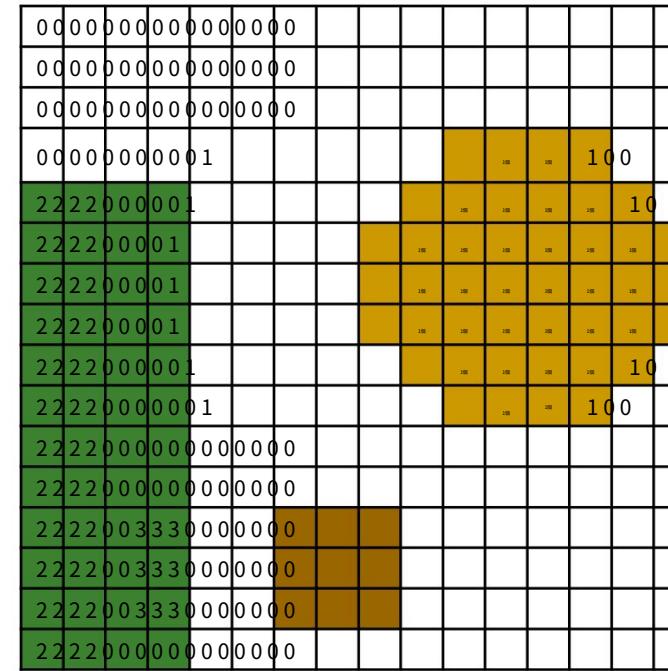
$$\mu R = \frac{1}{N} \sum_{k=0}^{N-1} \left(\overline{r_k} - \bar{r} \right)^2$$

n徑向距離標準差 :

$$s_R = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} \left[\left(\overline{r_k} - \bar{r} \right)^2 \right] / N}$$

n越大越好 !

例子



region	region	row	column	of perimeter	circularity	圓度半徑	radius	length		
數區	中心	中心					半徑	2個	平均方差	
半	44	6個	11.5	21.2	10.2	15.4	3.33	.05		
2個	48歲	9	1.5	28	16.3	2.5	3.80	2.28		
3個	9	13	7	8個	7.1	5.8	1.2	0.04		

連通性n如果存

在從p到q的路徑完全由S的像素組成，則稱像素
 $p \in S$ 連接到 $q \in S$ n連通性是等價關係n對
於 S中的任意三個像素p、q和r，我
們有以下屬性：
 n 反射率：像素p連接到p
 n 交換性：如果p連
接到q，則q連接

至

n 傳遞性：如果p連接到q，q連接到r，則p連接到r n一組像素，
其中每個像素都連接到所

有其他像素，稱為連接分量

Foreground, Background n

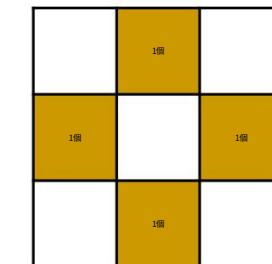
Foreground

n 圖像中所有 1 個像素的集合稱為前景，記為 S_n 被稱為背景

n 的所有其

他組成部分被稱為空洞 n 對象和背景應使用不同的連通性 n 如果 8-連通性用於 S ，則 4-連通性應用於

[n](#) (為什麼 ? 第 43 頁)



Boundary, Interior, Surrounds n

Boundary

n S 的邊界是 S 的像素集合

在 n 中有 4 個鄰居的 S

邊界通常表示為 $S \setminus S^c$ n 內部 n 內部是 S 的像素集

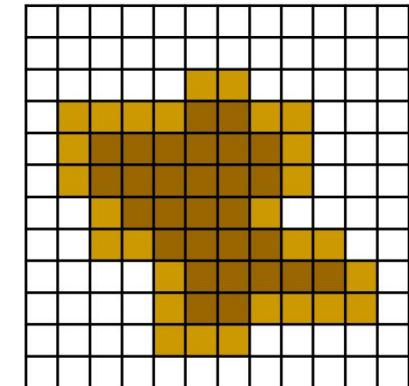
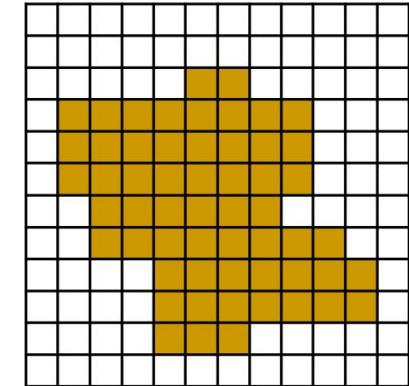
合

不在它的邊界 n S 的內部

是 $(S - S^c) \cap$ Surrounds n 區域

T 包圍區域 S (或

S 在 T 內部), 如果從 S 的任何一點到圖像邊界的任何 4 路徑必須相交 T



組件標籤

n唯一標記每個正連通分量簇n零元素被認為是背景的一部分並保持為零

n兩種算法： n遞歸算法（非常低效，僅在並行機上使用） n順序

算法n遞歸算法： n

掃描圖像找到一個未標記的 1 像素並為其分配一個新標籤

大號

n遞歸地為它的所有 1 個鄰居分配一個標籤 L n如果沒有更多未標記的 1 像素則停止n轉到步驟 1

順序算法（標記）

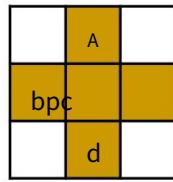
n 通過光柵掃描標記 4-connected 組件

(逐行從左上角開始) n 如果 p =

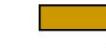
0 則忽略 n else

如果 a 和 b 沒有標記，增加標籤和標籤 p n else 如果只有 a 或 b 中的一個被標記然後複制標籤到 p n else 如果 a 和 b 被標記

n 如果 a 和 b 標記相同則將標籤複製到 p n 否則將任一標籤複製到 p 並記錄標籤的等價性



..	101
..	000
..	0	10	..
0000			



..	102
..	000
..	0	..	0
0000			

計算圖像中的對象

n用於計算前景對象 **n**外角圖案(E) 是具有

三個 0 和一個 1 像素的 $2'2$ 掩碼 **n**內角圖案(I) 是具有三個 1 和一個 0 像素的 $2'2$

掩碼

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

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

n算法 count_objects: **n** E = 0,

I = 0 **n** For L

= 0 to row_number

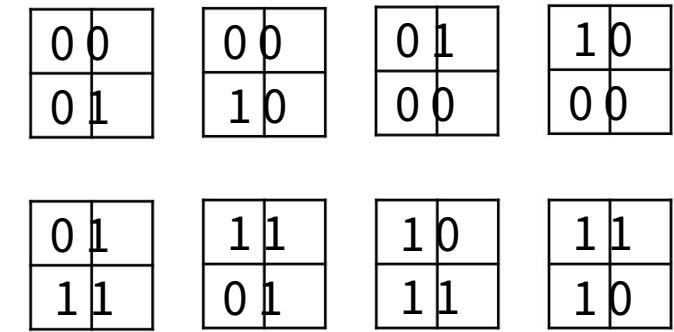
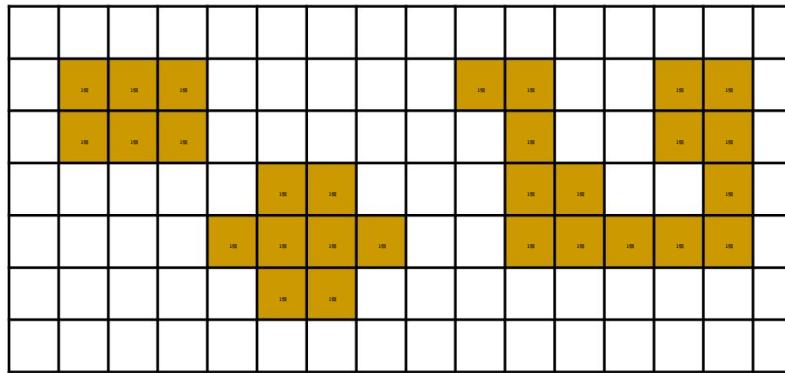
n For P = 0 to column_number

q If external_match(L,P) then E = E

 + 1 **q** If internal_match(L,P) then I

= I + 1 **n** Return (EI)/4

計算前景對象



e 的數量 :21 i 的數
量 :9 對像數量 =
 $(21-9) / 4 = 3$ (為什麼?)

例子

n元件標示

4-連接



8連



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

=

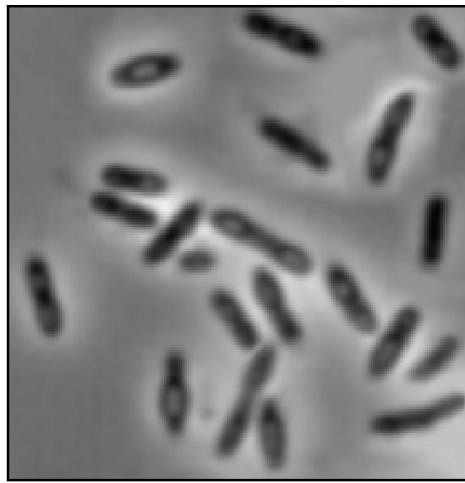
0	0	0	0	0	0	0	0
0	0	2	2	2	2	0	0
0	0	2	2	2	2	0	0
0	0	2	2	2	2	0	0
0	1	0	0	0	0	0	0
0	1	0	0	3	3	3	3
0	1	0	0	3	3	3	3
0	0	0	3	3	0	0	0

=

0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	0	0	0
0	1	0	0	2	2	2	2
0	1	0	0	2	2	2	2
0	0	0	2	2	0	0	0

例子

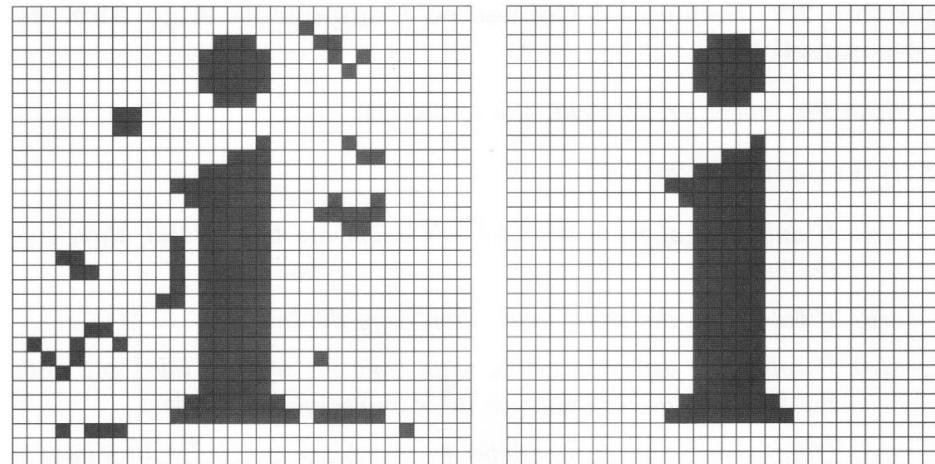
n物體計數



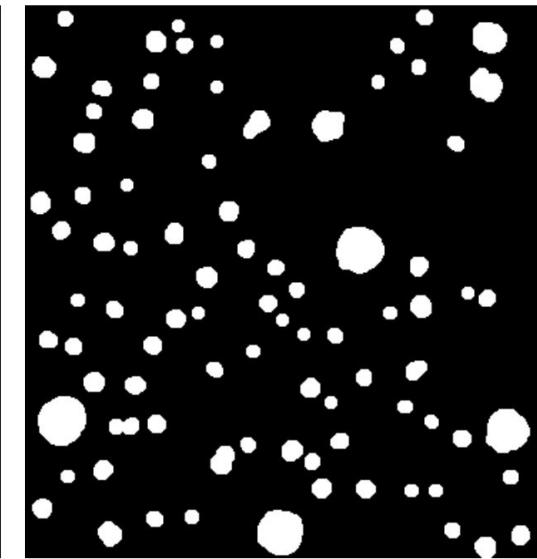
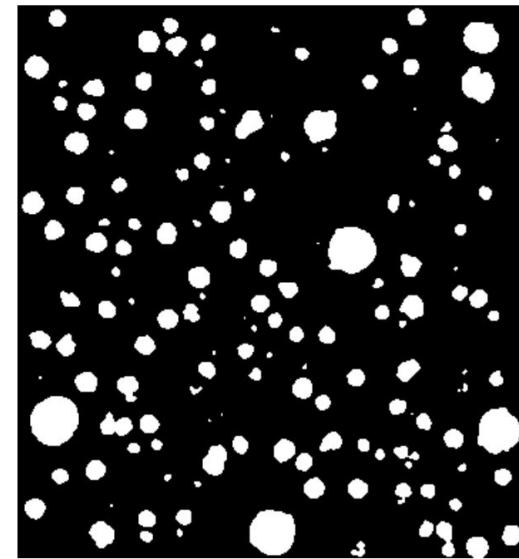
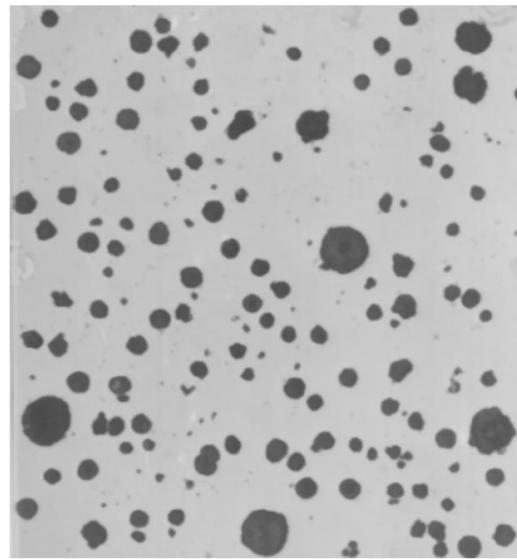
大小過濾

在二值圖像中，通常噪聲區域很小。如果對象的大小大於 T_0 像素，則可以在組件標記後使用大小過濾器去除噪聲。

尺寸小於 T_0 的所有組件通過將相應像素更改為0來移除。



例子



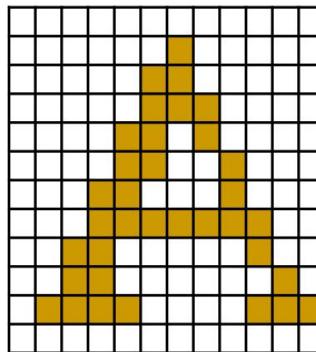
歐拉數

n虧格數或歐拉數可以用作一個特徵

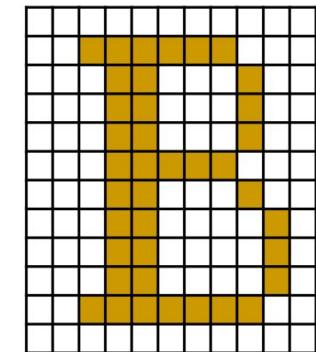
目的

n類型定義為組件數減去孔數： $E = C - H$

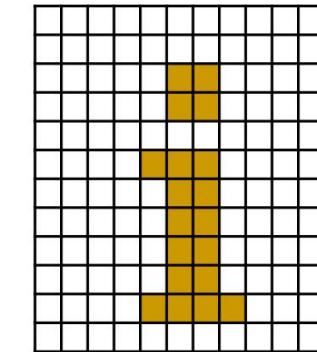
n Genus 提供了一個簡單的拓撲特徵，對平移、旋轉和縮放具有不變性



$$E = 0$$



$$E = -1$$



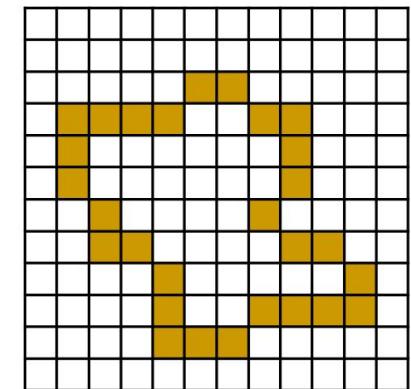
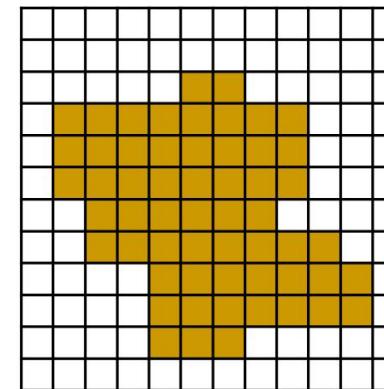
$$E = 2$$

Region Boundary n 連通分

量 S 的邊界 是 S 中與相鄰的像素集

n 在大多數應用中，我們希望以特定順序 跟踪 邊界上的
像素

n 邊界 跟隨 算法 選擇一個開始
像素 $s \in S$ 並 跟踪 邊界 直到它回到起始像素

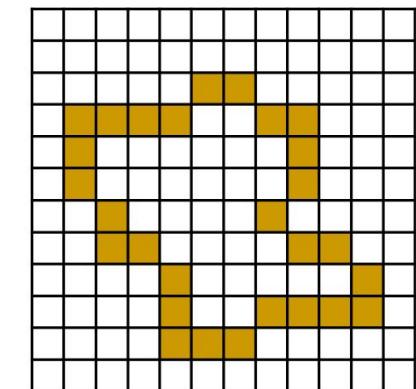
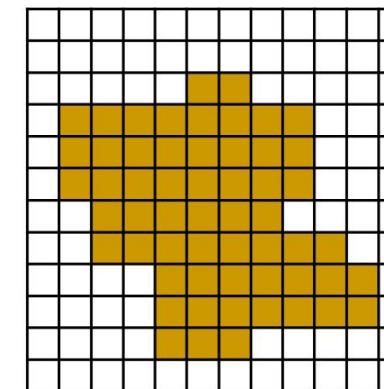


區域邊界

n邊界跟蹤算法**n**通過光柵掃描為該區域

找到一個起始像素 $s \in S$ 設邊界跟蹤中的當前像素用 c 表示 , 設 $c = s$, 設 s 西邊的 4-neighbor 為 $b \in n$ 設 c 的八個 8-neighbors , 以 b 開始 , 順時針順序為 n_1, n_2, \dots, n_8 。

找到 n_i , 對於 S 中的第一個 i 設置 $c = n_i$ 和 $b = n_{i-1}$ 重複上面



測距

n求兩個或兩個像素之間的距離

圖像的組成部分**n**對於所有

像素p、q和r，任何距離度量必須滿足以下所有屬性：**n**

$d(p,q) \geq 0$ 且 $d(p,q) = 0$ 當且僅當 $p = q$ **n**

$d(p,q) = d(q,p)$ **n** $d(p,r) \leq d(p,q) + d(q,r)$ **n**

常見距離函數：

n歐幾里德：

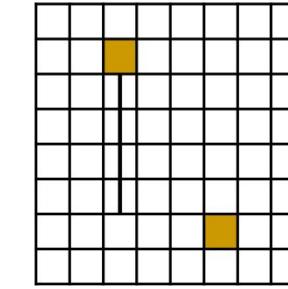
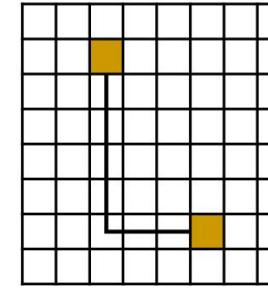
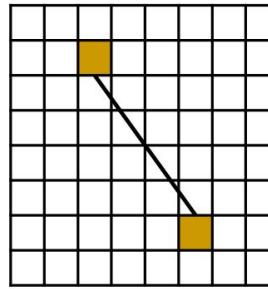
$$d_{\text{歐幾里德}}([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$$

| $d_{\text{歐幾里德}}([i_1, j_1], [i_2, j_2])$ | **n** City-block:

$$| d_{\text{City-block}}([i_1, j_1], [i_2, j_2]) | = \max(|i_1 - i_2|, |j_1 - j_2|)$$

不同的距離測量

歐氏距離 城市街區距離 棋盤距離



$$\begin{array}{cccc}
 \sqrt{8} & \sqrt{5} & 2 & 5 \\
 \sqrt{5} & 2 & \sqrt{1} & \sqrt{2} \\
 3 & 2 & & 1 & 0 & 1 \\
 \sqrt{5} & 2 & \sqrt{1} & \sqrt{2} & \sqrt{5} & 2 \\
 \sqrt{8} & \sqrt{5} & 2 & 5 & \sqrt{8} &
 \end{array}$$

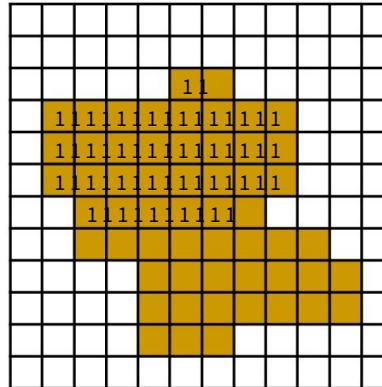
$$\begin{array}{c}
 3 \\
 2 \\
 3 \\
 2 \\
 1 \\
 0 \\
 1 \\
 2 \\
 3 \\
 3 \\
 2 \\
 1 \\
 0 \\
 1 \\
 2 \\
 3 \\
 3 \\
 2 \\
 2 \\
 2 \\
 2 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3
 \end{array}$$

$$\begin{array}{c}
 3 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3 \\
 2 \\
 2 \\
 2 \\
 2 \\
 2 \\
 3 \\
 3 \\
 2 \\
 1 \\
 1 \\
 1 \\
 2 \\
 3 \\
 3 \\
 2 \\
 1 \\
 1 \\
 1 \\
 2 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3 \\
 3
 \end{array}$$

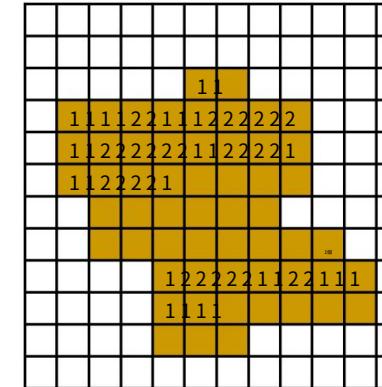
距離變換

n在某些應用中，需要物體成分的像素與背景之間的最小距離

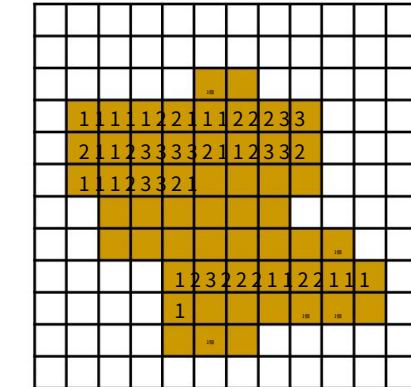
n距離變換是計算到背景區域 M 的距離，對於S中的所有像素 n $f_0[i,j] = f[i,j]$ n $f_m[i,j] = f_0[i,j] + \min(f_m[u,v])$ 其中(u,v)在(i,j)的4-neighbor 中



第 0 次通過

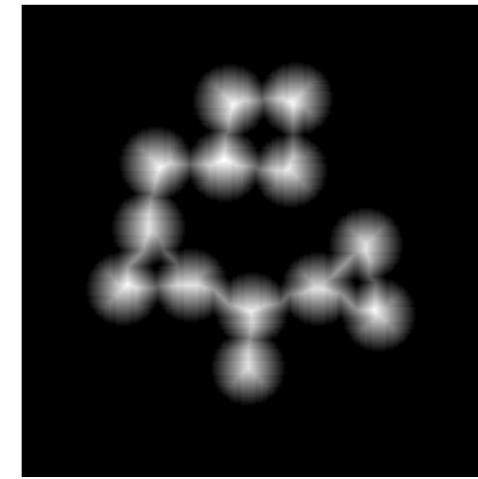
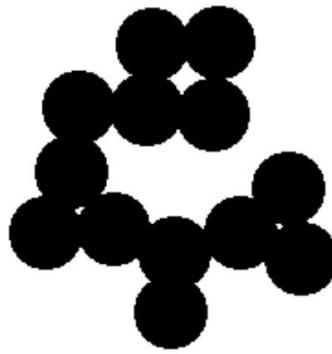


第一關



第二關

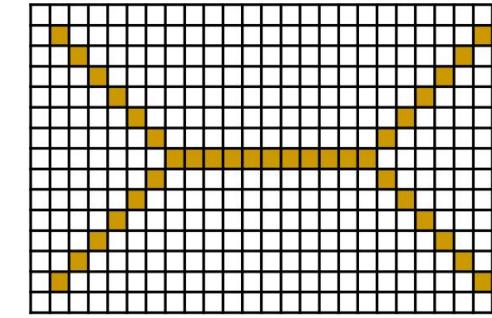
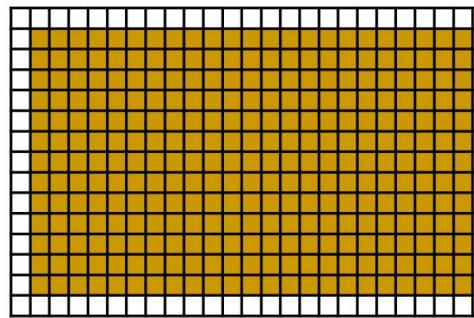
例子



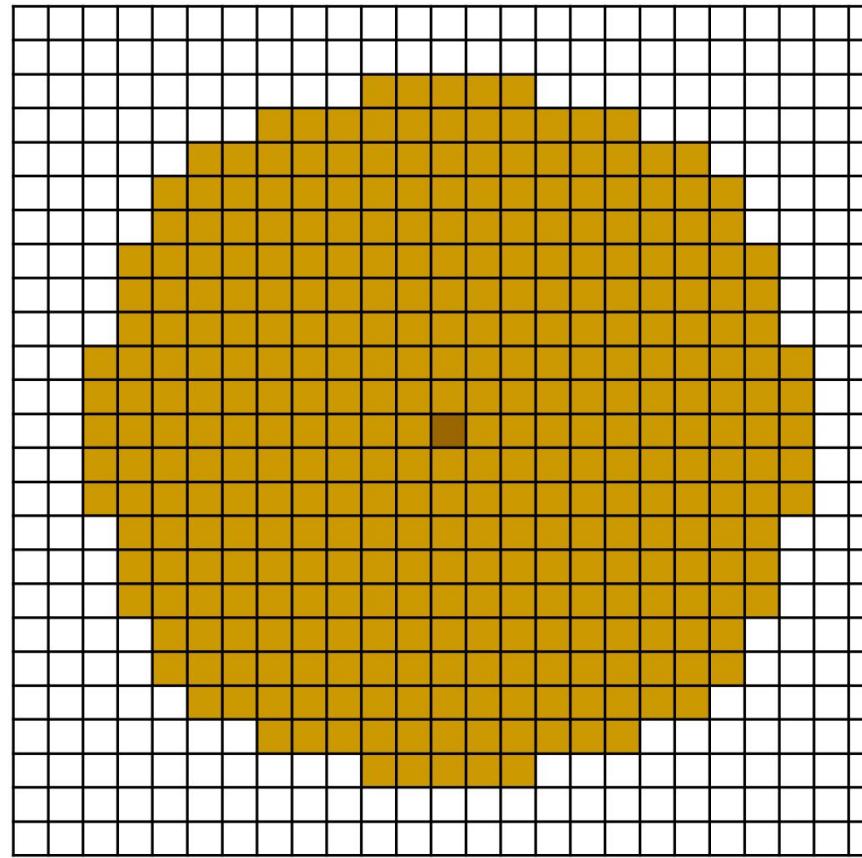
中軸

n如果 $d([i,j], \quad) \geq d([u,v], \quad)$, 則從S中的像素 $[i,j]$ 到的距離
 $d([i,j], \quad)$ 是局部最大的對於 $[i,j]$ 附近的所有像素 $[u,v]$

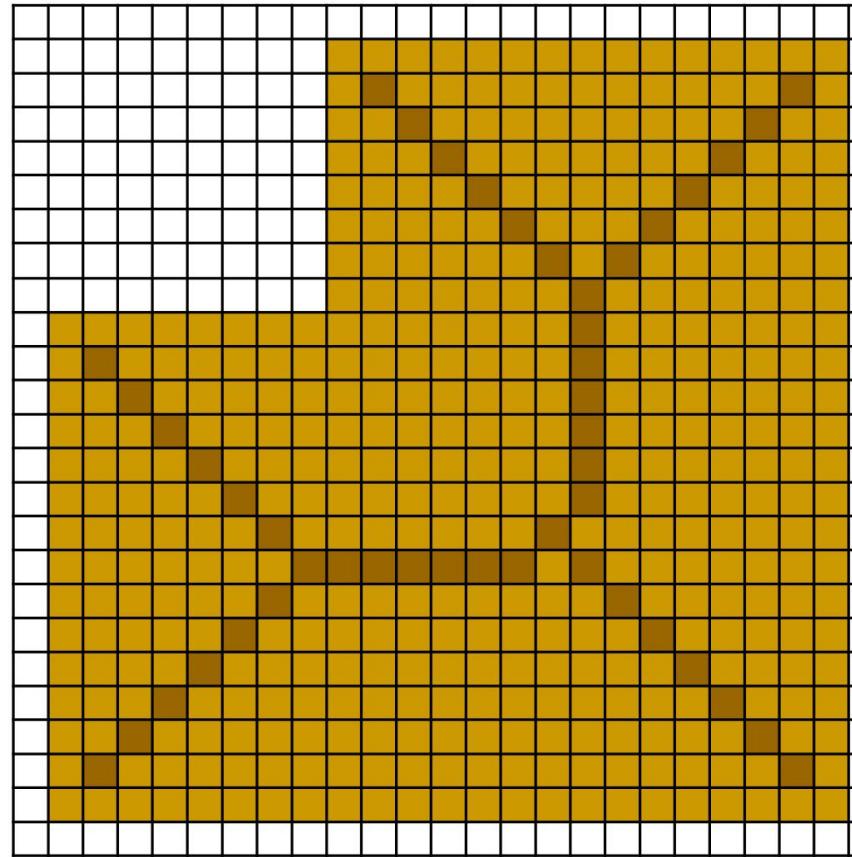
n S中與的距離為局部最大值的像素集稱為S的**骨架**、**對稱軸**或**中軸**，用 S^* 表示



例子



例子



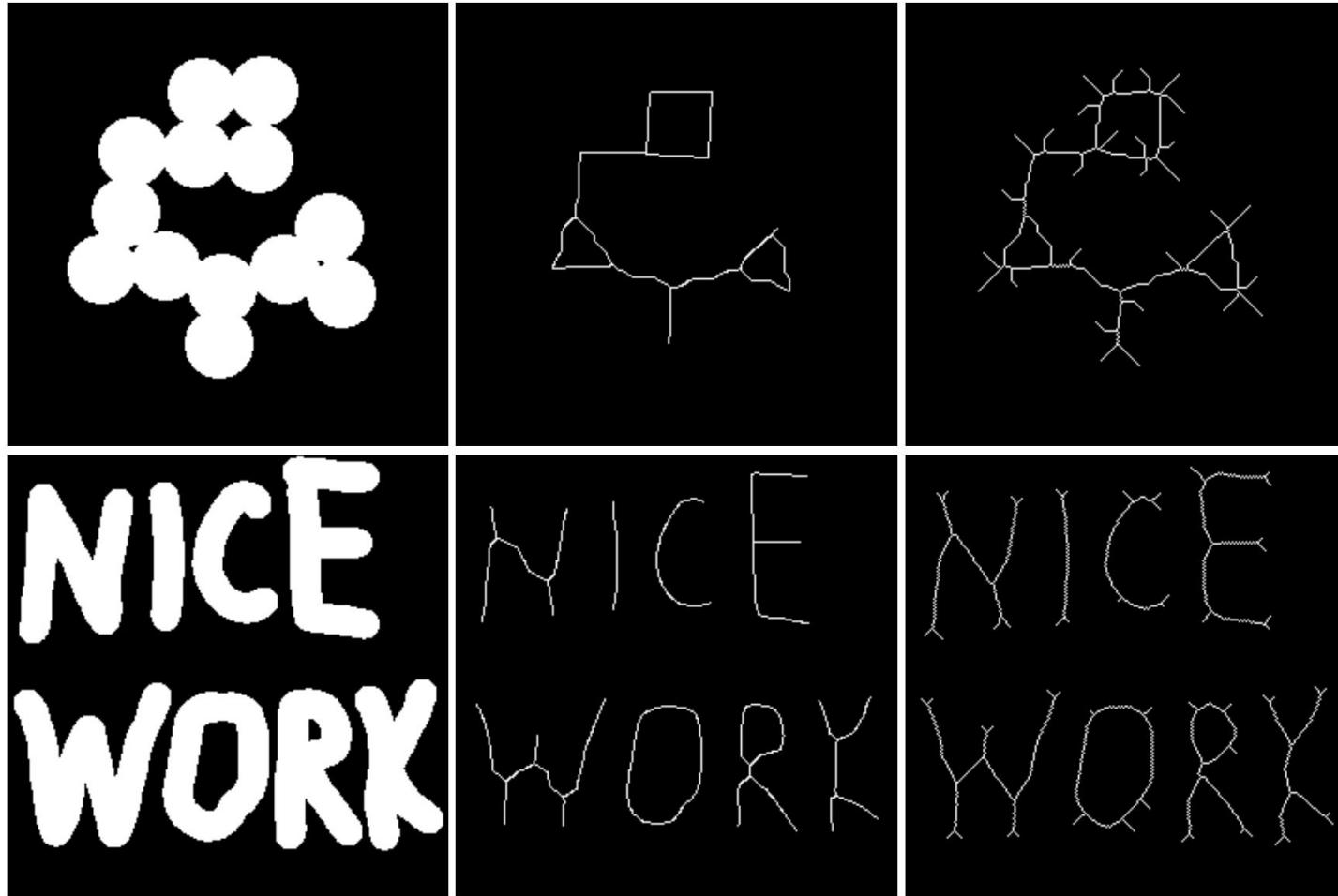
中軸

n 原始集合S可以從 S^* 和 S^* 的每個像素到的距離重建

n S^* 是S的緊湊表示 n S^* 用於表示區域的形狀 n 通過刪除 S^* 中與距離較小的像素，我們可以創建 S^* n 的簡化版本兩種表示形式 邊界 和 中軸： n 對於任意對象，邊界是區域的更緊湊表示 n 要確定給定像素是否在該區域中，

中軸是更好的表示

例子



細化n細化： n

將二值圖像區

域縮減為接近其中心線的線，也稱為骨架或核心線n將圖像成分縮減到其基本成分

信息，以便於進一步分析和識別n細化要求：

n連接的圖像區域必須細到連接線
結構

n細化結果應至少為8 連接n應保持近似的端線位置n細化
結果應接近中間線n應盡量減少由細化引起的外來支線（短
分支）

間伐

n在其至少 $3'3$ 像素的鄰域範圍內檢查圖像中的每個像素，並“剝離”區域邊界，一次一個像素，直到區域被縮小為細線 n 該過程被執行迭代地

Expanding and Shrinking n

Expanding

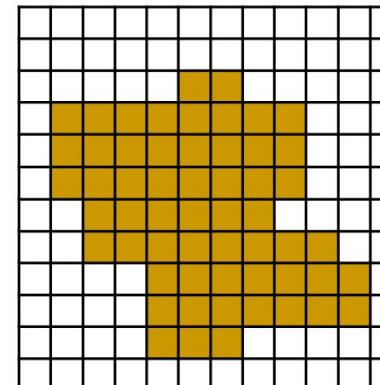
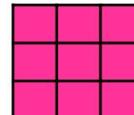
n 一個組件允許改變，這樣一些背景像素被轉換為 1 n 如果像素的任何鄰居像素從 0 變為 1

是 1

縮小 n 對象

像素被系統地刪除或轉換為 0 n 如果像素有任何鄰居，則將像素從 1 更改為 0

是 0



擴大和縮小 n 讓 $S(k)$: S 擴大 k 倍 , $S(-k)$: S 縮小 k 倍 , 然後 n

$(S_m) - n^{-1} (S_n) m^{-1} S(mn) n S \rightarrow (Sk) - k n$

$S \rightarrow (Sk) k n$ 擴展和收

縮可以用

來確定

孤立的組件和集群 (圖 2.23 和 2.24) n 擴展後收縮可
用於填充不需要的孔

n 先收縮後擴展可用於去除孤立的噪聲像素

3/20/19

n工讀生招聘 (工學院) n需

要人數 :5~6人n合作

時間 :3月中~12月n選款方

式 :計劃1人1個月選6000元n (實用時薪計
算依實驗室決定去年約250元)

n標記內容 :紅綠燈資料集路上常見類型n標記工

具 :Matlab的地圖真值標註器 (暫定) n聯繫葉天文 ,

tw950107@gmail.com , 122 實驗室

二值圖像形態學

n詞形態學是指形式和結構n在計算機視覺中，它可以用來指代區域的形狀

n數學形態學的運算最初定義為集合運算

n形態學運算符可以： n變薄、 n

變

厚、 n查

找邊界、 n查找

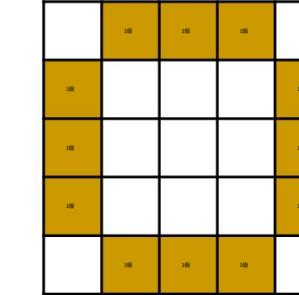
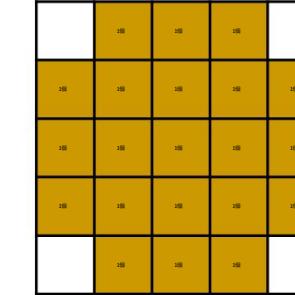
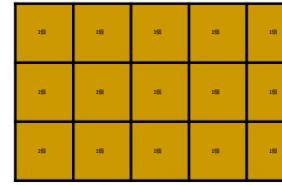
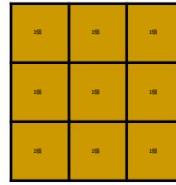
骨架（醫學軸）、 n凸包、

n等等

結構元素

n二進制形態學的操作輸入二進制
圖像 B 和**結構元素S**

n結構元素 S 通常是另一個較小的二值圖像 **n** 它表示一
個形狀 **n** 它可
以是任意大小並且
具有任意結構，可以是
由二值圖像 **n** 表示例子：



點集和符號

n二進制對像被視為點集n對於點集A和B表示：

n由x翻譯A為Ax = { ai + x | ai ∈ A} n B的反射為Br = {-bi | bi ∈ B}
n A的補碼Ac = { ai | ai ∉ A} n A和B的差為A - B
= {ci | (ci ∈ A) 異或(ci ∈ B)}

基本操作

n二元形態學的基本操作是膨脹、腐蝕、閉合和打開n膨脹擴大區域n腐蝕使區域變小n閉合操作可以關閉區域中的內部孔並消除沿邊界的海灣

n開操作可以去除小部分的剛從邊界出來進入背景區域的區域

一些應用

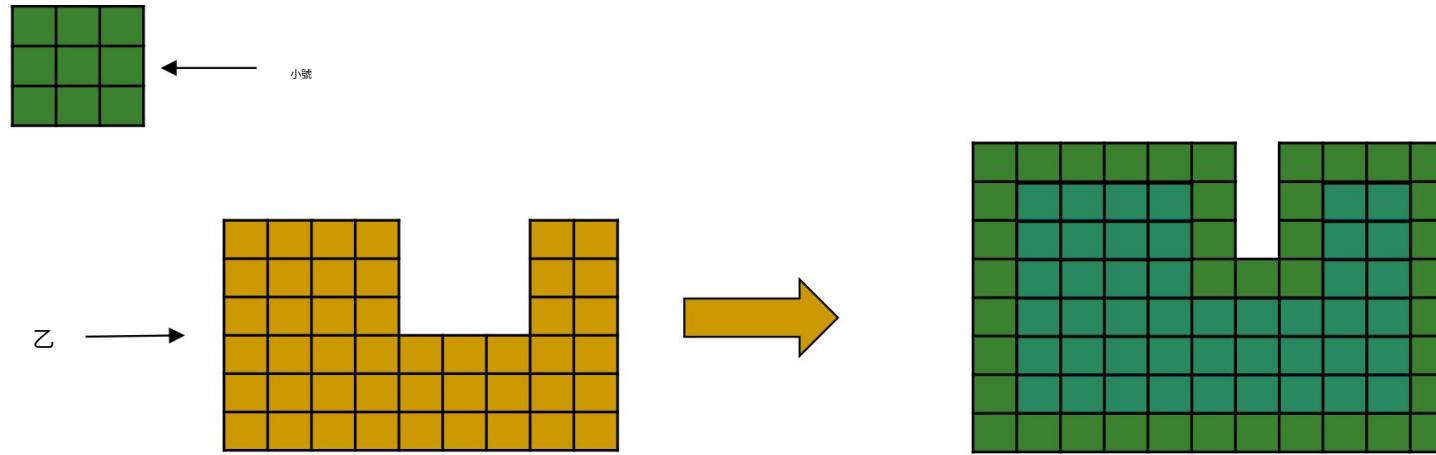
n二進制形態學可用於提取物體的原始特徵，可用於識別物體n形狀匹配系統可使用形態學

特徵檢測以快速檢測用於對象識別的基元

擴張

n通過結構元素 S 對 B進行膨脹：

$$B \circledast S = \{x \mid (\exists y \in S) (y \in B)^T\} = E_b E_B S_b$$



n示例：使用 3'3 結構元素擴展 A
B 以原點為中心

例子

Cross-Correlation Used
To Locate A Known
Target in an Image

Text Running
In Another
Direction

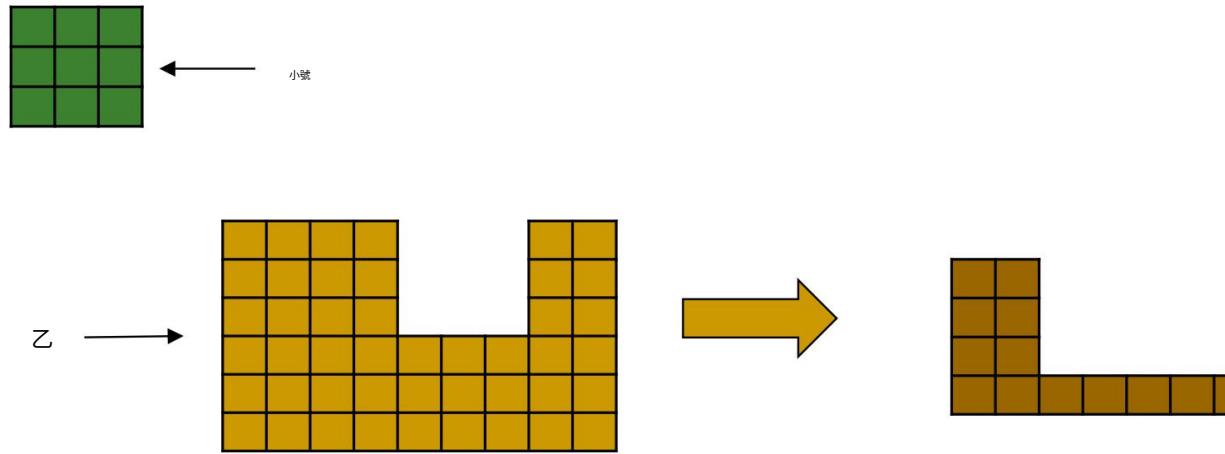
**Cross-Correlation Used
To Locate A Known
Target in an Image**

Text Running
In Another
Direction

侵蝕

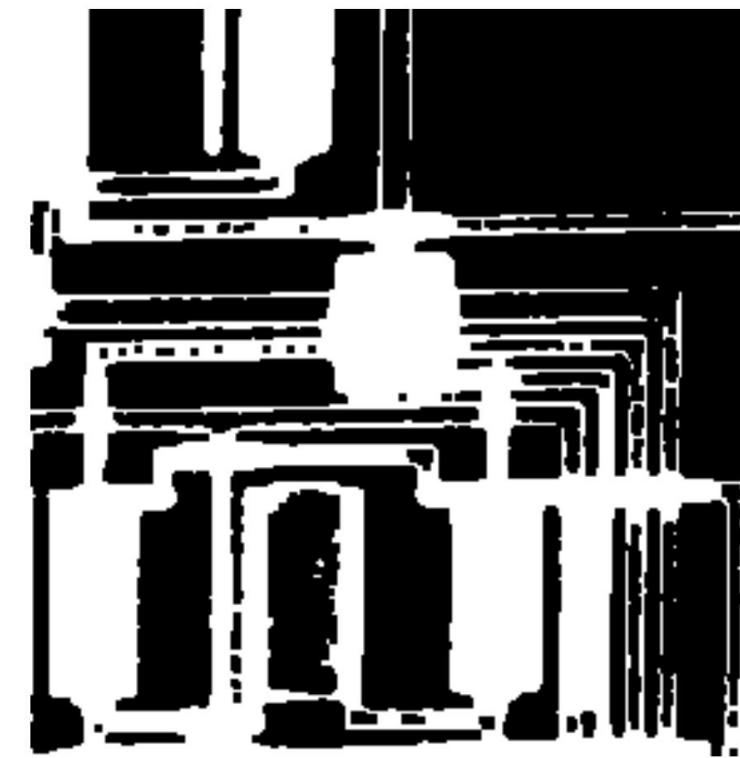
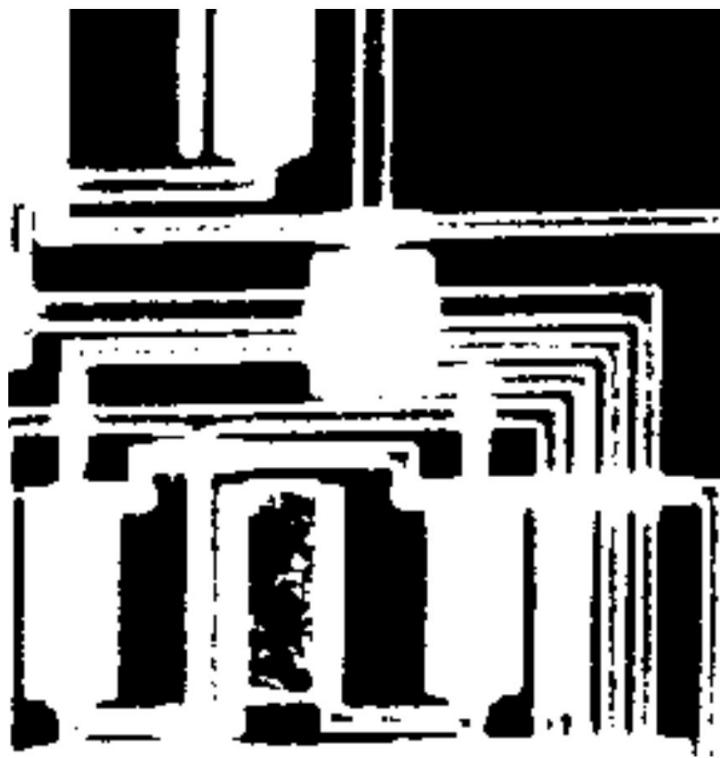
n結構元素S對B的侵蝕：

$$B - S = \{x \mid Sx \cap B\} = \{b \mid b + s \in B, \quad s \in S\}$$



n示例：用 3'3 結構元素腐蝕 A
B 以原點為中心

例子



結合膨脹和腐蝕

n結合膨脹和腐蝕 n打開n關閉n加厚n

變薄n骨

架

直觀解讀

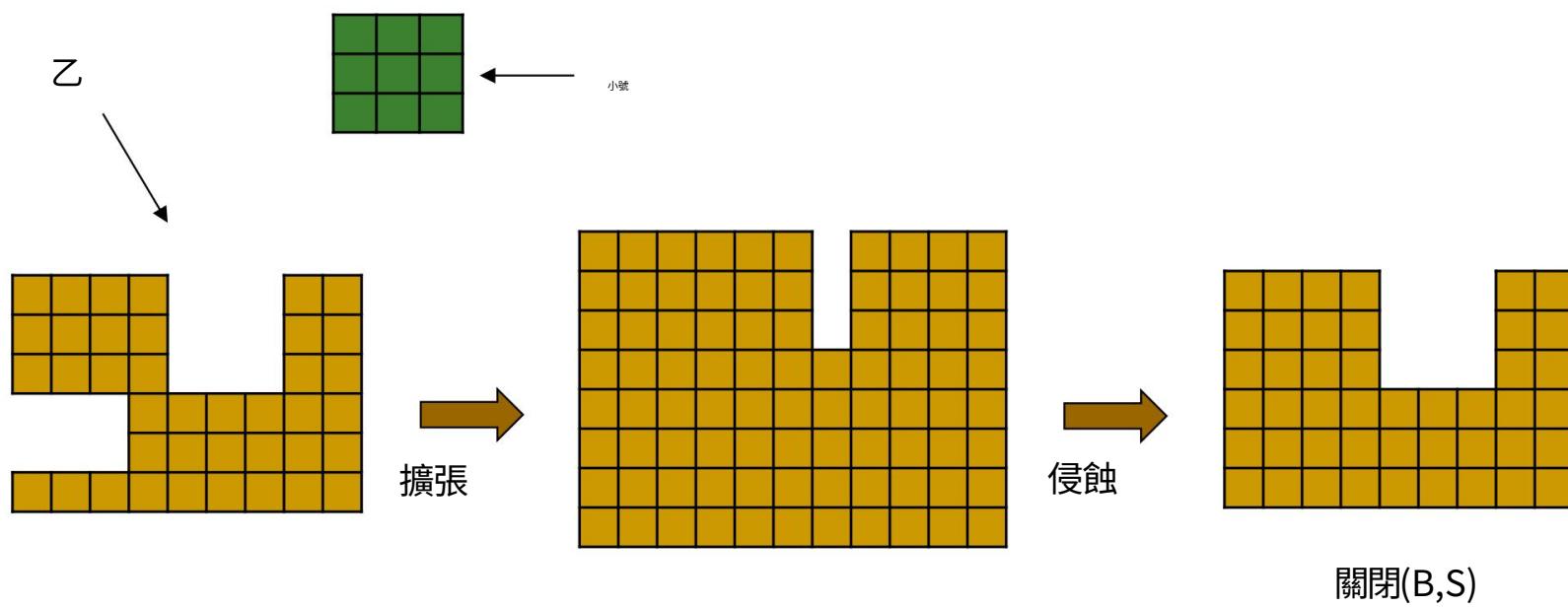
n膨脹擴大一個對象 n腐蝕收
縮一個對象 n打開 n平滑輪廓

n擴大狹窄的縫隙 n消
除細小的突起 n封閉 n填充
狹窄的縫
隙、孔洞和小裂縫

關閉

n收尾 :就像 “從外面平滑”

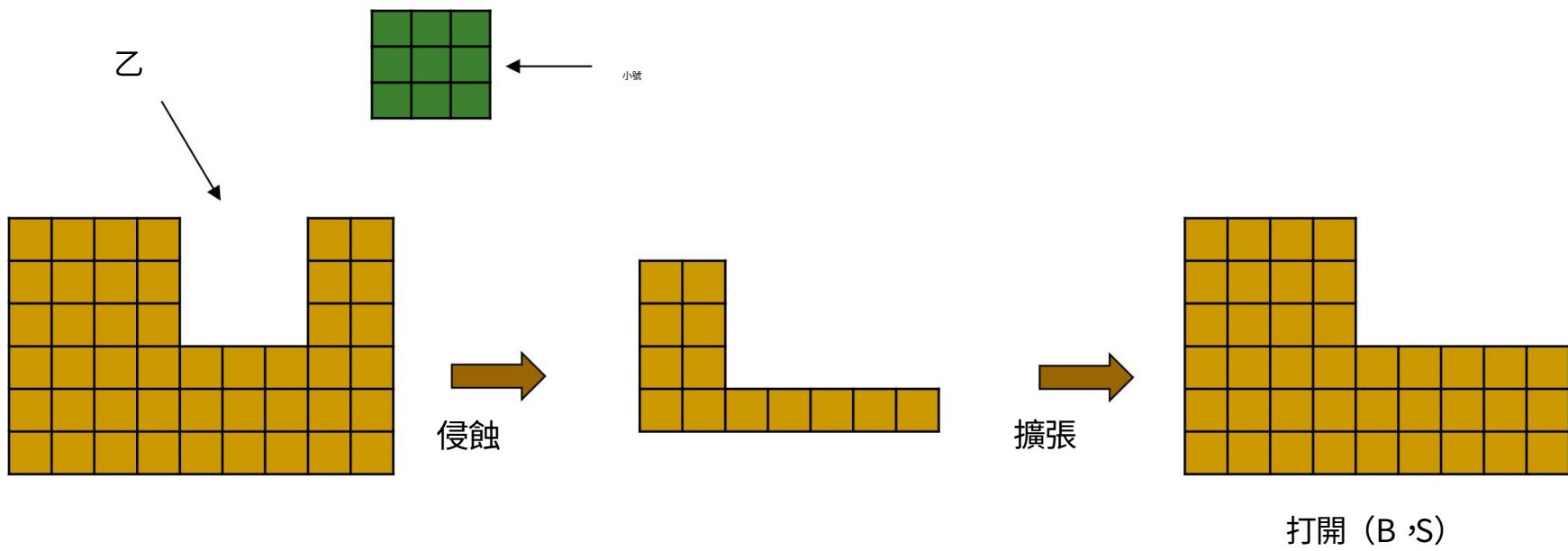
$$B \cdot S = (B \circledast S) - S$$



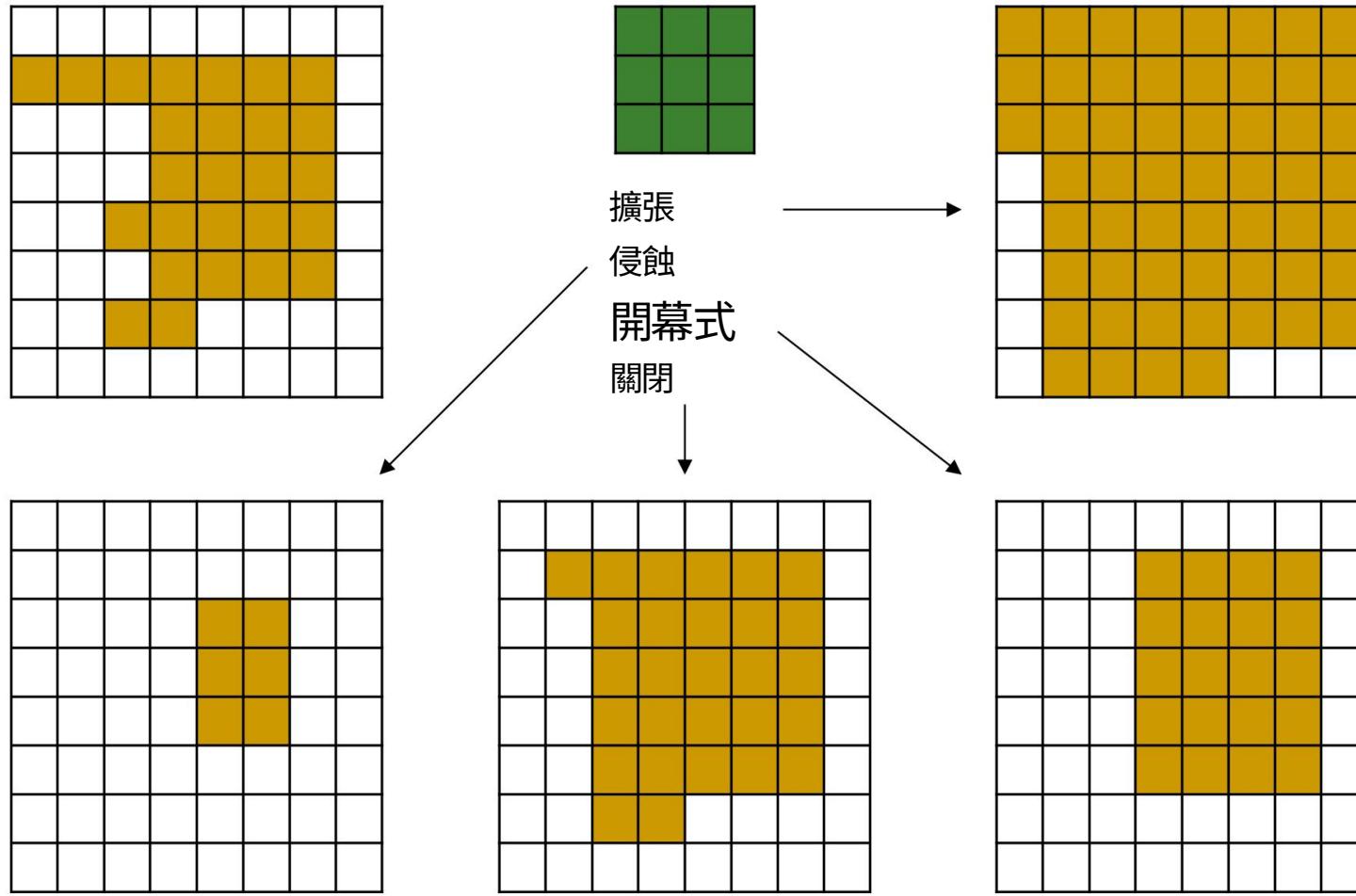
開幕式

n開放 :就像 “從內部平滑”

$$B.S = (B - S) \circledast S = \bigcup \{Sx \mid Sx \cap B\}$$



更多示例



幂等性

n多次應用打開或關閉沒有進一步的影響

n打開(打開(A,B),B) = 打開(A,B) n關閉

(關閉(A,B),B) = 關閉(A,B)

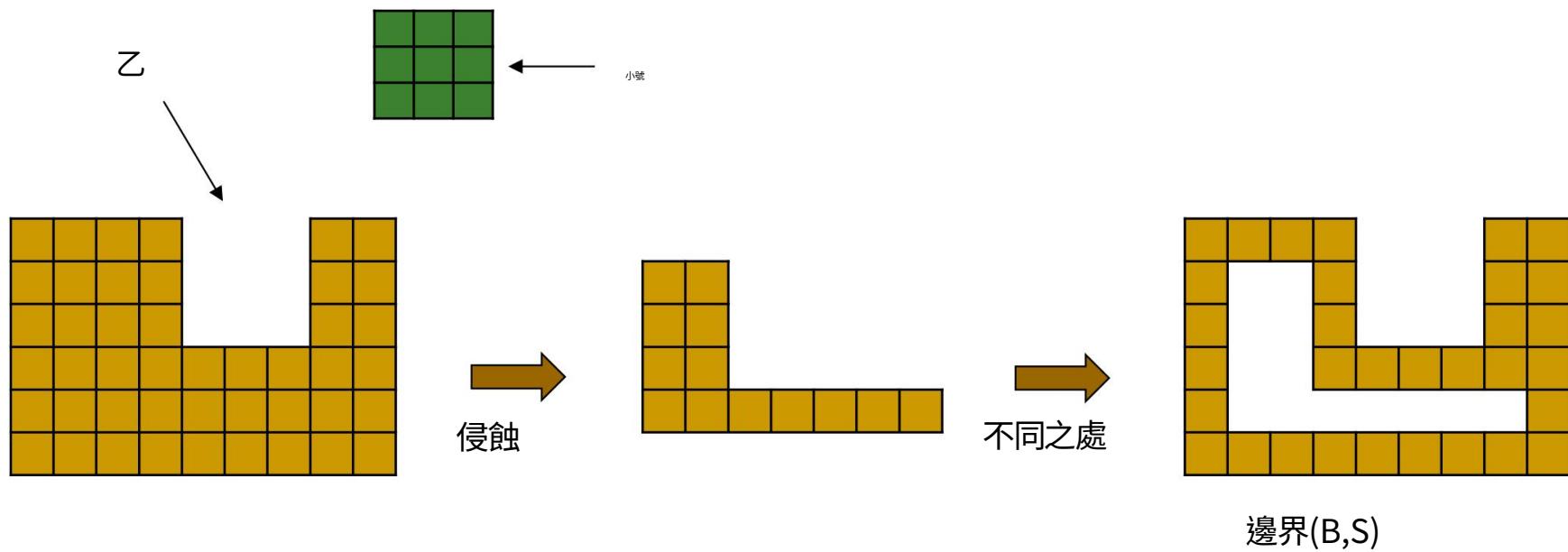
額外的結構化操作

n尋找對象的邊界n區域填充n

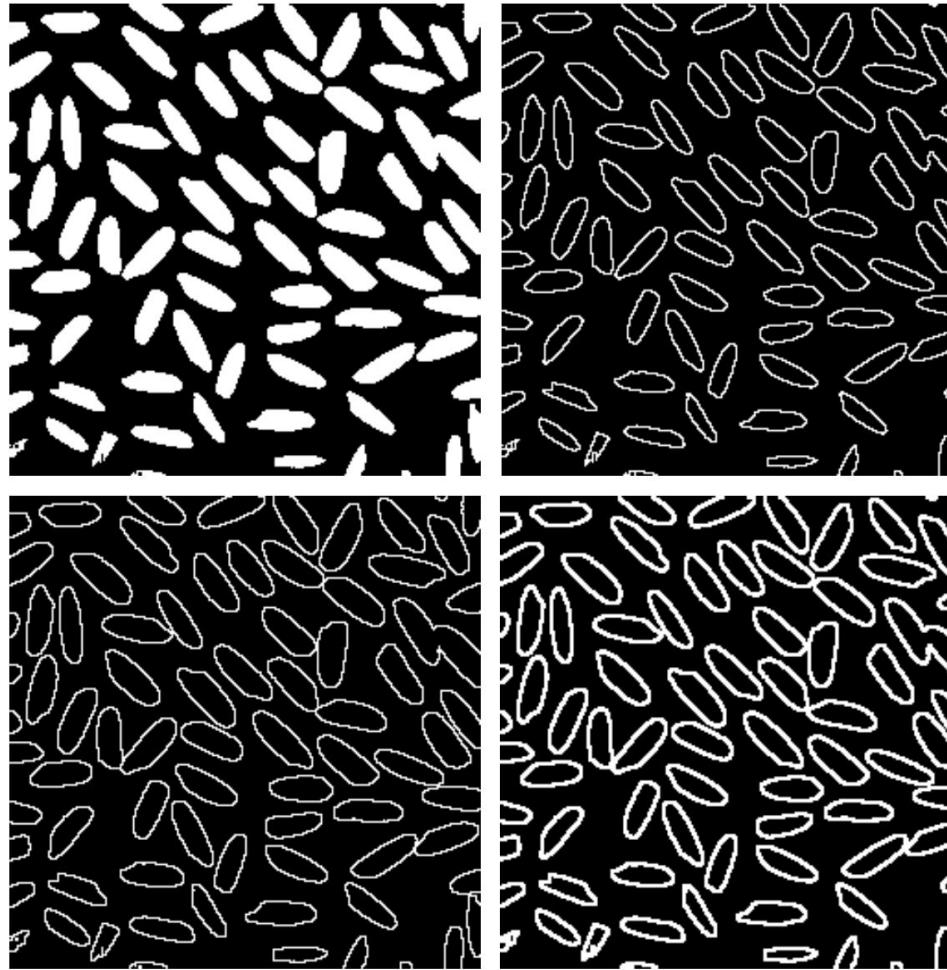
骨架

尋找邊界

$$\text{邊界}(B, S) = B - B \ominus S$$



例子

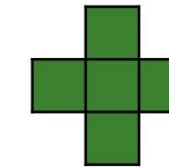


區域填充

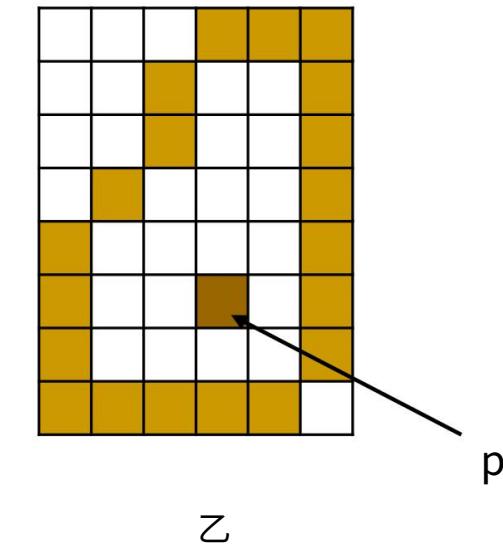
n問題：

n用 1 填充 8-連通邊界 A 紿定內部點
邊界 p

小號



n使用結構化元素 S ,以及n迭代膨脹



n路口

n補語

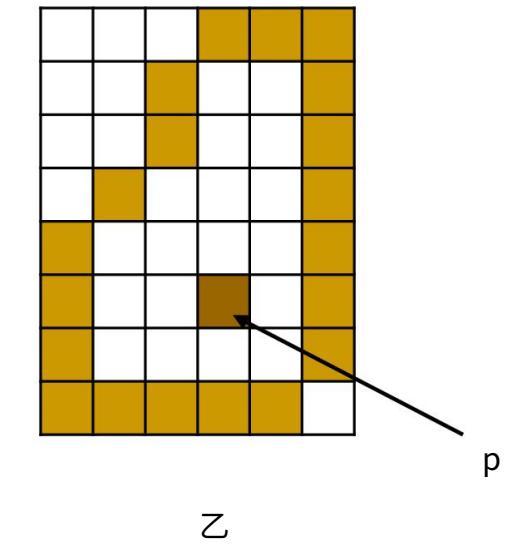
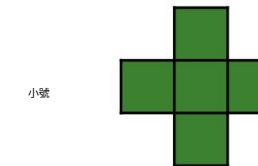
區域填充

n讓 $C_0 = p$ n計算

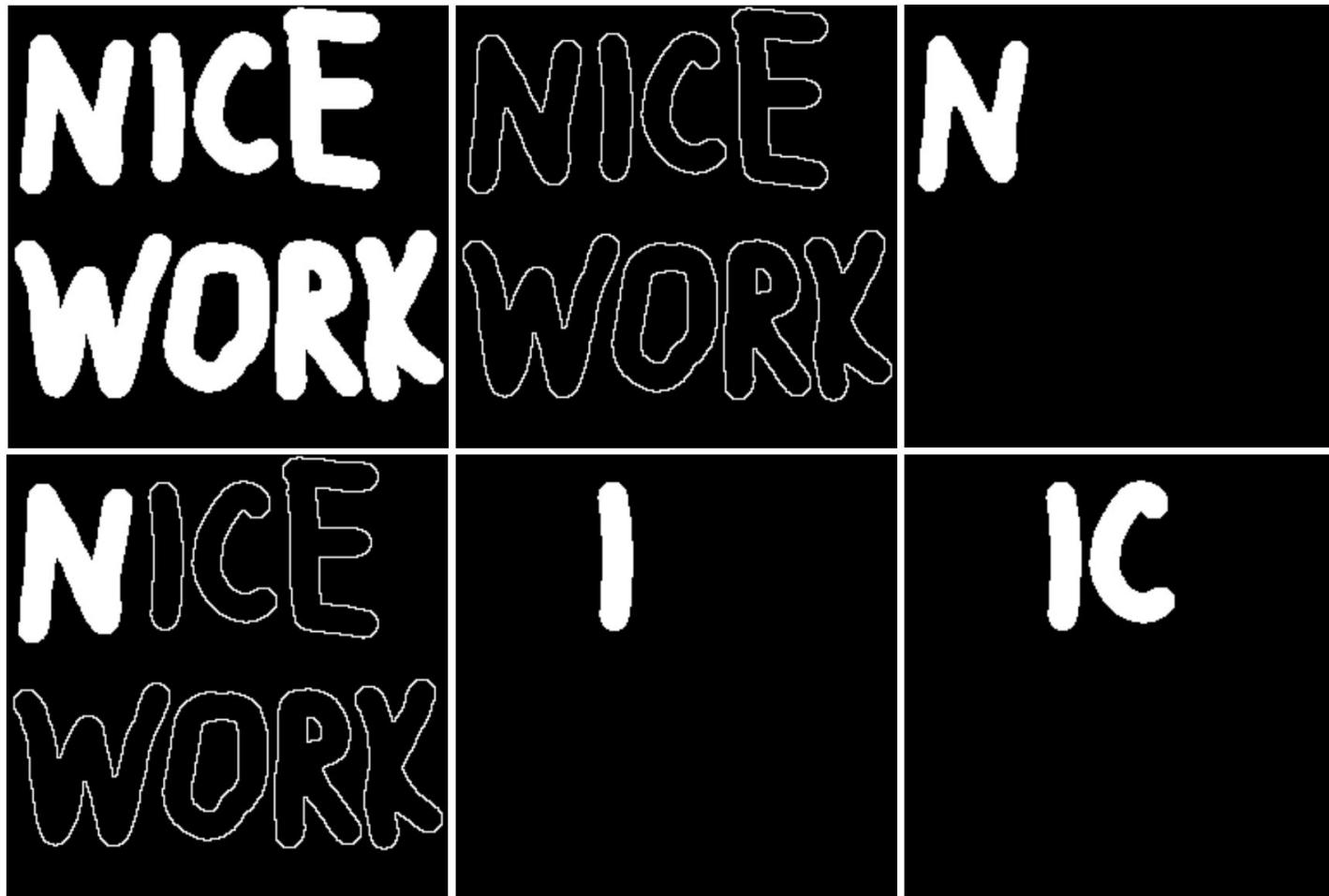
$$C_k = (C_{k-1} \cup S) \setminus B_c \quad , \text{對於 } k = 1, 2, \dots$$

n當 $C_k = C_{k-1}$ 時停止n C_k 是

B 的內部



例子



閱讀

n Jain 書的第 2 章