

# 画像情報システム

第5回 2値化画像(膨張・収縮と線図形化处理)

木更津高専情報工学科 和崎

# 1. 近傍と隣接

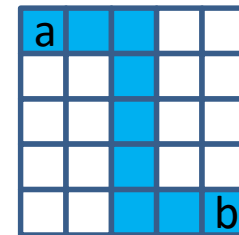
- 近傍: 注目画素の周辺にある画素の定義
  - 4近傍: 注目画素の上下左右にある計4画素
  - 8近傍: 注目画素の上下左右斜めにある計8画素
- 隣接: 2つの画素が接しているときの定義
  - 4隣接 (4近傍にあるとき)、8隣接 (8近傍にあるとき)
- 連結: 2つの画素が4隣接または8隣接で切れ目なく接続されていること
  - 4隣接のみで接続: 4-連結、8隣接があるとき: 8-連結

$x_3$ $(i, j)$	$x_2$ $(i, j)$	$x_1$ $(i, j)$
$x_4$ $(i, j)$	$x$ $(i, j)$	$x_0$ $(i, j)$
$x_5$ $(i, j)$	$x_6$ $(i, j)$	$x_7$ $(i, j)$

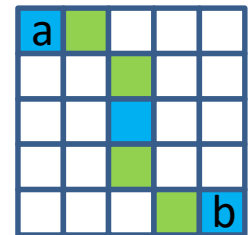
注目画素:  $x$

4近傍:  $\{x_0, x_2, x_4, x_6\}$

8近傍:  $\{x_0, x_1, x_2, x_3,$   
 $x_4, x_5, x_6, x_7\}$



4-連結



8-連結

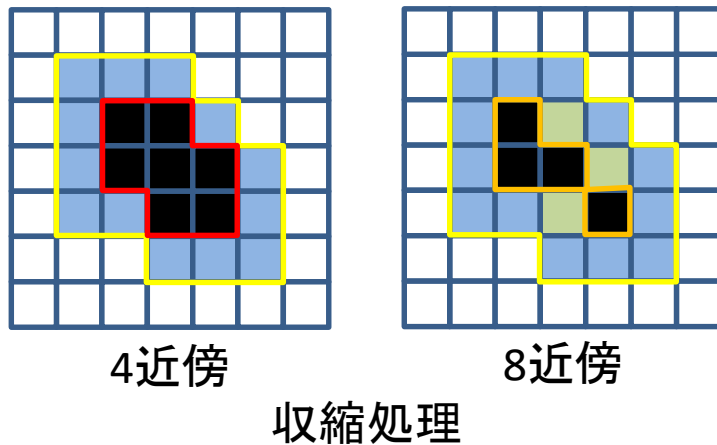
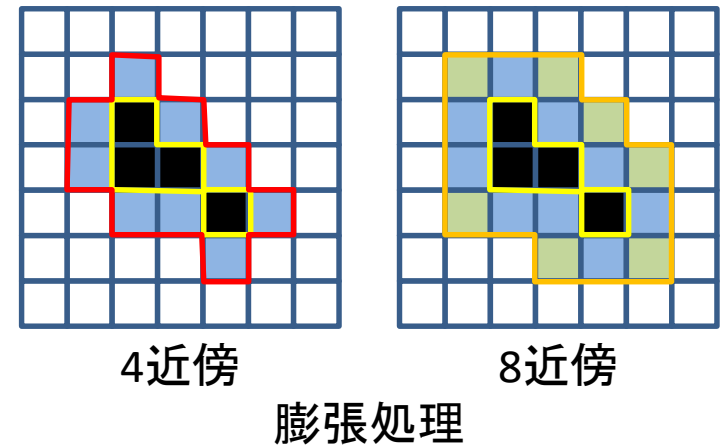


: 1画素

: 0画素

## 2. 膨張と収縮(1)

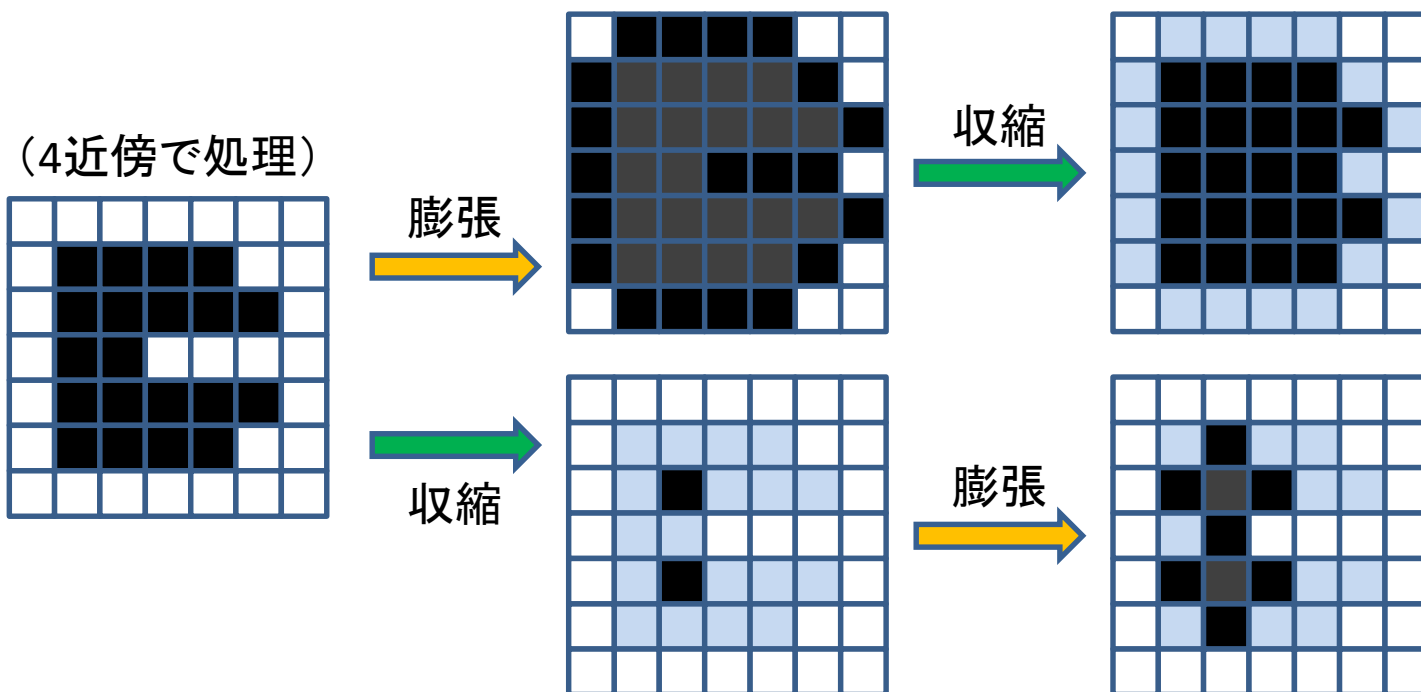
- 膨張
  - 1画素を1層分太くさせる処理
  - 0画素にとっては、収縮処理
- 収縮
  - 1画素を1層分細くさせる処理
  - 0画素にとっては、膨張処理



	: 処理前の1画素群
	: 4近傍処理後の1画素群
	: 8近傍処理後の1画素群

## 2. 膨張と収縮(2)

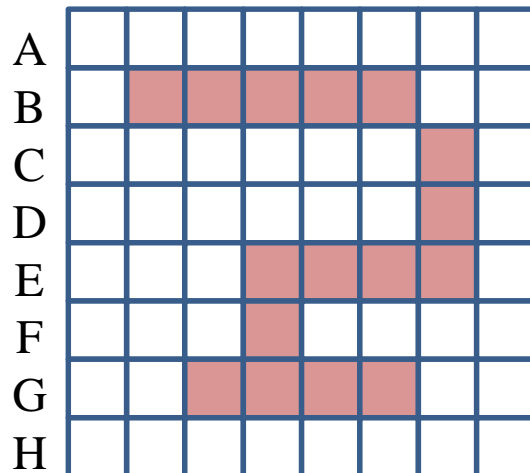
- 膨張と収縮の組み合わせ
  - 処理の順番で結果が全く異なることが多い
  - 膨張→収縮: 細い溝を埋める
  - 収縮→膨張: 細い枝の除去
  - 4近傍か8近傍で収縮・膨張
  - 収縮・膨張を複数回行うこともある



### 3. 2値化図形の表現(1)

- ランレングス表現

- 2次元の画像を1次元(一方向)に分解
  - 1次元に分解した各データを1次元状に繋ぎ直して符号化も
- 画素値とその連続する長さ(ラン)を並べて記録、符号化
- 白地が圧倒的に多く、連続した黒が現れるような画像に有利
  - FAXなど



A～Hを各々1次元化

→ (0,8)  
→ (0,1),(1,5),(0,2)  
→ (0,6),(1,1),(0,1)  
→ (0,6),(1,1),(0,1)  
→ (0,3),(1,4),(0,1)  
→ (0,3),(1,4),(0,2)  
→ (0,2),(1,4),(0,2)  
→ (0,8)

A:B:C:D:E:F:G:Hを1次元化


(0,9),(1,5),(0,8),(1,1),(0,7),  
(1,1),(0,4),(1,4),(0,4),(1,1),  
(0,6),(1,4),(0,10)



左右の白地が結合される  
分だけ、データが縮小

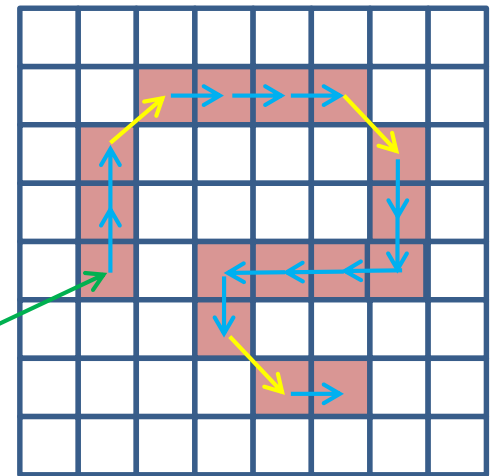
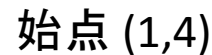
### 3. 2値化図形の表現(2)

- チェインコード表現

- 線図形(細線化された図形)や2値化図形の境界の符号化
  - 任意点を始点とした繋がりを方向符号化する
    - 一般的には8方向量子化符号
    - 4方向を優先しないと角の画素が欠ける
    - ループや分岐などの考慮が必要
- 

3	2	1
4	注目画素	0
5	6	7

## 8方向量子化符号



## 始点の座標

(1,4,2,2,1,0,0,0,7,6,6,4,4,4,6,7,0)

## チェーン符号化

# 4. 線図形化処理

- 細線化処理
  - 2値化図形から、**線幅1の中心線を抽出**する処理
    - 途中で切れたりしないこと、不必要なひげが生じないこと、縮まないこと、交差部で歪まないこと
  - 文字認識やパターン認識の前処理
    - 幾何学的特徴の把握、データ圧縮
- 境界線処理
  - 2値化図形の**境界線を抽出**
  - チェーンコードで表現、データ圧縮
- ハフ変換
  - **切れ切れの点をつなぐ線分を抽出**
  - 検出したい線分が方程式で表現できる場合に有用
- 最小二乗法
  - **切れ切れの点から、直線または曲線をあてはめる**処理方法
  - ばらつきの多いデータから、妥当な直線や曲線を求める

# 課題

- 課題19

- 以下の4近傍膨張・収縮処理画像を作成せよ

- 19-1: 膨張→収縮→収縮→膨張を行った処理画像を作成せよ
    - 19-2: 収縮→膨張→膨張→収縮を行った処理画像を作成せよ

- 以下の8近傍膨張・収縮処理画像を作成せよ

- 19-3: 膨張→収縮→収縮→膨張を行った処理画像を作成せよ
    - 19-4: 収縮→膨張→膨張→収縮を行った処理画像を作成せよ

- 課題20

- 入力画像をランレングス表現でテキストファイルに出力せよ

- 1行目には、提出者のユーザID(例:j09400)だけを書き込んで改行すること。
    - 2行目以降に解答を出力するものとし、出力形式は(画素値,画素数)、画素値は白画素を0, 黒画素を1とすること。括弧やカンマは全て半角とし、指定された以外の余分な空白や改行等はいれないこと。
    - 画像の上側から処理を行うこと。なお、BMP形式では画像の下側からデータが記録されているので、注意すること。
    - 20-1: 横方向に1列ごとに1次元データとして扱い、列の最後で改行すること。最後の列も同様とする。
    - 20-2: 画像の横方向に全ての列をつなげて1次元データとして扱うこと。改行はしないこと。
    - 提出するテキストファイル名はそれぞれ、ans20-1.txt, ans20-2.txt とすること。



# 追加課題

- 追加課題1 (No.2 の追加課題)
  - 入力画像の諧調数を指定された諧調数に削減するプログラムを作成せよ。
    - 指定する諧調数は2～254諧調とする
    - 指定した諧調数が入力画像よりも大きいときは処理しないこと
    - 新たな諧調数となるように画素値の区間を設定し、区間内の画素値を全てその区間の平均画素値となるようにすること
- 追加課題4 (追加課題1が必要)
  - モノクロ画像をディザ法で2値化する
    - 最初に画像の諧調数を16諧調に削減する
      - 諧調は0～15で表すこと
    - 画像を4×4画素単位の領域に分割する
      - 画像の左上から右方向、下方向に分割する
    - 各領域について、右図の値をしきい値として対応する画素を2値化する
      - 図の値以下なら黒画素(0)、超えていれば白画素(255)
    - 画像データの並びに注意すること
      - BMP形式では左下が(0,0)

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5