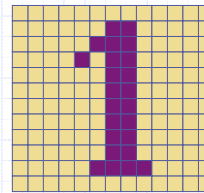


二次元画像処理

日浦慎作

パターンとシンボル

パターン



シンボル

- 0 青
- 1 赤
- 2 緑
- 3 黄

- 均質な要素の配列
- 各要素値の並びが重要

- 不均質均質な要素の配列
- 各要素が独立に意味を持つ

画像の処理と認識・理解

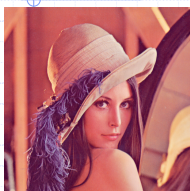
- ◆画像処理・画像変換
(パターン→パターン)
 - 画質改善
 - 画像符号化・圧縮
 - メディア変換 (不可視情報の可視化)
- ◆画像認識・画像理解
(パターン→シンボル)
 - 2次元パターン認識
 - 3次元画像計測・認識
- ◆画像生成 (シンボル→パターン)
 - コンピュータグラフィックス

狭義の画像処理

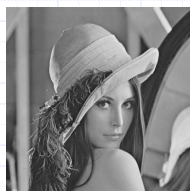
二次元画像処理

- ◆フィルタ演算
 - 画像をぼかす・鮮明にする
 - 画像中のエッジ(輪郭)を抽出する など
- ◆明度変換
 - コントラストの強調, 二値化 など
- ◆二値画像処理
 - ノイズ除去・ラベリング・細線化 など
- ◆画像圧縮
 - 可逆圧縮・非可逆圧縮

画像処理が対象とするデータ



カラー画像
24bit/pixel



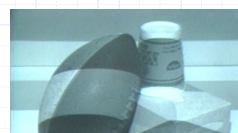
濃淡画像
8bit/pixel



二値画像
1bit/pixel

二値画像処理(教科書5章)

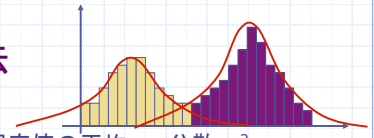
- ◆画像を「白」と「黒」だけで扱う処理
 - 図形の処理として, もっとも基本的
- ◆二値化とは
 - 画像を白と黒の領域に分ける処理
 - どのぐらいの明るさにするか? が問題



二値化と閾値決定

- ◆ 図と地の割合が予測できる場合 (文書等)
 - P-タイル法
ヒストグラムを一方から加算した割合がちょうど p になる値を利用
- ◆ ヒストグラムがはっきりとした双峰性
 - 2つのピーク間の最小値をしきい値とする
- ◆ その他の場合
 - 判別分析法 (次スライド)

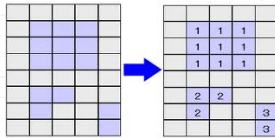
判別分析法



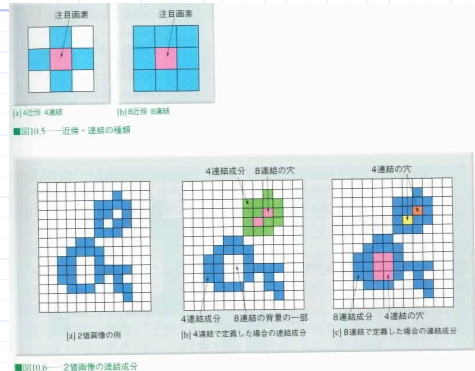
- ◆ 特徴量
 - 全画素の明度値の平均 μ , 分散 σ^2
 - 閾値以上・以下の分布をクラス1,2に分類
 - 各クラス x の割合 w_x , 平均 μ_x , 分散 σ_x^2
- クラス内分散 $\sigma_w^2 = w_1\sigma_1^2 + w_2\sigma_2^2$
- クラス間分散 $\sigma_B^2 = w_1(\mu_1 - \mu)^2 + w_2(\mu_2 - \mu)^2 = w_1w_2(\mu_1 - \mu_2)^2$
- $\sigma^2 = \sigma_w^2 + \sigma_B^2$
- ◆ 閾値の決定
 - クラス間の分離度 σ_B^2 / σ^2 を最大にする閾値

ラベリング

- つながった領域に「同じ値」を割り当てる処理
- 連続領域の数を調べる
 - それぞれの連続領域の面積や重心を求めるなどの処理が可能になる



4近傍と8近傍



ラベリング結果の違い



ラベリング

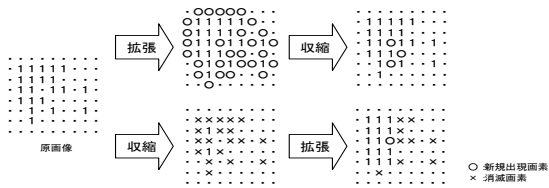


重複リスト

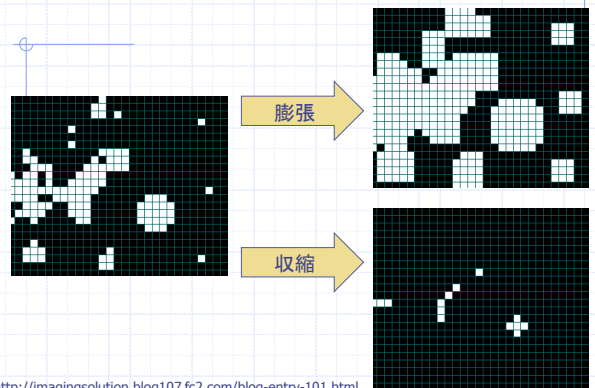


- [1パス]
上または左の画素と同じラベルを付与
- 左上の画素が異なるラベルを持つ場合
 - 重複リストに追加
 - 上も左も、0画素である場合
 - 新しいラベル番号を付与
- [2パス]
重複リストを元に、ラベルを更新

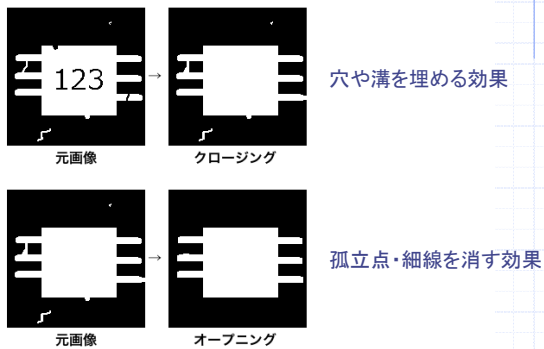
膨張・収縮



- ◆ 膨張・収縮 - 穴を埋める効果
- ◆ 収縮・膨張 - 孤立点を除去する効果



膨張・収縮処理は単独で行う事はまれで、膨張・収縮を繰り返し処理を行う場合が多くあります。とくに、同じ回数分だけ膨張して収縮する処理をクロージング (Closing)、同じ回数分だけ収縮して膨張する処理をオープニング (Opening) とよびます。



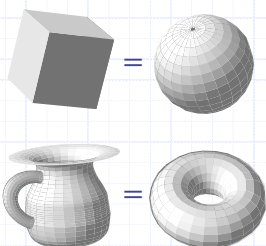
距離変換・スケルトン



- ◆ 距離変換 - 何度目の収縮処理で0画素になるか
- ◆ スケルトン - 距離変換画像の極大点 (近傍画素値が中央画素の値以下)
- ◆ 元の画像を復元可能

近傍演算による大局的情報：トポロジーの利用

- ◆ トポロジーとは
 - 変形しても変化しない図形の性質



二次元二値画像のトポロジー

- ◆ 連結成分の穴の数に対応

```

000000000000 000000000000
001110000000 001111000000
001111110000 001001100000
000011110000 000100100000
000110011000 000111110000
000000000000 000000000000
    
```

穴なし

穴一つ

オイラー数

◆(連結成分の個数) - (穴の個数)

```

000000000000 000000000000 000000000000
001110000000 001111100000 001111100000
001111110000 001100110000 001101110000
000011110000 000110010000 000111010000
000110011000 000111111000 000111111000
000000000000 000000000000 000000000000
    
```

オイラー数: 1 オイラー数: 0 オイラー数: -1

オイラー数の計算方法

◆4近傍の場合

```

000000000000 000000000000 000000000000
001111100000 001100100000 001111100000
001100110000 001100110000 001100110000
000110010000 000110010000 000110010000
000111110000 000111110000 000111110000
000000000000 000000000000 000000000000
    
```

1の個数 縦・横2接続 2x2接続
 ("1"の数:18) - (縦・横2接続の数:8+12) + (2x2接続の数:2) = 0

オイラー数の原理(1)

◆位相が不変な操作

```

000000 000000 000000 000000 000000
011100 011100 011100 011100 011110
001000 001110 001100 001100 001110
000000 000000 000000 001100 001100
    
```

画素数	+1	+1	+1	+1	+1
縦・横接続	+1	+1	+2	+3	+4
2x2接続	±0	±0	+1	+2	+3
オイラー数	±0	±0	±0	±0	±0

000000
 011110 注意: 左の0は穴ではない
 001010 図("1")が4近傍の場合、地("0")は8
 001100 近傍で考える

オイラー数の原理(2)

◆位相が変化する操作

```

000000 000000 000000 000000 000000
000000 111010 000110 011110 011110
001000 000010 001110 001110 001110
000000 000010 001100 001110 000100
    
```





画素数	+1	+1	+1	+1	+1
縦・横接続	+0	+2	+4	+4	+4
2x2接続	±0	±0	+2	+4	+2
オイラー数	+1	-1	-1	+1	-1
操作	出現	接続	接続	穴埋め	接続

8近傍のオイラー数

◆ちょっと複雑

画素数	縦・横 2接続	斜め接続	2x2領域中、3画素	2x2接続
V	E	D	T	F

オイラー数 = V - E - D + T - F

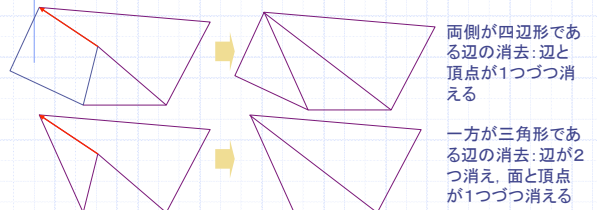
 4 - 4 - 2 + 4 - 1 = 1
  3 - 0 - 2 + 0 - 0 = 1
 8 - 8 - 4 + 4 - 0 = 0
  6 - 4 - 4 + 2 - 0 = 0

余談: オイラー数について



◆例: 多面体の性質

- (頂点数) - (辺数) + (面数) = 2 - 2 * (穴数)
- ◆ 立方体: 8 - 12 + 6 = 2 四角錐: 5 - 8 + 5 = 2

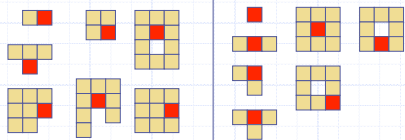


→ 最終的に4面体(オイラー数: 2)に帰着

消去可能性

◆消去可能画素とは

- 画像全体の連結性が変化しない画素
- 位相構造を変化させずに図形を変換



消去可能
消去不可能
4近傍の場合

消去可能性の計算

◆以下の式が1のとき消去可能

X_4	X_3	X_2
X_5	X_0	X_1
X_6	X_7	X_8

◆4近傍の場合

$$X_1 + X_3 + X_5 + X_7 - X_1 X_2 X_3 - X_3 X_4 X_5 - X_5 X_6 X_7 - X_7 X_8 X_1$$

◆8近傍の場合

$$X_1 + X_3 + X_5 + X_7 - X_1 X_2 X_3 - X_3 X_4 X_5 - X_5 X_6 X_7 - X_7 X_8 X_1$$

◆8近傍: 4近傍の図と地を入れ替えて計算

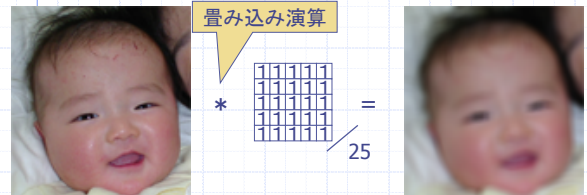
濃淡のある画像の処理

◆二値画像処理よりも高度

- 画像の明るさや色を調整する
- 画像をぼかしたり、鮮明にしたりする
- 画像から輪郭線を抽出する
- ... などなど

フィルタリング

◆畳み込み演算フィルタ

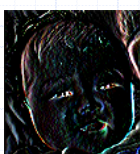


$$g(x, y) = \iint k(u, v) \cdot f(x-u, y-v) du dv = k * f$$

畳み込みフィルタの種類

平滑化

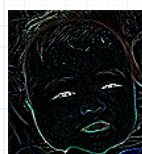
微分
(距離1)



(10倍に明るく)

ソーベル

ラプラシアン



(3倍に明るく)

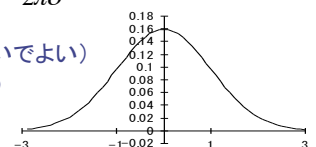
フィルタの数学的定義

◆ガウシアンオペレータ

- 平滑化オペレータ(数学的意味は後述)
- 畳み込みカーネル関数

$$k(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

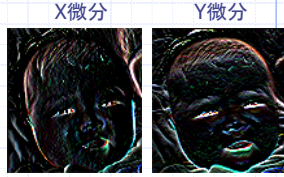
- 無限に続く関数(実際には3σぐらいでよい)
- σは、オペレータの広がり(平滑化の度合い)



微分フィルタ

◆2次元微分フィルタ

$$\nabla f(x, y) = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right]$$



- ベクトル値を持つフィルタ

◆エッジ強度

$$Df(x, y) = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

◆ラプラシアン

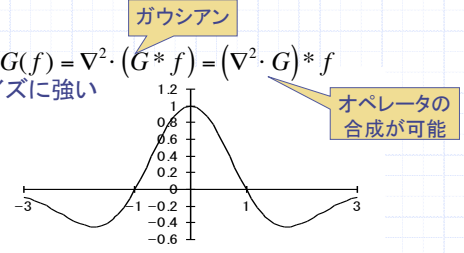
$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

LoG フィルタ

◆平滑化と微分フィルタを組み合わせたもの

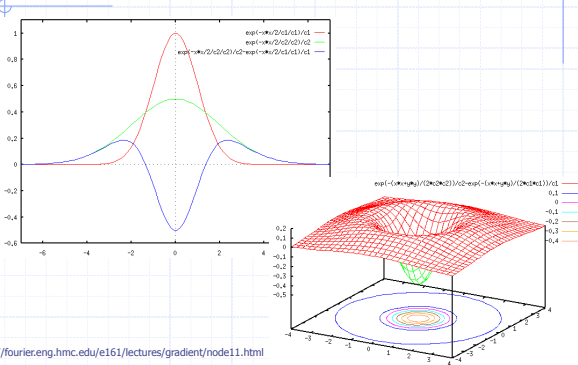
$$LoG(f) = \nabla^2 \cdot (G * f) = (\nabla^2 \cdot G) * f$$

- ノイズに強い

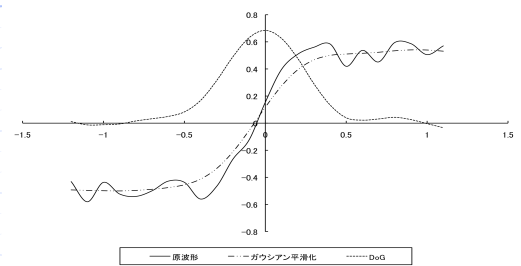


LoG オペレータ

DoG (Difference of Gaussian)



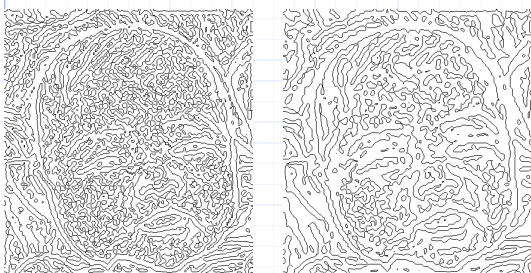
LoG フィルタの効果



- ◆ ノイズに対して強い微分が可能
(感応する空間周波数帯を選択可能)

LoG フィルタの例

LoG のゼロクロス抽出



エッジの位置が、若干移動することあり

二次元フーリエ変換

◆フーリエ変換の定義

- 連続 $F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot e^{-2j\pi(ux+vy)} dx dy$

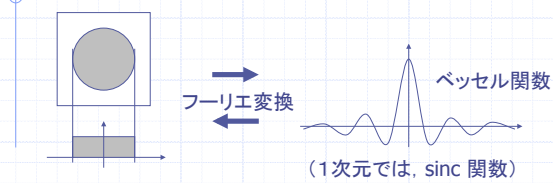
- 離散 $S(u, v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} s(m, n) \cdot e^{-2j\pi\left(\frac{um}{M} + \frac{vn}{N}\right)}$

◆性質

$$f * k = F^{-1} \{F\{f * k\}\} = F^{-1} \{F\{f\} \cdot F\{k\}\}$$

- 畳み込みは、積に移される
- 畳み込みオペレータは、空間周波数領域でのフィルタリングに置き換え可能(逆も真)

有限関数のフーリエ変換

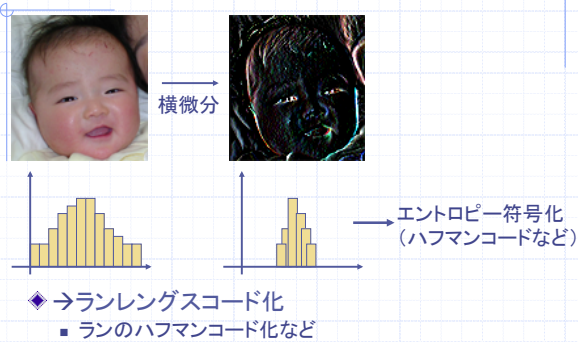


- ◆有限関数のフーリエ変換は, 有限ではない
 - ある周波数以上の信号をカットするための畳み込みオペレータの径は無限

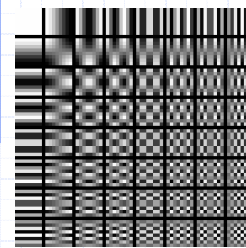
畳み込みOp. のフーリエ変換

- ◆ガウシアン
 - ガウス関数のフーリエ変換はガウス関数
→ ローパスフィルタ
- ◆微分・二次微分
 - ハイパスフィルタ
- ◆DoG, LoG
 - バンドパスフィルタ
- ◆総称して, 線形フィルタと呼ばれる

画像圧縮



直行変換符号化



離散コサイン変換の基底

- ◆K-L 変換
 - 理論上最高性能
 - 時間がかかる
 - 基底伝送の必要性
- ◆アダマール変換
 - 乗算が不要
- ◆離散コサイン変換
 - K-L 変換に近い性能

JPEG符号化

