

情報デザイン専攻

画像情報処理論及び演習I

-領域抽出-

ラベリング、細線化

第7回講義
水曜日1限
教室6218情報処理実習室

吉澤 信
shin@riken.jp, 非常勤講師
大妻女子大学 社会情報学部

独立行政法人
理化学研究所

Shin Yoshizawa: shin@riken.jp

今日の授業内容

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec07.pdf
www.riken.jp/brict/Yoshizawa/Lectures/Report02.doc

- ① 細線化・ラベリング
- ② 演習:
 - ✓ ラベリング・細線化のプログラムを動かしてみる.
 - ✓ 大津法プログラミング.

第2回のレポート(6/18×切)は今日の内容なので
頑張ってくださいーp(^)q

Shin Yoshizawa: shin@riken.jp

前回の復習: 閾値による二値化

閾値 ↓ ← → 閾値 ↓

✓ その画素値が閾値(threshold)より大 or 小で領域を二つに分ける.

0 255

閾値: 64 閾値: 96 閾値: 128 閾値: 160

Shin Yoshizawa: shin@riken.jp

前回の復習: 大津の二値化法

✓ 白の分布と黒の分布の「分離度」が大きくなるように閾値を自動的に決める.

✓ 分離度: $\frac{\text{クラス間分散}}{\text{クラス内分散}}$

頻度

黒の分布 黒画素クラス 画素数の ω_1

しきい値 t

白の分布 白画素クラス 画素数の ω_2

0 255 画素値

黒画素クラスの平均 m_1 全平均 m_t 白画素クラスの平均 m_2

©CC-ARTS 監修

Shin Yoshizawa: shin@riken.jp

細線化&ラベリング

✓ 二値化後の典型的処理として細線化とラベリングがある.

二値化 → ラベリング

細線化

ABCDE 12345

△ ○ □

©CC-ARTS 監修

Shin Yoshizawa: shin@riken.jp

前回の復習: ラベリングとは?

✓ ラベリング(Labeling): 連結領域を抽出する事.

✓ 連結領域: 同じ画素値の繋がった領域.

- 4連結: 左右上下.
- 8連結: 3x3の領域.

二値化 多値化 ラベリング

©S. Yoshizawa, RIKEN

Shin Yoshizawa: shin@riken.jp

4連結 VS 8連結

注目画素

4連結

8連結

4連結成分

8連結の穴

8連結成分

4連結の穴

[a] 2値画像の例

[b] 4連結で定義した場合の連結成分

[c] 8連結で定義した場合の連結成分

Shin Yoshizawa: shin@riken.jp

ラベリングのアルゴリズム(再帰)

再帰関数で書くと超簡単!

多値へも簡単に拡張可能.

```

for(y=0;y<sy;y++)
  for(x=0;x<sx;x++)
    if(bin[y][x] == BLACK){
      Recursive(x,y);
      out[y][x] = Label;
      label += L_BASE;
    }
  
```

bin[i][j]: 黒 or 白
out[i][j]: 出力のラベル
sx, sy: 画像サイズ

- main関数の中で黒なら再帰関数を呼び出す。
- 再帰が帰ってきたらラベルを変えて繰り返し。

- 再帰関数で8連結の周り呼び出しながら同じ値ならラベルを付けていく。
- 同時に黒→白。

```

int Recursive(int x, int y){
  if((out[y][x] != BLACK) || (x<0) || (y>=sy) || (y<0)) return 1;
  out[y][x] = Label;
  bin[y][x] = WHITE;
  return (Recursive(x, y+1)+Recursive(x, y-1)+
    Recursive(x+1, y+1)+Recursive(x-1, y+1)+
    Recursive(x+1, y)+Recursive(x-1, y));
}
  
```

Shin Yoshizawa: shin@riken.jp

重要: アルゴリズム(キュー or スタック)

残念ながら再帰関数は入れ子(階層的な呼び出し)の回数がOS毎に制限(高々10-20程度).

定理: 再帰アルゴリズムは繰り返しアルゴリズムに常に書き換える事が可能.

再帰の代わりにキューやスタック構造を使う.

再帰呼び出し

Stack

Queue

Push

Pop

再帰呼び出し

Stack

Queue

Push

Pop

Shin Yoshizawa: shin@riken.jp

ラベリングのアルゴリズム(キュー or スタック)2

初期Push

Popのループ

再帰のmainとほぼ同じ.

Put関数

8方向へPush.

Stack

Queue

Push

Pop

```

int doLabeling(Image *img){
  int i, j;
  queue<int> pqueue;
  mycheck[INDEX2D(sx, sy)] = 1;
  int currentLabel = 1;
  int i, j;
  Put(sx, sy, currentLabel, pqueue, mycheck);
  int popID = 1;
  while(pqueue.empty() == false){
    popID = pqueue.front();
    pqueue.pop();
    int i = (popID / sx);
    int j = (popID % sx);
    if(i < 0 || i >= sy || j < 0 || j >= sx) continue;
    Put(i, j, currentLabel, pqueue, mycheck);
    printf("Error ! impossible !! %d %d\n", i, j);
  }
  return 0;
}

void Put(int sy, int sx, Image *img, int currentLabel, std::queue<int> &pqueue, bool *mycheck){
  int i, j;
  for(i=sy-1; i<=sy+1; i++){
    for(j=sx-1; j<=sx+1; j++){
      if(i==sx || j==sy) continue;
      if(mycheck[INDEX2D(i, j)] || !bin[i][j] || i[j] != currentLabel) continue;
      mycheck[INDEX2D(i, j)] = 1;
      pqueue.push(INDEX2D(i, j));
    }
  }
}
  
```

Shin Yoshizawa: shin@riken.jp

細線化(thinning, 骨格化:skeletonization)

領域抽出後(二値化)に領域を線状に簡略化する事、ただし通常は入力(二値画像)と同位相の形状.

出来るだけ中心に細く、端点でない境界画素を削除していく.

分岐点

端点

通過点

細線化

文字認識等で非常によく用いられる!

Shin Yoshizawa: shin@riken.jp

細線化その2

同位相: 連続変形で変換可能である事:

- 球、平面、トーラス等はそれぞれ異なる位相.
- 穴(境界)の数、ハンドル(トーラス)の数等で分類.

↑のコップとトーラスは同位相

Shin Yoshizawa: shin@riken.jp

連結数

✓ **連結数**: 境界線追跡をしたとき、その画素を通過する回数: **消去で連結数が変わらない=同位相.**

$N_4: 0 \quad 1 \quad 2 \quad 3 \quad 4$

$$N_4 = \sum_{k \in C} (f(x_k) - f(x_{k_1}))f(x_{k_2})f(x_{k_3})$$

| 連結数 | 画素の特徴 |
|-----|-----------|
| 0 | 孤立点または内部点 |
| 1 | 端点または境界点 |
| 2 | 連結点 |
| 3 | 分岐点 |
| 4 | 交差点 |

$$C = \{1, 3, 5, 7\}, \quad f = \begin{cases} 1 & \text{注目画素} \\ 0 & \text{注目画素} \end{cases}$$

$$\bar{f} = 1 - f, \quad x_0 = x_1.$$

$$N_8 = \sum_{k \in C} (\bar{f}(x_k) - \bar{f}(x_{k_1}))\bar{f}(x_{k_2})\bar{f}(x_{k_3}) \quad \text{4連結 8連結}$$

©CC-ARTS 協会

Shin Yoshizawa: shin@riken.jp

細線化その3

✓ **中心軸(Medial Axis)**の近似である事が多い。
 ✓ 細線化後は線分の幾何特徴(長さや円形度等)を計算。
 ✓ 様々な方法: **境界・連結数を変えない・端点を消去.**

- テンプレートを用いた繰り返し法:
- Stentford法、**Hilditch**法(連結数を使う、少し複雑なのでskip、演習のthinning.hに実装)、田村法、Zhang-Suen法。
- 中心軸を用いる方法、etc.

©C. Lee et al. PC'98

Shin Yoshizawa: shin@riken.jp

中心軸(Medial Axis)

✓ **定義**: 接触円の中心の軌跡。
 ✓ **接触円**: 二点以上で境界に接している境界内の円。

境界、接触円、境界との接点 x 、中心軸、接触円群、中心軸、境界

x での厚み

✓ H. Blum, 1967.

©www.cim.mcgill.ca/~frigo, ©math.berkeley.edu/~sethian, ©www.math.ucsa.edu

Shin Yoshizawa: shin@riken.jp

中心軸と距離場

✓ 中心軸は距離場の等高線が特異点となる点の集合。

特異点: 滑らかでない点、微分出来ない点、勾配が零。

Shin Yoshizawa: shin@riken.jp

ポロノイ図(Voronoi Diagram)

✓ 2点間を結ぶ線分の垂直二等分線の一般化。

©www.ghull.org

Shin Yoshizawa: shin@riken.jp

ポロノイ図と中心軸

✓ 中心軸はポロノイ図の滑らかな曲線への一般化である。

一般化Voronoi図の部分集合

Shin Yoshizawa: shin@riken.jp

3D中心軸

✓多次元の中心軸もあり、CGやCAD等で応用されている。

3Dの中心軸は面、孤立点と線の集合

Shin Yoshizawa: shin@riken.jp

中心軸の応用

✓応用: 認識, 接触判定, 曲面再構成, Meshing, 変形, ...

N. Amato et al. SIGGRAPH'98
S. Zhu and A. Yalle, ICV, 2003, 1996.
M.-C. Chang and B. Kana, CVPR'08
B. Levy and Y. Liu, SIGGRAPH'10
G. Brachmann and C. O'Sullivan, ACM SCA'02
S. Yoshizawa et al., EG'07

Shin Yoshizawa: shin@riken.jp

細線化の応用例: ベクトル化

分岐点 端点 通過点

©CG ARTS 2008
©Mitsuhiko Okabe et al. SIGGRAPH'05
©J. Seo et al. SIGGRAPH 2007

Shin Yoshizawa: shin@riken.jp

演習: ラベリング & 細線化をしてみよう!

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Ex04.zip

前回の続き+ラベリング、細線化:

1. 演習7: Ex04内に用意されたプログラム群を動かして、ラベリングと細線化を実行。ラベリングの閾値を変えて実行してみましょう。
2. (前回の続き)クラス間分散を計算して大津法のプログラムを作成(Lec06.pdfの演習6)。
3. レポート2(6月18日〆切)を解く。

Shin Yoshizawa: shin@riken.jp

演習プログラムの説明1

Ex04.zipをダウンロード→解凍。
コンパイルは「make」、詳細はMakefileを見てください。

- ✓ LabelingRemoveSmall.cxx:(引数の数3) pgmを大津法+ラベリング(8連結)+第三引数以下の領域サイズを一つにまとめる(小さい面積の領域を統合)+ラベル毎に疑似カラーでppmで保存。
 - 実行方法: ./LabelingRemoveSmall 入力pgm 出力ppm 削除する領域の面積閾値(int)
- ✓ Thinning.cxx(引数の数2): 大津法+Hilditchの細線化。
 - 実行方法: ./Thinning 入力pgm 出力pgm
- ✓ ヘッダーファイル: otsu.h: 大津法、label.h: ラベリング、color.h: 疑似カラー、thinning.h: 細線化。
 - 実装の細部に興味がある人は見てください。

Shin Yoshizawa: shin@riken.jp

演習プログラムの説明2

ラベリングのIDを1つつ付けた場合に、各ラベルを輝度値としてグレースケール画像に出力すると、連結領域が分かりにくい→疑似カラーで表示。

グレースケール: 0,1,...,255 疑似カラーマップ: color.h

領域数3

入力 大津法二値化 ラベリング(ID=輝度値) ラベリング疑似カラー

Shin Yoshizawa: shin@riken.jp

演習プログラムの説明3

疑似カラーでも領域数が多いと分かりにくい→領域の面積(画素数)が閾値より小さい領域は全て同じラベル(黒)+ラベルの振りなおし.

領域数627 領域数46 領域数31 領域数26

閾値:0 閾値:30 閾値:60 閾値:120

領域数627

入力 大津法二値化 ラベリング(ID=輝度値) ラベリング疑似カラー

Shin Yoshizawa: shin@riken.jp

演習プログラムの実行例

領域抽出+ラベリングを行うと、表示の綺麗さだけでなく定量的な解析が可能になる(数、面積、境界形状の長さや曲率など).

領域数220 領域数19 領域数14 領域数9

閾値:0 閾値:30 閾値:60 閾値:120

領域数220

入力 大津法二値化 ラベリング(ID=輝度値) ラベリング疑似カラー

Shin Yoshizawa: shin@riken.jp

演習プログラムの実行例2

細線化は幅が1画素の線になる. 線あり:黒・なし:白の表示.

入力 大津法二値化 細線化

入力 大津法二値化 細線化

ト
ト

Shin Yoshizawa: shin@riken.jp

演習プログラムの実行例3

後期でやるフィルタ処理・エッジ強調と組み合わせると、より有用.

大津法二値化 細線化 エッジ(勾配)強度 細線化

スタイル化エッジ 細線化 スタイル化エッジ 細線化

Shin Yoshizawa: shin@riken.jp

演習プログラムの実行例4

大妻女子大学 社会情報学部
School of Social Information Studies, Otsuma Women's University

大妻女子大学 社会情報学部
School of Social Information Studies, Otsuma Women's University

大妻女子大学 社会情報学部
School of Social Information Studies, Otsuma Women's University

大妻女子大学 社会情報学部
School of Social Information Studies, Otsuma Women's University

大妻女子大学 社会情報学部
School of Social Information Studies, Otsuma Women's University

Shin Yoshizawa: shin@riken.jp

次回の予定

✓ 目標: 大津法がプログラミング出来る様になる!

内容(7-9): 領域抽出
特に大津法・ラベリング

| | | | |
|-----|-----------|--|----|
| 1回 | 基礎 | | |
| 2回 | | | |
| 3回 | | | |
| 4回 | | | |
| 5回 | アフィン変換・補間 | | |
| 6回 | | | |
| 7回 | | | |
| 8回 | | | |
| 9回 | 領域抽出 | | 入力 |
| 10回 | | | |
| 11回 | | | |
| 12回 | | | |
| 13回 | 画像合成 | | |
| 14回 | | | |
| 15回 | | | |

二値化 多値化 ラベリング