

情報デザイン専攻

画像情報処理論及び演習I

-領域抽出-
大津の二値化法、ラベリング、細線化

第8回講義
水曜日1限
教室

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/briect/Yoshizawa/Lectures/index.html
www.riken.jp/briect/Yoshizawa/Lectures/Lec05.pdf

- ① 細線化・ラベリング
- ② 演習:
 - ✓ 二値化のプログラム作成
 - ✓ ラベリング・細線化のプログラムを動かしてみる。

今回のレポートは今日の内容なので頑張ってくださいねー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 しきい値 t 白の分布

黒画素クラス 画素数 w_1 白画素クラス 画素数 w_2

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

0 255 画素値

CCG-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

ABCDEF ABCDEF
12345 12345

△ ○ □ △ ○ □

CCG-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

CCG-ARTS 監修

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連結成分

4連結の穴

8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);
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連結の周りを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>=sx)||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+1))+Recursive(x-1, (y-1)));
}

```

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.

$f(f(f(\dots)))$

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

演習のlabel.h

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.

```

int out_label[image][image]{0};
int Label[image][image];
int seq, int seq_max, int d_label, bool *mycheck;
int i, j;
size_t n;
int in=0;
size_t n;
bool *mycheck;
for(i=0;i<image;i++)
for(j=0;j<image;j++)
mycheck[i][j]=0;
int seq=0;
Label[image][image]=0;
for(i=0;i<image;i++)
for(j=0;j<image;j++)
if(mycheck[i][j]==0){
Label[i][j]=seq;
LabelInt++;
}
return 0;
}

```

```

void Put(int seq, int sx, int sy, int currentLabel, std::queue<int*> *pqueue, bool *mycheck){
for(i=sx-1;i<=sx+1;i++)
for(j=sy-1;j<=sy+1;j++){
if(i==sx&&j==sy)
continue;
if(mycheck[i][j]==0){
mycheck[i][j]=currentLabel;
*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、ただし通常は入力の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.

異なる位相

↑のコップとトーラスは同位相

Shin Yoshizawa: shin@nriken.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**

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

$$N_4 = \sum_{k \in C} (f(x_k) - f(x_k) f(x_{k+1}) f(x_{k+2}))$$

$$N_8 = \sum_{k \in C} (\bar{f}(x_k) - \bar{f}(x_k) \bar{f}(x_{k+1}) \bar{f}(x_{k+2}))$$

$C = \{1, 3, 5, 7\}, f = \begin{cases} 1 \\ 0 \end{cases}$
 $\bar{f} = 1 - f, x_0 = x_1$

連結数	画像の特徴
0	孤立画素または閉閉点
1	端点または境界点
2	連結点
3	分岐点
4	交差点

注目画像: 注目画像

Shin Yoshizawa: shin@nriken.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。
 ✓ テンプレートをを用いた繰り返し法:
 Stentiford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.

注目画像

Shin Yoshizawa: shin@nriken.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 、 x での厚み、中心軸、接触円群、中心軸、境界

✓ H. Blum, 1967.

Shin Yoshizawa: shin@nriken.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: 滑らかなでない点、微分出来ない点、勾配が零。

Shin Yoshizawa: shin@nriken.jp

ポロイ図(Voronoi Diagram)

✓ 2点間を結ぶ線分の垂直二等分線の一般化。

Shin Yoshizawa: shin@nriken.jp

ポロイ図と中心軸

✓ 中心軸はポロ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

The image shows two 3D models. On the left, a hand is shown with a green mesh and a blue center axis. On the right, a horse is shown with a green mesh and a blue center axis. The center axis is represented as a collection of surfaces, isolated points, and lines.

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, ...

The image shows several applications of center axes. It includes a hand mesh, a hand with a center axis, a hand with a center axis and a mesh, and a hand with a center axis and a mesh. It also shows a hand with a center axis and a mesh, and a hand with a center axis and a mesh. The image includes references to research papers: T. Amato et al., SIGGRAPH '98; B. Levy and V. L. Lee, SIGGRAPH '01; S. Zhu and A. Yeh, ICV '03, 1996; M. C. Chang and B. Kohn, CVPR '06; G. Brostow and C. O'Sullivan, ACM SIGACT; 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

The image shows the vectorization of a character and various objects. It includes a character with a vectorized outline, a character with a vectorized outline, and a character with a vectorized outline. It also shows a character with a vectorized outline, a character with a vectorized outline, and a character with a vectorized outline. The image includes references to research papers: E. T. Sui et al., SIGGRAPH '97.

Shin Yoshizawa: shin@riken.jp

レポートについて1

演習の時間に採点結果を返します！

- ✓ 採点に納得がいけない人は講義終了後に交渉可。
- ✓ レポートは7末(予定)まで受け付けますが、 \times 切後は点数に0.8倍。
- ✓ 名前と学籍番号かいてくださいねー(^-^;
- ✓ 良く書いて(調べて、考えて)いたり、coolな解答にはボーナス点を加算しました。

Shin Yoshizawa: shin@riken.jp

レポートについて2

演習の時間に採点結果を返します！

- ✓ レポートは何度出してもOK: 解けなかった問題が分かったりプログラムが書けたら、その部分だけ追加で提出OK。ただし(何回目の提出を)明記する事 & \times 切すぎたら0.8倍はかわらず。
- 今回(Q9,Q10出来なくて)点数低かった人もQ9&Q10やれば50*0.8点を加算しますよー。
- ✓ 誰もやってくれる人がいなかったんで... $\cdot\cdot\cdot$ (/Д`) $\cdot\cdot\cdot$ 、第一回レポートExtra問題の点数を100点に引き上げます!、プログラム得意な人は挑戦してね(^v^)! Extraだけ \times 切過ぎても0.8倍はなし!

Shin Yoshizawa: shin@riken.jp

演習: 二値化のプログラムを作ってみよう!

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

前回の続き+ラベリング、細線化:

1. Ex04内に用意されたプログラム群を動かしてみる。
2. 大津法の代わりに閾値をマニュアルで与えて二値化した結果を用いてRun_ex04_1.sh(又はRun_ex04_2.sh)と同様の結果を出力するプログラムの作成。
3. (前回の続き)クラス間分散を計算して大津の方法をプログラムしてみる([otsu_Chi.h](#)を完成させる)。

Shin Yoshizawa: shin@riken.jp

演習:プログラムの説明1

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

Ex04.zipをダウンロード→解凍.
コンパイルはMakefileを見てください.

- ✓ ex04_0.cxx:(引数あり) pgmを引数を閾値として二値化.
- ✓ ex04_1.cxx: pgmを大津法で二値化.
- ✓ ex04_2.cxx: pgmを大津法+ラベリング.
- ✓ ex04_3.cxx: pgmを大津法+ラベリング+ラベル毎に疑似カラーでppmで保存.
- ✓ ex04_4.cxx:(引数intあり) pgmを大津法+ラベリング+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.
- ✓ ex04_5.cxx: 大津法+Hilditchの細線化.

Shin Yoshizawa: shin@riken.jp

演習:プログラムの説明2

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

ヘッダーファイル:

- ✓ otsu.h: 大津法、使い方はex04_1.cxx.
- ✓ [otsu_chi.h: 大津法雛形](#).
- ✓ label.h: ラベリング、使い方はex04_2.cxx
- ✓ color.h: 疑似カラー、使い方はex04_3.cxx
- ✓ thinning.h: 細線化、使い方はex04_5.cxx
- ✓ ppmio.h: ppm画像入出力、Ex01/ex01_2.cxx
- ✓ pgm.h: pgm画像入出力、Ex01/ex01.cxx
- ✓ SimpleImage.h: 画像クラス、Ex01/ex01.cxx

Shin Yoshizawa: shin@riken.jp

演習:プログラムの説明3

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

シェルスクリプト: ex04_0.cxx~ex04_5.cxxまでコンパイル(make)+実行(lena.pgm入力)+出力ファイルをdisplayで表示.

- ✓ Run_ex04_0.sh: ex04_0.cxx
「sh Run_ex04_0.sh 引数(int)」で実行、lena.pgm
- ✓ Run_ex04_1.sh: ex04_1.cxx~ex04_5.cxx
「sh Run_ex04_1.sh」で実行、lena.pgm
- ✓ Run_ex04_2.sh: ex04_1.cxx~ex04_5.cxxまでを入力ファイル名とex04_4の引数を与えて実行.
「sh Run_ex04_2.sh 入力ファイル名 引数(int)」

Shin Yoshizawa: shin@riken.jp

演習: 1-1

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

Ex04内に用意されたプログラム群を動かしてみる.

1. Run_ex04_0.shを引数32,64,96,128,192と変えて実行してみてください.

Shin Yoshizawa: shin@riken.jp

演習: 1-2

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

Ex04内に用意されたプログラム群を動かしてみる.

2. Run_ex04_1.shを実行してみてください.

Shin Yoshizawa: shin@riken.jp

演習: 1-3

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

Ex04内に用意されたプログラム群を動かしてみる.

3. Run_ex04_2.shを使ってCameraman.pgmとKanji_Iri.pgm及びOtumaWeb.pgmを引数100で実行してみてください.

Shin Yoshizawa: shin@riken.jp

演習: 1-3

Shin Yoshizawa: shin@riken.jp

演習: 1-3

Shin Yoshizawa: shin@riken.jp

演習: 2 (次のレポートに出します!)

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

大津法の代わりに閾値をマニュアルで与えて二値化した結果を用いてRun_ex04_1.sh (又はRun_ex04_2.sh) と同様の結果を出力するプログラムの作成。

- ✓ ヒント: ex04_2.cxx, ex04_3.cxx, ex04_4.cxx, ex04_5.cxxの閾値(threshold)を大津法で計算している部分をコメントアウトしてex04_0.cxxと同じ様にargv[]を使って引数で渡すだけ。

Shin Yoshizawa: shin@riken.jp

演習: 3 (次のレポートに出します!)

www.riken.jp/briect/Yoshizawa/Lectures/Ex04.zip

(前回の続き)クラス間分散を計算して大津の方法をプログラムしてみる(otsu_chi.hを完成させる)。

- ✓ 講義スライドLec04.pdfのアルゴリズム(page 5)を使って自分で大津の方法をプログラムしてみてください。
- ✓ ヒント: otsu_chi.hを書き換えてex04_1.cxxでincludeする。Otsu_chi.h中のdouble getSmax(int thr, int hsize, int *hist[])内でクラス間分散の分子を計算。thr閾値、hsizeヒストグラムのサイズ、*hist、ヒストグラム配列(hist[0]...hist[hsize-1])。
- ✓ 戻り値でクラス間分散の分子を返す。
- ✓ 注意点: 画素数の掛け算は大きな値なので、閾値thrで分けられた二つのクラスの画素数はlong longを使う事。
- ✓ また、途中計算の変数はdoubleを使う事。

Shin Yoshizawa: shin@riken.jp

来週の予定

www.riken.jp/briect/Yoshizawa/Lectures/index.html

©Sapiro and Ballester, SIGGRAPH 2000. ©Perez et al. SIGGRAPH 2003.

- ① 第三回レポート(二値化、ラベリング、細線化)。
- ② 画像合成・Inpainting
- ③ 演習。