

第6回CADセミナー

主催：日本放射線技術学会 画像分科会

共催：学術委員会，中部部会

日時：2002年2月2日 午後1時より5時まで

場所：岐阜大学工学部応用情報学科 / 会議室および演習室

今回の演習は，前回の岐阜大学でのCADセミナー（第3回）の続きとして企画しました．

今回が最初の参加の方は必要に応じて前回の内容から学習してください．

また，ホームページと合わせて学習されることをお勧めします．

第3回：<http://www.fjt.info.gifu-u.ac.jp/cad-sem3>

第6回：<http://www.fjt.info.gifu-u.ac.jp/cad-sem6>

エディタ（mule）の使い方：<http://toba.yanagi.gifu-u.ac.jp/freshers/2001/D/index.html>

<プログラム>

12:20 受付開始（応用情報学科1階会議室C127）

1:00 - 1:05 挨拶，概要説明（応用情報学科1階会議室C127）

岐阜大学工学部：藤田広志（学術委員長），原 武史（分科会委員）

1:05 - 1:45 教育講演（応用情報学科1階会議室C127）

金沢大学医学部：真田 茂（画像分科会委員）

内容：画像処理プログラミング・特徴抽出などについての教育講演

2:00 - 4:30 演習（応用情報学科7階会議室C728, 730）

4:30 - 5:00 研究室，学内見学

工学部応用情報学科・藤田研究室の見学，他

その後，岐阜市内において懇親会を行います．

岐阜大学バス停から17:30に会場行きのマイクロバスが出ます．

<セミナースタッフ>

画像分科会：真田 茂（金沢大学），原 武史（岐阜大学），

学術委員会：藤田広志（岐阜大学），松原友子（名古屋文理大学）

補助スタッフ：福岡大輔（岐阜工業高等専門学校），畑中裕司（岐大D3），中川俊明（岐大D3）

篠原範充（岐大D2），牛場洋明（岐大D2）

この資料は，第6回CADセミナーにおいて配付したものです（2/1/2002作成）．

また，セミナーで用いた画像の一部は日本放射線技術学会発刊の標準デジタル画像データベース

（胸部腫瘍陰影像および非腫瘍陰影像）を利用しています．

演習 6 : 画像の 2 値化

この演習はprogramディレクトリのno.6, no.6-2というディレクトリの中で行います。

この演習では、画像の 2 値化処理の手順を学びます。2 値化とは、画像を白と黒の 2 つの領域に分ける処理です。そのためには、どの画素値から上を白（もしくは黒）でしたから黒（もしくは白）にするのかという、「しきい値」を決める必要があります。

そのために、まず最初に手動でしきい値を決めるようなプログラムを作成します。次には、画像全体の平均値をしきい値に設定するプログラムを作成します。

そして、それらの演習をもとに、任意の値を入力して、画像の平均値からその入力した値を引いた値をしきい値とするようにプログラムを変更します。

< 実験6-1 : コンパイル >

プログラムをコンパイルしてみましょう。
実行ファイルがprocという名前で作成されます。

< 実験6-2 : 実行 >

実行してみましょう。ここでは、コマンドラインから「入力画像のファイル名」、「画像の幅」、「高さ」、「出力画像のファイル名」を指定します。

入力画像名 : chest.raw

幅, 高さ : それぞれ512

出力画像名 : 好きな名前

実行すると、

Input threshold =

と尋ねられるので、しきい値を入力します。0 ~ 255の間になります。その後、out.rawという名前で 2 値化処理後のファイルが作成されます。

< 実験6-3 : 画像表示 >

表示用のプログラムがあるので、画像のファイル名、画素の幅、高さを入力します。

< 実験6-4 : ソースファイルを見る >

muleでソースファイルを読んでください。コメントなどを確認すると、動作の流れが分かります。

(no.6ディレクトリに移動した後)

```
%cd no.6で移動できます。
```

```
% gcc -o proc main.c
```

注意 : -oはハイフンとオー (小文字)

```
% proc chest.raw 512 512 out.raw
```

```
% x8view ファイル名 幅 高さ
```

```
% x8view chest.raw 512 512 : 原画像表示
```

```
% x8view out.raw 512 512 : 処理画像表示
```

```
% mule main.c
```

<実験6-5：さまざまなしきい値で>

しきい値を変えて実験をしてみましょう。

<実験6-6：2値化処理のプログラム>

2値化処理のプログラムでは，if文を使ってしきい値よりも大きい画素値は255，小さい画素値は0に変換しています。

プログラム中のコメントをよく読んで下さい。関数名はbinarize_imageです。

<実験6-7：キーボードからの入力>

プログラム中，scanfという関数でキーボードから文字を入力し，その値を変数に代入しています。その方法を確認して下さい。

<実験6-8：画像の平均値をしきい値に>

次はしきい値を手動ではなく自動で設定する方法を考えます。今回は単純に画像の平均値をしきい値にします。平均値は，関数：get_averageに画像を渡すと帰ってきます。その戻り値がしきい値になるように2値化の関数：binarize_imageの引数として渡します。

コンパイルは前と同じです。

そして，実行して表示を行ってみましょう。

平均値も表示されていますか？確認して下さい。

<実験6-9：画像の平均値をしきい値に>

2つのプログラムを元にして，任意の値を入力して，画像の平均値からその入力した値を引いた値をしきい値とするようにプログラムを変更します。平均値を手動で変更するという事です。

そして，表示を行ってみましょう。

```
%mule main.c
```

でファイルを開き，コメント類を読んで下さい。特にbinarize_image(..)あたりを中心に読んで下さい。

scanfには変数の名前ではなく，&をつけて渡しているところに注意して下さい。

次にno.6-2というディレクトリで作業します。

```
% cd ../no.6-2
```

で移動できます。

コンパイル：

```
% gcc -o proc main.c
```

実行：

```
% proc chest.raw 512 512 out.raw
```

結果表示：

```
% x8view out.raw 512 512
```

がんばりましょう。

<ヒント>

no.6-2のmain.cを元に改造します。

scanfの使い方はno.6の内容を元にします。

平均値とscanfで入れる変数との差がbinarize_imageのしきい値の引数に渡るようにします。

演習 7 : 2 値画像の膨張・収縮処理

この演習はprogramディレクトリのno.7, no.7-2, no.7-3, no.7-4のディレクトリの中で行います。2 値画像処理の第 2 段階は、膨張・収縮処理を取り扱います。膨張収縮処理とは、2 値画像の白い領域もしくは黒い領域を広げたり、小さくしたりする処理です。小領域の削除などに非常に効果のある処理方法です。

もっとも簡単なアルゴリズムは、

・注目画素の 8 近傍のうち、1 つでも白（黒）があったら、注目画素を白（黒）にするというものです。この処理は、一般的には、モルフォロジ処理と呼ばれます。

この演習では、まず膨張収縮処理を構築します。膨張処理はDilation（ダイレーション）、収縮処理はErosion（エロージョン）と呼ばれます。

そして、この 2 つの処理を組み合わせた演算に、Opening, Closingという処理があります。

Openingは、erosion処理後にdilation処理を行う処理であり、Closingは、dilation処理後にerosionを行う処理です。

それらの構築についても演習を行います。を行うプログラムを作成し処理結果の 8 ビット画像を得ることを目的とします。

< 実験7-1 : 白い領域の膨張処理 >

入力画像の平均値で 2 値化を行い、膨張処理を行います。プログラムをコンパイルして、実行してみましょう。そして、その結果を表示してみましょう。

< 実験7-2 : ソースファイルの観察, 改造 >

muleを使ってファイルの中身を見ましょう。膨張処理を行っている関数はdilationです。どのように値を渡しているのか確かめましょう。if文がいくつかつながっており、一見ややこしいようですが、周りに 1 つでもmax_valueが入っていると中心部もmax_valueにする、という操作の繰り返しです。

また、同じ膨張処理を何度も繰り返すように改造してみましょう。

< 実験7-3 : 白い領域の収縮処理 >

実験7-1, 7-2と同じように作業してみましょう。

(no.7ディレクトリに移動した後)

コンパイル

```
% gcc -o proc main.c
```

実行

```
% proc chest.raw 512 512 out.raw
```

表示

```
% x8view out.raw 512 512
```

```
% mule main.c
```

引数の渡し方に注意が必要です。

配列を入れ替えるような新しい関数を作成するのも 1 つの方法です。

(no.7-2ディレクトリに移動してください)

< 実験7-4：白い領域のOpening処理 >

Opening処理は，収縮処理の後に膨張処理を行う処理です．一見，元に戻るようですが，元には戻りません．

前の2つの実験で収縮処理と膨張処理の関数を作成しましたので，その2つを使って実現します．サンプルはno.7-3のディレクトリにあります．ここでは，収縮処理はerosion.c，膨張処理はdilation.cというファイルに関数が書かれています．

全体の流れを決めているのはmain.cです．

コンパイルして，実行して，表示してみましょう．

< 実験7-5：Opening処理の詳細 >

Opening処理は，main文の中で

```
erosion(outimg,inimg,width,height);  
dilation(inimg,outimg,width,height);
```

といった2行で表現されています．画像の配列の渡し方に注意して下さい．最初のerosionでは，処理前がoutimg，処理後がinimgになります．そのerosion処理後の画像をdilationに渡さないといけませんので，dilationの入力がinimgになっています．出力は，多分もう使わないoutimgに入れるようにしています．

このことを理解したうえで，Opening処理を複数回実行するようにプログラムを改造しましょう．そして，結果ファイルの名前を変えて保存するようにして回数の違いによる画像の変化を観察しましょう．

< 実験7-6：白い領域のClosing処理 >

Opening処理とその改造が理解できれば，Closing処理は簡単です．erosion, dilationの順序を変えるだけです．no.7-3のmain文を参考にClosing処理に改造してみましょう．改造した結果の例はno.7-4のディレクトリに保存します．no.7-4ではMakefileを用意しているので，コンパイルは簡単です．

(no.7-3ディレクトリに移動してください)

コンパイル：

```
%gcc -o proc main.c dilation.c erosion.c
```

実行：

```
% proc chest.raw 512 512 out.raw
```

表示：

```
% x8view out.raw 512 512
```

引き続きno.7-3ディレクトリで行います．

コンパイル方法は上と同じです．

改造するのはmain.cだけで大丈夫です．

(no.7-4ディレクトリで行います)

Makefileでのコンパイル

```
% make
```

と入力するだけでコンパイルができます．このMakefileでは，実行ファイルはprocになります．

演習 8 : ラベリング処理

この演習はprogramディレクトリのno.8のディレクトリの中で行います。

この演習では2値画像のラベリング処理を取り扱います。ラベリング処理は、CAD技術の中でもっとも重要で基本的な処理です。

これは、抽出した領域の特徴を調べたり、個数を数えたりすることの基本になるからです。ラベリング処理とは、つながっている領域に番号を付けていく処理です。

つまり、画像中に孤立した領域が何個あるのかを数えたり、ある番号のついた領域がどんな形をしているのかを調べるのに有効な処理です。

ラベリングの手法は非常に多くありますが、今回は比較的単純な方法でラベルづけを行います。基本方針は以下のようです。ちょっとややこしいですが、頑張ってください。

1. 画像を走査してラベルがついていない画素を見つけ、ついていなければ新しい番号(ラベル)を付ける
2. 見つかった画像の連結性を調べて、つながっているが画素には同じ番号(ラベル)をつける。
3. 新しくつけたラベルの連結性もしらべて、連結があれば、同じ番号をつける
4. この操作を、ラベルがつけられなくなるまで続ける
5. そうしたらまた最初に帰り、ラベルのついていない画素を見つけ、そこに新しいラベルをつけ繰り返す。
6. 画像全部が終われば終了

また、ラベリング処理の前処理も重要です。ここでは、dilation, erosionとClosing処理を行った後にラベリングすることとしています。演習では、12個の濃度の違う円が書かれた画像(12circles.raw)と胸部画像などを題材に実験を行います。

<実験8-1:ともかく実行してみましよう>

画像中にある円の個数を数えるプログラムを作成しましょう。まず最初に原画像を表示しましょう。そして、サンプルプログラムをコンパイル、実行し、その結果を見てみましょう。実行結果の画像out.rawには、画素値のかわりにラベル番号(1番からつきます)が保存されています。

実行後の出力に、

The number of labels = 12

と表示されましたか?これで画像中に2値化で白く残った領域が12個ありますということが示されました。

なお、ラベル画像を見るためには、xlabelviewという表示プログラムを使って下さい。ラベル画像、幅、高さ、ラベル数を渡します。ラベルの番号によって違う濃度で表示します。

<実験8-2:他の画像でも実験(1)>

胸部画像:chest.rawと、12circles.rawをほか

(no.8ディレクトリに移動します)

原画像の表示:

```
% x8view 12circles.raw 512 512
```

コンパイル:

```
% make
```

実行:

```
% proc 12circles.raw 512 512 out.raw
```

ラベル画像の表示例:

```
% xlabelview out.raw 512 512 12
```

実行:

```
% proc 12circles2.raw 512 512 out.raw
```

した画像: 12circles2.rawでも実験してみましよう。ぼけた12個の円の画像の出力は12になりましたか? しきい値を手動で変換して個数の変化を確認して下さい。ラベル画像も表示してみましよう。胸部のラベル画像では, 肺野が体の外とつながり 1つのラベルになっていることが確認できると思います。

<実験8-2: 他の画像でも実験(2)>

12circles2.rawにノイズを加えた画像12circles3.rawを準備しました。この実験では, ノイズとぼけのある画像中に存在する円を正しく数えることができるのか挑戦しましよう。つまり, 正しく12個と表示できるようにノイズ除去を考え, 処理に付け加えましよう。

ラベル数が多くなるのはここではノイズが主な原因ですから, ノイズの除去を考えます。

ノイズ除去の方法は, 濃淡画像についてと2値画像についての2つに分けることができます。濃淡画像については演習4の3×3の平滑化フィルタが利用できます。3×3の領域を大きくすれば, 平滑化の効果が大きくなりノイズが除去できます。

2値画像については白い領域の収縮処理が利用できます。収縮処理を何度か繰り返すことで小さな領域は消えていきます。消えてしまった後で膨張処理を行えば, 再び小さな領域が発生することはありません。

このような方針でプログラムの改造を行いましよう。

円の濃淡が違うので, しきい値によっては3個, 6個, 9個と出力されるはずです。プログラム中では平均値で2値化していますが, scanfで値を入力するように改造して実験しましよう。

```
% xlabelview out.raw 512 512 [ラベル数]
```

引き続きno.8ディレクトリで作業します
実行:

```
% proc 12circles3.raw 512 512 out.raw
```

改造なしで実行すると, ラベル数が多くエラーになります。

ノイズ除去を行った後に2値化を行い, また, 何度か収縮処理を行うことで小領域を削除しましよう。

また, うまくいったときのラベル数をメモして, ラベル画像も表示してみましよう。

ラベル画像の表示

```
% xlabelview out.raw 512 512 [ラベル数]
```

私(原)が実験した例をno.8-2に示します。3x3で濃淡画像の平滑化を行った後, erosion, dilationを行い小領域の削除を行いました。その結果は, 14個でしたが, 正しく取っているのは9つでした。いちばん淡いところの3つは数えられませんでした。残念。

ここで, Coffee Break.ひとやすみ: 新設大学院の紹介です

平成14年4月から, 岐阜大学大学院に「医学研究科・再生医科学専攻」が設置されました。修士課程, 博士課程が設置の予定です。その中に, 知能イメージ講座が設置され, CAD関連の研究, 画像処理, 認識, 分子イメージングなどの専門分野の研究・教育を行っています。興味あります方は,

藤田教授: fujita@fjt.info.gifu-u.ac.jp
もしくは, 原: hara@info.gifu-u.ac.jpまで。

岐阜大学新病院(2004年)



演習 9 : 特徴抽出処理

この演習はprogramディレクトリのno.9, no.9-2というディレクトリの中で行います。

この演習では、ラベルづけされた画像からいくつかの特徴を抽出する手法を取り扱います。ラベルごとにその特徴を抽出することは、ラベルの形状の判定などを行う上で非常に基本的な処理になります。ラベルや領域の特徴のことを画像特徴や画像特徴量、もしくは単に特徴量と表現します。

特徴量には、面積や画素値の平均などといった非常に基本的なものから、領域の円形らしさを示す円形度、領域内の濃度むらを示す画素値の標準偏差など数かぎりなく考えられます。

この演習では、まず、基本的な特徴として、重心、面積を求めます。そして、次には領域の円形らしさを表現する円形度を求めます。

図形の重心は、領域内の座標の平均で定義されます。つまり、図形内の点(すなわちラベルがついている点)のx座標とy座標とをすべて加算して足した数で割れば計算できます。面積は、画像中では画素数で表現されますから、数えた数が面積になります。実際の面積に換算するのならば、1画素が何[cmxcm]を計算してかければよいです。

次に円形度を求めましょう。円形度はいろいろ定義できますが、円形のものほど1に近くなるように定義しなくてははいけません。もっとも計算しやすい方法の1つとして、次のアルゴリズムがあります。

1. ラベル番号:labelnum の重心にラベルと同じ面積をもつ円があったと仮定します。
2. 円形度をその円とラベル部分の共通に存在する面積の比率として定義します。

つまり、ラベルの面積を「s1」ラベルとその面積と等価の円の共通部分の面積を「U」とした場合、円形度をU/s1で表すということです。この場合、ラベル図形と円が完全に一致すれば値は1になり、円形から外れるほど値が小さくなります。このような計算を行うためには、前の重心の計算、ラベルの面積の算出が必要になり、その結果をもとに、さらに追加の円形度の計算を行う関数を作成します。

< 実験9-1 : 重心・面積の算出 (1) >

no.9のディレクトリに移動後、以下の作業を行い実験を行います。対象は12circles.rawで行いましょう。

まず、原画像の表示、確認、

そして、コンパイル、

次に、実行、

を行って下さい。実行結果では、12個のラベルが解析されて、それぞれの重心、面積が表示されます。

また、main.cの中身を見て、全体の流れをつかんで下さい。

関数labelsizeという関数が、ラベル画像、ラベル番号を受けてそのラベルの重心と面積

```
% cd no.9
% x8view 12circles.raw 512 512
% make
% proc 12circles.raw 512 512 out.raw
% mule main.c
```


を調べます。center_and_size.cというファイルに書かれていますので、内容を読んで下さい。

< 実験9-2：重心・面積の算出（2） >

実験9-1と同じ実験を画像を変えて行いましょう。使う画像はcircle.rawです。これは半径50pixelsの円が埋め込まれていて、その面積はおよそ7,850pixelsになります。実行結果ではどうなりますか？確認しましょう。

< 実験9-3：重心・面積の算出（3） >

実験9-1と同じ実験を画像を変えて行いましょう。12circles2.raw, 12circles3.raw, chest.raw, の3種類を入力にして実験してみましょう。そして、ラベル画像も表示して12circles.rawと12circles2.rawの面積の違いを確認しましょう。circles2.rawはボケを持っているため、領域が広く取られるので、面積が大きくなる傾向を確認できると思います。

胸部の画像の結果では、実験8-2で確認した肺野と外部がつながってしまった非常に大きい部分の面積が求められていることが分かります。

< 実験9-4：円形度の計算（1） >

no.9-2のディレクトリへ移り、実験を行います。ここでは、円形度を求めています。円形度は、roundshape.cというファイルの中の、roundshapeという関数で計算します。ここでは、ラベル画像とラベル番号を渡し、その番号の円形度を出力します。まず、circle.rawを入力して円形度を計算してみましょう。また、計算の基準となった2つの面積を両方表示するように改造しましょう。

< 実験9-5：円形度の計算（2） >

画像を12circle.raw, 12circle2.raw, 12circle3.rawに変更して実験しましょう。12circle3.rawもエラーなく計算できるように改造しましょう。

```
% mule center_and_size.c
```

半径50画素の円の面積（画素数）は普通の円の面積（半径×半径× π ）を求めるように計算できますが、実際にはデジタル値なので多少の誤差があります。

実行：

```
% proc circle.raw 512 512 out.raw
```

結果出力は、Size = 7860

となると思います。

実行例：

```
% proc 12circles2.raw 512 512 out2.raw
```

```
% proc 12circles3.raw 512 512 out3.raw
```

```
% proc chest.raw 512 512 out4.raw
```

ラベル画像表示例：

```
% xlabelview out2.raw 512 512 12
```

など、出力に合わせる。ただし、12は計算されたラベル数です。

12circles2.rawはラベル数12。

12circles3.rawはエラー発生。

chest.rawはラベル数19。

ラベルの面積の算出には実験9-2での関数を使っています。

コンパイル：% make

実行：% proc circle.raw 512 512 out.raw

結果：円形度は0.9964となりました。

関数：roundshapeの中で計算しています

ちなみに、ラベルの面積はSizeとして表示済みです。

ヒント：

ラベル数が多いのがエラーの原因。

大元はノイズが多いから。したがって、ノイズ除去（実験8-2）の結果を利用しましょう。

<まとめ>

今回のセミナーでは、基本的であり、CADのための重要な要素を取り扱いました。まず最初に取り扱った画像の2値化は、処理自体は非常に簡単な処理です。しかし、ここでのしきい値の決定は、もっとも困難な問題になります。2値化処理は、画像中に存在する物体の領域を決める（領域分割）の基本手段ですので、非常に重要な処理です。画像関係の研究分野では、このような領域分割の問題は1つの代表的な研究テーマになるほどです。

2値化を行えば、後は膨張収縮処理などのモルフォロジ処理、ラベリング処理、特徴抽出処理とつなげていくことができます。そして、今回の演習では、その内容まで取り扱いました。これによって、画像中に存在する物体を分離してその特徴を調べる（例えば、面積、円形度）ことができるようになりました。

これからの課題には、抽出した特徴に基づいてラベルを取り除いたり、原画像にもどって、濃淡値から別の特徴を抽出するという処理があげられます。

今回の課題では、医用画像から少し離れて画像中に存在する円形の淡い領域の数を数えました。これは、CAD風に表現すれば、画像にある白い領域（きっと病変部）の検出を行ったということになります。検出位置は重心です。そして、間違えてラベルを付けたところが偽陽性候補ということでした。

以上のように、CADは今回取り扱った内容を中心技術として、それらをさらに拡張することによって構築できます。基本的な技術はそれほど難しいことではないと理解していただけたと思います。

次のセミナーでは、濃淡画像の特徴量を取り扱う予定でいます。

<付録>

配付したCD-ROMのprogramsディレクトリ中にあるx8viewディレクトリには、演習で用いた画像表示プログラムが保存されています。Makefileも一緒に保存してありますので、コンパイルして利用ください。作成者は、岐阜工業高等専門学校/福岡大輔先生です。セミナーの利用および自宅での学習の範囲においての利用は快諾をいただいています。LINUX上でのコンパイルには、MesaGL, GLUTが必要です。

一般に、Linuxのインストール時に「Graphic Workstation」と「Developer Tools」を入れる、もしくは指定すれば最初から動作するようになるようです。

また、ラベル画像表示のプログラムxlabelviewもこのx8viewを元に作りました。

コンパイル方法 : make

実行方法 : x8view 表示画像ファイル名 幅(画素数) 高さ(画素数)

この冊子は、岐阜大学工学部 原 武史が作成しました。

CARS 2002

Computer Assisted Radiology and Surgery
16th International Congress and Exhibition

<http://cars-int.de/>

June 26-29, 2002, Paris, France (CAD関連の国際会議)

言葉で示します

LINUX上での入力を示します

言葉で示します

LINUX上での入力を示します