

第9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9月21日 1pm-5pm, 22日 10am-5pm

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，岐阜大学で開催した2回のCADセミナー（第3回，第6回）の続きとして，さらに内容を拡張して企画しました。

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，これまでのセミナーの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>

第9回： <http://www.fjt.info.gifu-u.ac.jp/cad-sem9>

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>

<9月21日（1日目）>

12:20-13:00 受付

13:00-13:20 開始のご挨拶・ガイダンス（1階）

岐阜大学・藤田広志，原 武史

13:20-15:00 体験演習・一連の流れの演習（7階）

15:10-15:30 演習A：前処理に関する講義（1階）

講師：岐阜高専・畑中裕司

15:30-15:50 演習B：検出処理に関する講義（1階）

講師：岐阜高専・福岡大輔

16:00-17:00 演習：前処理・検出処理（7階）

希望の方は研究室の見学を行います。

17:10-17:20 明日の予定についての説明（1階）

担当：岐阜大学・原 武史

18:00-20:00 懇親会（17:30頃バスが発射します）

<9月22日（2日目）>

10:00-10:10 2日目ガイダンス

岐阜大学・原 武史

10:10-10:30 演習C：偽陽性削除処理（1階）

講師：名古屋文理大学・松原友子

10:30-10:50 演習D：検出性能の評価（1階）

講師：岐阜大学・原 武史

11:00-12:00 演習：偽陽性候補削除/性能評価（7階）

12:00-13:00 昼食（受付時に有料で承ります）（1階）

13:00-16:00 自由演習（7階）

16:00-16:30 成果発表（1階）

16:30-16:50 まとめと更なる研究のために

講師：岐阜大学・原 武史

16:50-17:00 セミナー終了のご挨拶

岐阜大学・藤田広志，原 武史

<セミナースタッフ>

画像分科会：原 武史（岐阜大学） 中部部会：津坂昌利（名古屋大学）

学術委員会：藤田広志（岐阜大学），松原友子（名古屋文理大学）

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

牛場洋明，篠原範充（岐大D3）

この資料は，第9回CADセミナーにおいて配付するものです（9/20/2002作成）。

また，セミナーで用いた画像の一部には，イギリス・The Mammographic Image Analysis Society（MIAS）データベースおよび日本放射線技術学会発行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。

1 . セミナーの概要と目的

今回のCADセミナーは、これまでのCADセミナーに参加して修得したプログラミングの技術を用い、CADアルゴリズムの全般的な流れを理解することを目的とします。そのために、学習用に簡便に作成したCADプログラムを用いて乳房X線写真上の腫瘍陰影を自動的に検出するシステムを、受講者自身が順序立てて構築します。

このセミナーではCADのプログラムを4つの段階にわけて取扱います。それぞれの段階において短時間(約20分)の講義を行い、教科書などに書かれた処理や集計結果と、実際のプログラムによる実現例と作業との間を埋めていきたいと考えています。

また、セミナーの最後に参加者の成果発表の時間を設定し、可能な方は口頭で処理内容と検出性能を発表していただきます。また、そのための演習として、2日目の午後に自由演習の時間としました。

これら2日間の演習と講義を通じて、受講者はCADの研究の進め方と重要なポイントの多くを理解し、体験することができると考えています。

2 . セミナーで用いる画像

今回のセミナーでは、体験画像とMIAS画像の2種類を使います。

< 体験画像 >

これは練習・体験用の人工画像3枚です。主に体験演習で用います。1024 x 1024画素、と512 x 512画素の2種類、8ビットの画像です。原が演習用に作成しました。配付するCD-ROMにも保存されています。

保存場所 : cad-sem9/data/phantom

ファイル名 : nphantom.raw, mphantom1.raw, mphantom2.raw

nphantomは何も検出対象がない画像。mphantom1は検出対象を1つ含む画像。

mphantom2は検出対象と検出したくない対象を複数含む画像。

nphantoms.rawのようにsがつく画像は、512 x 512画素(体験演習で利用)

< MIAS画像 >

これは演習用の画像で、イギリス・MIASデータベースの無料版から、演習に最適と考えられる典型的な悪性腫瘍症例20例と正常例20例を選択しました。再配布ができないので、配付するCD-ROMには保存されていません。

1024 x 1024画素と512 x 512画素の2種類、8ビットの画像です。

保存場所 : cad-sem9/data/malignant 明らかな悪性腫瘍を含む画像が20枚

cad-sem9/data/normal 正常例が20枚

ファイル名 : mdb023.rawのようにmdbの後に数字が3桁つきます。

mdb023.rawのように数字だけの画像は1024 x 1024画素(演習Aで利用)

mdb023s.rawのようにsがつく画像は、512 x 512画素(体験演習で利用)

体験演習：まずは全体をながめてみましょう

この演習はprogramディレクトリの「taiken」ディレクトリの中で行います。

また、検出対象が明確な3つの人工画像を用いて演習を行います。

今回のCADセミナーで構築する完成版の一例がこのディレクトリに保存されています。まずは、細部の説明へ入る前に全体を見通して、一度動かしてみましよう。

作業の流れは、

- ・原画像を表示して、どのような画像か、どの部位を検出するかを確認する
- ・プログラムを動かして、原画像を処理してみる
- ・処理結果の画像を表示して、目的の部位が抽出されているか確認する

になります。

実験準備：

ktermを開きprogram/taikenディレクトリにいることを確認して下さい。

<実験・体験 - 1：原画像の表示>

まず3種類の原画像を表示して見ましよう。そして、何を検出するのかを決めましよう。

ファイル名：nphantoms.raw, mphantom1s.raw, mphantom2s.raw

<実験・体験 - 2：流れを見る>

プログラムのmain関数を見て、全体の流れをつかみます。細部にはこだわりません。muleでmain.cを開き、一番下にあるmain関数の中を読んでください。

/* ** 準備 **** */ というところが前準備の部分になります。配列の大きさを確保し、原画像を配列に読み込んでいます。

/* ** 検出処理 **** */の部分で1次検出を行います。ここでは画像の平均値と標準偏差からしきい値を決める2値化処理を行っています。平均 + 標準偏差の1.7倍がしきい値になっているのがわかりますか？

/* ** ** 偽陽性削除処理 **** ** */の部分で、前

kterm内で

% pwd と打って、...../programs/taiken と表示されればOKです。

表示プログラム：x8viewを使います。

x8view ファイル名 幅 高さ
を入力すると画像が表示できます。

```
% x8view ../../data/phantom/nphantoms.raw  
512 512
```

エディタ：muleを使います。

```
% mule main.c
```

malloc関数で確保

image_read_char関数で配列に読み込む

get_average関数で画像の平均値を算出

get_var関数で画像の分散を算出

binarize_image関数で2値化を実行。

thresh1 + 1.7*sqrt(var1)がしきい値として与えられている。

の段階で拾いすぎた候補を削除する処理を行います。ここでは、まず2値画像のラベリングを行い領域を数え番号付けを行います。そして、それぞれの領域に対して、

- ・領域（ラベル）の重心座標（ x, y ）
- ・面積（大きさ：ピクセル数）
- ・丸さ（円形度：0~1の値）

- ・領域内の画素値の平均値

の特徴をそれぞれの関数で調べ、それらの値を保存する配列（feature）に格納しています。このようなラベルの特徴（大きさや画素値の平均など）の値のことを特徴量といいます。

その後、fp_reductionという関数で配列featureを解析して、不要な領域を削除します。これによって、拾いすぎ候補を削除します。

fp_reduction関数では、残したいラベルの特徴量について範囲を定め、その範囲を満たすラベルだけ残す処理を行っています。その条件は3つあり、

- ・ラベルの大きさの下限
- ・円形度の下限
- ・平均値の下限

です。この3つの条件のうち1つでも満たさない特徴量をもつラベルは除去されます。その結果、条件を満たさない候補は偽陽性として削除されることとなります。

/* **ラベル画像の保存** */の部分で、最終結果をファイルに出力しています。

< 実験・体験 - 3 : コンパイルする >

このディレクトリには、ソースファイルしか保存されていません。このプログラムを実行するためにはコンパイルする必要があります。コンパイルには、いくつか方法がありま

labeling関数でラベル付けを行います。この関数はつけたラベルの数の合計を整数値で返します。

labelsize関数でi番目のラベルのついた領域を解析し、重心と面積を求めます。

roundshape関数でi番目のラベルの円形度を求めます。

labelpixvalue関数でi番目のラベル内の画素値の平均値を求めます。

配列featureはmain.cの先頭部分に外部変数として宣言しています。256行4列の2次元配列になります。行がラベル番号、列が特徴量が保存される場所になります。

いったん、main.cを表示しているmuleを終了して、次にfp_reduction.cを開きましょう。

```
% mule fp_reduction.c
```

```
#define MIN_SIZE 1400 で決定
```

```
#define MIN_ROUND_RATIO 0.6 で決定
```

```
#define MIN_PIXVALUE 180 で決定
```

for文でラベル番号を動かし、if文で条件判定を行う。

remove_label関数でラベルの除去を行う。

image_write_char関数でラベル画像を単純に保存しているだけです。ラベル画像は濃淡で表現され、濃度はラベル番号の数字に一致します。

Makefileがあるので、単純に

```
% make
```

と入力すればコンパイルが始まる。

```
% make clean
```

すが、この演習ではコンパイル方法を記述した「Makefile」を用意しているので簡単にできます。

コンパイル後にできる実行ファイルの名前は「proc」です。

<実験・体験 - 4 : 実行する>

コンパイルしたプログラム「proc」を実行します。入力には人工画像を用いましょう。512x512画素の画像がよいです。3枚の人工画像について、それぞれ結果のファイルを作成しましょう。出力画像のファイル名は、out1.raw, out2.raw, out3.rawなどなんでも構いません。でも、nphantoms.rawの出力ならnphantoms.out.rawなどとファイル名の対応がつくほうが理解しやすいです。

<実験・体験 - 5 : 結果を観察する>

体験 - 4 で作成した3つの結果画像を観察して、自分が検出したかった陰影が表示されるかどうかを確認します。表示プログラム「xlabelview」はラベル画像に適切な色を割り振って表示します。目的の陰影は出力画像に残っていますか？

<実験・体験 - 6 : 検出性能を求める>

体験 - 5 に基づき、正しく検出されたかどうかを数え検出性能を求めましょう。

真陽性率 (True-positive rate : TP) =
(正しく検出できた個数) ÷ (正しく検出したい個数)

真陰性率 (True-negative rate : TN) =
(正しく検出しなかった画像枚数) ÷
(検出する必要のない画像の枚数)

画像あたりの偽陽性数 (False-positives per image : FP) =
(間違えて検出した個数) ÷ (全体の画像枚数)

と入力すると、何もコンパイルしていない状態に戻る。

1 枚目

```
% proc ../../data/phantom/mphantom1s.raw  
512 512 out1.raw
```

2 枚目 :

```
% proc ../../data/phantom/mphantom2s.raw  
512 512 out2.raw
```

3 枚目 :

```
% proc ../../data/phantom/nphantoms.raw  
512 512 out3.raw
```

出力画像 : out1.raw, out2.raw, out3.raw

表示方法 :

```
% xlabelview ファイル名 幅 高さ
```

例 : % xlabelview out1.raw 512 512

out1.rawのラベル数 : 0

検出できず

out2.rawのラベル数 : 4

1 つは検出できた。

3 つは拾いすぎ

out3.rawのラベル数 : 0

正しく検出しなかった

これらから計算すると検出性能は、

TP = 1/2, TN = 1/1, FP = 画像あたり3/3

(つまりTP=50%, TN=100%, FP=画像あたり1個)

以上で全体の体験は終了です。以下の8項目をセルフチェックしましょう。

: x8viewで画像は表示できましたか？

: muleでプログラムは読みましたか？

: makeでコンパイルできましたか？

: プログラムは実行できましたか？

: xlabelviewで結果は表示できましたか？

: 検出目的の陰影を結果画像に探せましたか？

: 検出結果から検出性能を計算できましたか？

: 性能向上の方法を予想できますか？

演習 A : 前処理

この演習はprogramディレクトリの「a」ディレクトリの中で行います。

CADなどの高度な画像処理では、処理をひとまとめに考えず、

「前処理」、「中間処理」、「後処理」と、複数の段階にわけて構成する事が一般的です。これは、複数の処理を一度に考案しても、その整理や機能の説明が困難になったり、効率のよい修正や改造が行なえなくなったりするからです。また、それぞれの中でもできるだけ機能が分かりやすい単位で処理を構成する事が重要といえます。これは、C言語の中では関数という単位で処理を構築する事と非常によく似ています。

もちろん、処理の内容によってはその処理の種類が前処理や中間処理などと明確に区別ができるものではありません。しかし、これからつくる処理がどの段階の処理にあるのか理解する事は非常に重要です。

また、実験や研究を行いやすくするためにも、自分が使いやすい形で画像を処理しておくことも前処理に含まれます。つまり、事前に前処理を行った画像を保存しておき、普段は中間処理から作業を始める、ということができるようになります。

CADのプログラムにおいては、

前処理 : 画像の転送, 入力, 解像度・濃度分解能の変更, ノイズ除去処理など

中間処理 : 候補の検出処理, 偽陽性削除処理

後処理 : 検出結果の表示処理

などがあげられます。

この演習では、前処理として、以下の手順で処理を行なうプログラムを構築します。原画像は、M I A Sデータベースの画像であり、乳房X線写真をイメージスキャナによりデジタル化してあります。用いる画像のサンプリング間隔は0.2mm, 濃度分解能8ビットの画像です。

実験準備:

ktermを開きprogram/aディレクトリにいることを確認して下さい。

< 原画像の表示 >

原画像を表示してみましょう。画像の大きさを確認しましょう。右の例が表示できたら他の画像も表示してみましょう。

ファイル名の数字の最後にsがついていない1024 x 1024画素の画像を表示しましょう。mdb023s.rawと「s」のついている画像は512 x 512画素の画像です。

kterm内で

% pwd と打って,/programs/a
と表示されればOKです。

表示プログラム: x8viewを使います。

癌症例:

% x8view ../data/malignant/mdb023.raw
1024 1024

正常例:

% x8view ../data/normal/mdb056.raw 1024
1024

< 前処理のプログラムの実行 >

サンプルの前処理のプログラムは、

- ・ 画像を読み込み
- ・ 3x3画素の平滑化を行い
- ・ 画像の解像度を半分に落とし
- ・ 保存する

という流れになっています。

まず、main.cを見てmain関数で全体の流れを把握した後に、プログラムをコンパイルして、実行してみましょう。

実行結果のファイル名は、「.raw」の前に「s」をつける事とします（例：mdb023s.raw）。

実行したら結果画像を表示して、解像度が半分になっていることを確認しましょう。

< 演習 A - 1 >

convolve_smooth関数を改造して、平滑化の大きさを5x5にしてみましょう。

< 演習 A - 2 , 3 >

すべての画像の解像度を原画像の0.2mmから0.4mmへ変更しましょう。といっても、何度も前のようにコマンドを入力するのは疲れるし、間違えます。そのために、「シェル」という機能を使って、作業を自動化します。

方法は簡単で、キーボードから入力することをそのままmuleで適当なファイル名（例えばconvert.sh。最後のshはシェルの意味）に保存します。つまり、いっぱいコマンドをコピーして、ファイル名だけ書き換えれば繰り返しの作業が間違いなく実行できます。また、間違えてもやり直しは簡単です。

convolve_smooth関数で実現
resampling関数で実現

main.cを見る：% mule main.c

コンパイル　：% make

実行：

```
% proc ../../data/malignant/mdb023.raw 1024
```

```
1024 mdb023s.raw
```

結果確認：

```
% x8view mdb023s.raw 512 512
```

解答・方法は省略。

```
% mule convert.sh
```

として始めます。シェルスファイルということを示すために、先頭に「#/bin/sh」と書くのがよいでしょう。そのファイルの中にコマンドを書いていきます。

改行して続ければ、上から順番に実行してくれます。

以上で演習 A は終了です。以下の 10 項目をセルフチェックしましょう。

- | | |
|-------------------------|---------------------------|
| ： x8viewで画像は表示できましたか？ | ： 出力画像が1/4になったことを確認しましたか？ |
| ： 解像度という言葉は理解できましたか？ | ： 平滑化は理解できましたか？ |
| ： 画素という言葉は理解できましたか？ | ： 畳み込みは理解できましたか？ |
| ： 解像度を変更するという意味はどうですか？ | ： 平滑化を5x5に改造できましたか？ |
| ： サンプルをコンパイルして実行できましたか？ | ： シェルを使ってコマンドを実行できましたか？ |

演習 B : 検出処理 : 乳房 X 線写真上の腫瘤陰影の検出

この演習はprogramディレクトリの「b」ディレクトリの中で行います。

このCADの中核となる腫瘤陰影の検出処理を構築します。これは、先の「前処理」、「中間処理」、「後処理」の3つの段階の2つ目にあたります。

検出の方針は、つぎのようにします。

腫瘤陰影は、背景と比べて比較的明るい(白い)領域にあるので、あるしきい値で2値化(演習6参照)を行うことで検出する事とします。

そのしきい値の決定は、何らか自動的にこなうことが必要です。そうしないとCADにはなりません。また、どのような画像についても妥当なしきい値が決まるような手法である必要もあります(実はそれがいちばん難しいところです)。

この方針は、画像の平均値と標準偏差からしきい値を算出する手法とします。そして、2値化後の画像を保存することとします。しきい値は、画像の平均値と標準偏差から算出される値とします。平均を m 、標準偏差を s_d とするとき、しきい値： t_h は

$$t_h = m + x \cdot s_d$$

とする事とします。これで x をパラメータとしてしきい値を変更できるようになります。

なお、この演習以降の処理では、演習Aで解像度を半分に落とした画像(mdb??s.rawとして保存した画像)を入力として利用することとします。演習Aが最後までできなかった人でも作業できるように、念のため、あらかじめ画像は作成して保存してあります。

実験準備：

ktermを開きprogram/bディレクトリにいることを確認して下さい。

<原画像の表示>

解像度を変更した原画像を表示してみましよう。特に腫瘤陰影を含む画像を表示して、腫瘤陰影らしい領域を確認して下さい。

<検出プログラムの実行>

サンプルの前処理のプログラムは、

- ・画像の読み込み
- ・画像の平均値と標準偏差の算出
- ・2値化処理
- ・2値画像の保存

という流れになっています。

まず、main.cを見てmain関数で全体の流れを把握した後に、プログラムをコンパイルして、実行してみましよう。

実行結果のファイル名には、「b」をつける事とします(例：mdb023sb.raw)。

kterm内で

% pwd と打って、...../programs/bと表示されればOKです。

表示プログラム：x8viewを使います。

癌症例：malignantの中にあります。

正常例：normalの中にあります。

(演習1・演習A)

get_average, get_var関数で実現(演習6)

binarize_image関数で実現(演習6)

(演習1・演習A)

main.cを見る：% mule main.c

コンパイル：% make

実行：

実行したら結果画像を表示して、白黒で白い領域がいくつか存在する画像となることを確認しましょう。また、典型的な腫瘍陰影を含む画像（例：mdb134s.raw）は、腫瘍陰影が孤立した白い領域として表示されることを確認しましょう。

< 演習 B - 1 >

しきい値を { 平均 + 標準偏差の1.5倍, 2倍など } と自由に变化させてみましょう（サンプルは1.5倍）。ポイントは、腫瘍陰影らしい領域ができるだけ孤立して白く表示されるような値を見つける事です。mphantoms.raw, nphantoms.rawの2枚の人工画像も使って実験してみましょう。

< 演習 B - 2 >

しきい値の変数を#defineを使ってmain関数の先頭部分に分かりやすく表示しましょう。このように数字をプログラムの外に出しておくと数値を変更する場合に非常に便利です。ただし、数値を変更した場合には、コンパイルし直すことを忘れないようにしましょう。

< 演習 B - 3 >

画像の画素値の平均値と標準偏差を画像の背景（真っ黒の部分）を無視して計算するように改造しましょう。

< 余裕のある方：演習 B - 3 >

画像全体からしきい値を求める事をやめて、中心領域がその周辺よりも一定値以上白い場合には白とするようなプログラムを作成しましょう。

```
% proc ../../data/malignant/mdb023s.raw 512
512 mdb023sb.raw
結果確認：
% x8view mdb023sb.raw 512 512
```

muleでmain.cを開き
% mule main.c
main関数中のbinarize_imageの引数を変更します。引数の意味は、binarize.cに書いてあります。

先頭に
#define ALPHA 1.5
と書いて、いままで数値を書いていた所をALPHAという文字に置き換えます。このようなときにはすべて大文字とすることがよいとされています。

get_averageの中で、画素値が背景（すなわち0）であったときには計算しないようなif文を入れる。get_varも同様。

解答省略

以上で演習 B は終了です。以下の9項目をセルフチェックしましょう。

- | | |
|-------------------------|--------------------------------|
| : 悪性症例は表示できましたか？ | : 人工画像の腫瘍は白い領域でしたか？ |
| : 正常症例は表示できましたか？ | : #defineをつかって数値を外に書けましたか？ |
| : 腫瘍陰影を目でみつけられましたか？ | : if文を使った背景とそれ以外の領域の判定はできましたか？ |
| : サンプルをコンパイルして実行できましたか？ | |
| : 典型的な腫瘍は白い領域でしたか？ | |

演習 C：偽陽性削除処理：1 次検出後の拾いすぎ削減

この演習はprogramディレクトリの「c」および「c-2」ディレクトリの中で行います。削除の方針は、以下のようにします。

まず、検出処理後の2値画像に対してラベリング処理（演習8参照）を行ない、そのラベルごとに本物の腫瘍陰影といえるか否かを以下の条件で判定します。

- ・丸くないものを削除：円形度で判定
- ・小さすぎるものは削除：面積で判定
- ・暗い陰影を削除：画素値の平均で判定

また、2値化処理の後に非常に小さな領域が多数発生する場合には、ラベリング処理の前に膨張収縮処理（演習7参照）が必要になります。これは、演習で用いているラベリングのプログラムは、ラベル数が1画像あたり255までしか対応しないためです。

処理は前の処理に付け加える形で構築します。入力画像には演習Bと同じ縮小画像を用います。演習Bの出力画像ではないので注意して下さい。

したがって、処理の流れは

- ・画像読み込み（演習B）
必要となる知識：画像の平均値と標準偏差の算出（演習6, 演習6-2）
2値化処理（演習6, 演習6-2）

- ・偽陽性削除処理
ラベリング
ラベル毎の特徴量の表示

- ・ラベル画像の保存

となります。

また、以上のプログラムをたたき台として、

- ・新しい特徴量を計算するプログラムを作成する
- ・特徴量の判定を行うプログラムを作成する
- ・ラベルを削除するプログラムを作成する

ことによって、一番最初の体験演習と同じプログラムを構築することを目標とします。（解答例がc-2ディレクトリに保存されています）

実験準備：

ktermを開きprogram/cディレクトリにいることを確認して下さい。

< 腫瘍陰影の位置の確認 >

腫瘍陰影を含む画像を表示して、腫瘍陰影らしい領域を確認して下さい。

< 検出プログラムの実行 >

サンプルのプログラムは、
メイン関数：main.c

kterm内で

% pwd と打って、...../programs/c
と表示されればOKです。

表示プログラム：x8viewを使います。

癌症例：malignantの中にあります。

正常例：normalの中にあります。

ラベリング関数：labeling.c
中心と大きさを調べる関数：center_and_size.c
円形度を調べる関数：roundshape.c
からできています。

まず，main.cを見てmain関数で全体の流れを把握した後に，プログラムをコンパイルして，実行してみましょう。

実行結果のファイル名には，「l」をつける事とします（例：mdb023sl.raw）。

実行したらxlabelviewを用いて，結果画像を表示します。

< 演習 C - 1 >

ラベル内の（原画像の）画素値の平均を求める関数を作成しましょう。

< 演習 C - 2 >

任意のラベル番号がついたラベルを消すプログラムを作成しましょう。

< 演習 C - 3 >

ラベルの面積（大きさ），円形度，画素値の平均のそれぞれの3つの値が，それぞれともある一定の値以上の値であるラベルだけ残すようなプログラムを作成しましょう。そして，そのラベル画像を出力するようにしましょう。

これができれば，簡単な偽陽性削除ができるようになり，もっとも簡単なCADのプログラムの完成となります。

main.cを見る：% mule main.c

コンパイル：% make

実行：

```
% proc ../../data/malignant/mdb023s.raw 512  
512 mdb023sb.raw
```

ラベル画像の表示には，xlabelviewを使います。使い方：

```
% xlabelview ラベル画像 幅 高さ
```

ラベルごとに色分けして表示されます。

例：% xlabelview mdb023sl.raw 512 512
典型的な腫瘍陰影を含む画像（例：mdb134s.raw）の結果では，腫瘍陰影が孤立した領域として表示されているが，それ以外にもいくつかの領域に色がつけられていることが確認できます。

解答：

「c-2」ディレクトリlabelpixvalue.c

解答：

「c-2」ディレクトリremove_label.c

解答：

「c-2」ディレクトリ全体。

コンパイル：make

実行：同じ

方針：

特徴量をいったん配列に保存して，その配列から削除するラベル番号を決める関数を構築する（解答例では配列はfeature[256][5]という2次元配列。関数はfp_reduction.cに書かれている内容）。

以上で演習Cは終了です。以下の8項目をセルフチェックしましょう。

：サンプルをコンパイルして実行できましたか？
：ラベリングの意味は理解できましたか？
：特徴量という言葉の意味は理解できましたか？
：ラベルごとに特徴量が求まることは理解できましたか？

：if文の判定基準に特徴量を使えましたか？
：if文の条件が複数ある判定はできましたか？
：腫瘍陰影の特徴は確認できましたか？
：配列に入れた特徴量は理解できましたか？

演習 D : 検出性能の評価 : 性能向上のために

この演習はprogramディレクトリの「d」ディレクトリの中で行います。

この演習は、プログラミング作業ではなく、忍耐づよいパラメータ調整の作業からなります。グラフ用紙を用意して特徴量を平面上にプロットすることが重要です。そして、真陽性率、真陰性率、画像あたりの偽陽性数を算出することがこの演習の目的です。

まず、もう一度用語と計算方法の定義です。

真陽性率 (True-positive rate : TP) =
(正しく検出できた個数) ÷ (正しく検出したい個数)

真陰性率 (True-negative rate : TN) =
(正しく検出しなかった画像枚数) ÷
(検出する必要のない画像の枚数)

画像あたりの偽陽性数 (False-positives per image : FP / img) =
(間違えて検出した個数) ÷ (全体の画像枚数)

この3つの値を自分で作成したプログラムについて求めてもらいます。つまり、fp_reduction.cの先頭の3つパラメータを変更し、また、fp_reduction.cの中に独自の条件式 (if文) を付け加えることによって、TPが高くFP/imgが少なくなるように改造を行ってもらいます。新しい特徴量を導入していただいても結構です。

そして、その結果を2日目の午後にその成果発表を口頭で行っていただきます。発表していただきたい内容は、

- ・ 処理全体の流れ
- ・ 追加した処理の内容
- ・ FP削除の条件
- ・ 検出性能 (TP, TN, FP/img)

です。地道な作業が必要ですので頑張ってください。

実験準備 :

ktermを開きprogram/dディレクトリにいることを確認して下さい。

< サンプルプログラムの実行 >

サンプルのプログラムは、演習Cの解答例です。c-2ディレクトリと同じものがdにも保存されています。

メイン関数 : main.c

ラベリング関数 : labeling.c

kterm内で

% pwd と打って、...../programs/dと表示されればOKです。

中心と大きさを調べる関数：center_and_size.c
円形度を調べる関数：roundshape.c
画素値の平均を求める関数：labelpixvalue.c
任意のラベル番号がついたラベルを消す関数：remove_label.c
からできています。

まず，main.cを見てmain関数で全体の流れを把握した後に，プログラムをコンパイルして，実行してみましょう。

実行結果のファイル名には，「r」をつける事とします（例：mdb023sr.raw）。

実行したらxlabelviewを用いて，結果画像を表示します。ラベルごとに色分けして表示されます。

< F R O C 曲線の作成 >

いろいろパラメータを変更するとそれぞれでT P率，画像あたりのF P数が求まります。それを縦軸をT P率，横軸を画像あたりのF P数とした平面にプロットするとF R O C 曲線ができます。ただし，点の結び方には注意が必要です。

< 性能向上のためのポイント >

- ・ 1次検出では正しく候補として検出されていますか？
- ・ 1次検出の前に何か処理を加えるとよくなりそうではありませんか？
- ・ 1次検出のあと現在のFP削除処理の前になんらかの処理を加えるとよくなりそうではありませんか？
- ・ 特徴量による判別に何らかの工夫ができそうではないですか？

main.cを見る：% mule main.c

コンパイル　：% make

実行：

```
% proc ../../data/malignant/mdb023s.raw 512  
512 mdb023sr.raw
```

ラベル画像の表示には，xlabelviewを使います。使い方：

```
% xlabelview ラベル画像 幅 高さ
```

いくつか点が打てますが，一番上の点を結べばよいと思います。ただし，F P数が増えてもT P率が下がるというときには他にもっとよいパラメータがあると思ってもよいでしょう。

< 左のポイントへのコメント >

- ・ 1次検出で検出されていないとTPは上がりません。
- ・ 前処理が重要かもしれません。
- ・ 小領域の削除，モフォロジ処理が有効かもしれません。
- ・ 今は単一の条件だが，2つの特徴量の平面上で直線の上下のどちらにあるのかで判定できるかも？

以上で演習Dは終了です。以下の7項目をセルフチェックしましょう。

- | | |
|--------------------------|--------------------------|
| ： 1つでも腫瘍陰影は検出できましたか？ | ： 画像あたりの偽陽性数の意味は説明できますか？ |
| ： 1枚でも正常例には検出がありませんでしたか？ | ： パラメータを変更して性能を計算できましたか？ |
| ： 真陽性率の意味は説明できますか？ | ： F R O C 曲線は作成できましたか？ |
| ： 真陰性率の意味は説明できますか？ | |

<まとめ>

今回は、これまでのCADセミナーに参加して修得したプログラミングの技術を用い、CADアルゴリズムの全般的な流れを理解することを目的として、いくつかのサンプルプログラムを用いて受講者の皆様が乳房X線写真上の腫瘍陰影を自動的に検出するシステムを構築しました。そして、多くの方はパラメータ調整とその検出性能の評価、FROC曲線の作成まで到達できたと思います。

検出精度はともかくとして、このような作業の流れはまさにCAD研究のそれとまったく同じです。一般の研究と異なる点は、用いるデータベース、検出処理、FP削除処理、特徴量、といったあとはそれぞれの独自性・発想などに起因することだと思えます。特に検出処理に関しては、いろいろな研究施設が独自の手法を考えて論文にしています。しかし、その多くの方法は、手法を分解すれば当然ですが一般的な計算手法の組み合わせでできているものです。

したがって、受講者の皆様も多くの手法についての知識を習得し、それらを有機的に組み合わせることでまったく新しい手法を開発できるチャンスがあります。このセミナーの内容をもとに皆様がCADの研究をスタートできるとするならば、主催する画像分科会にとってこの上ないよこびです。

今回のセミナーは初めての試みとして1泊2日の日程で行いました。C言語の処理に慣れている方にとっては少し物足りなさがあったかもしれません。逆に、C言語が初めてという方にとっては、常に前のセミナーの内容に戻る必要があり、時間が不足したのかもしれません。セミナーでは処理や特徴量などに関してはまったく取扱っていない内容も沢山あります。今後はそれら内容も含めるようにしますが、受講者の皆様も自学をすすめて多くの新しい処理を修得されることを強く望みます。

<付録>

配付したCD-ROMのprogramsディレクトリ中にあるx8viewとxlabelviewディレクトリには、演習で用いた画像表示プログラム、ラベル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。以前のCADセミナーのプログラムとは若干変更が加えられています。原作者は、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されています。ここに心より感謝の意を記します。なお、これら2つのプログラムの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ります。

また、ラベル画像表示のプログラムxlabelviewもこのx8viewを元に、原が作成しました。

コンパイル方法 : どちらも「make」だけ

実行方法 : x8view 表示画像ファイル名 幅(画素数) 高さ(画素数)
画像上でマウスをクリックするとクリックした座標とその画素値が表示されます。

実行方法 : xlabelview ラベル画像ファイル名 幅(画素数) 高さ(画素数)
画像上でマウスをクリックするとクリックした座標とそのラベル番号が表示されます。

終了方法 : ESCキーをおす、もしくは、「q」キーをおす。

この冊子は、岐阜大学工学部 原 武史が作成しました。

CARS 2003

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

Computer Assisted Radiology and Surgery
17th International Congress and Exhibition

June 25-28, 2003, London, UK. (CAD関連の国際会議)