ロボットビジョン

知能機械工学科3年　後期　テキストNo.1　Ver.4

担当教員：綴木　馴

はじめに

本講義の目的は，単純作業を淡々とこなすという無駄な作業を行うものではない．むしろ，このマニュアルによって与えられた，きっかけを基に，学生自ら問題点を探し出し，また，その問題解決のために自ら試行錯誤することで，真の実力を身につけることを目的としている．したがって，解説部分は極力詳細なものを付け加えた．これらの資料を基にc言語による画像処理の方法を自らの力を以て理解し，同時にそれらの有用性および問題点を見つけ出し，問題解決に取り組んで欲しい．

報告書を出すに当たっては，わざわざテキストを書き写すというような，無駄な時間と労力を費やす必要は全くない．むしろそのような時間があれば，有益な思考のための時間として使って頂きたい．

したがって，報告書を出すに当たって，このマニュアルと重複するところは，著者のホームページからダウンロードしたものを添付しても良いし，この実験マニュアルそのものをコピーしたものを報告書に添付してもかまわない．ただし，報告書は手書きではなく，TeXやWordなどのワープロを使うことが望ましい報告書のスタイルは他の読者が読みやすいものであれば，自ら工夫したオリジナルな方法で作成して構わない．レポートの「書き方が分からない」,「最低限書くべきモノは何か？」等という問いが多いが，それこそ学生自ら考えるべき問題である．初めて読む読者が，そのレポートを読んで，どういう実験を行ったのかが分かるレポートに仕上げればよい．

よって，結果や議論をまとめることには特に力を注いで欲しい．

次のページに実験の目的の概略をリストアップしておく．

1　講義の目的

本講義の目的は以下の通りである．

目的 1. 任意の画像ファイルをソフトを使ってテキスト画像ファイルに変換する．

目的 2. ライブラリの概念を理解する．

目的 3. 領域確保関数の概念を理解する．

目的 4. c 言語によりテキスト画像ファイルを配列に読み込む．

目的 5. 画像の特性を調べる

目的 6. 配列に入った画像ファイルにさまざまな加工を行う．

目的 7. 画像ファイルを出力する．

目的 8. 手際よい良い，報告書作成能力を身につける．

以上を以て，この実験に真摯に取り組み，演習・研究の能力の向上を目指す．

2　関数，ライブラリ，領域確保

まず，本稿では著者が作成した関数群を簡単に説明する．関数とは他の言語ではサブルーチンと言われているが，c 言語では特に関数と読んでいる．ライブラリとはメイン関数に書ききれないものを他のファイルに書き表したもので，下記のプログラムでは，stdio.h，stdlib.h，math.h，file.h，okimari.h，area.h，mainlib.h がライブラリである．ライブラリには関数が書かれているが，本稿ではその関数を特に理解する必要は全くない．むしろ，関数の使い方を覚えて頂きたい．プログラミングでは良く「習うより，慣れろ」と言われるが，それは，まさに的を射た言葉である．ライブラリやその中にある関数は，著者が勝手に作ったモノなので，理解しても意味がない．理解するよりも，使えるようになってもらい，今後の資源として役立てて頂ければ，著者としては本望である．以下にやや長いが，画像を読み込んで小細工をして３つのファイルに出力するプログラムを示す．

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "file.h"
#include "okimari.h"
#include "area.h"
#include "mainlib.h"
/*
┌──────────────────────────────────────────────────────────────────────┐
│画像を読み込んでちょっと加工して，３つのファイルにコピーするプログラム．
│また，画素数を計算してファイルに出力するプログラム．
│ 2015/2/24
```

```
└──────────────────────────────────────────────────────────────────────────────
┘ */
int main_m;
/*****************************************************************************
int main(unsigned main_m){
FILE *fp1_g,/***画像の様々なデータ.dat***/
*fp2_g,/***出力画像１.pgm***/
*fp3_g,/***出力画像２.pgm***/
*fp4_g; /***出力画像３.pgm***/
unsigned i1_m, i2_m,
col_g,/***画像の行のサイズ***/
row_g,/***画像の列のサイズ***/
smpNo_g,/***サンプル数***/
kaicyo_g,/***画像の階調***/
L_g,/***画像の辺の長さ***/
N_g; /***画像の素子数***/
char *file_g,
*infile1_g,/***入力ファイル（画像）の名前***/
*infile2_g,/***入力ファイル（パラメータ）の名前***/
*outfile1_g,/***出力ファイル 1***/
*outfile2_g,/***出力ファイル 2***/
*outfile3_g,/***出力ファイル 3***/
*outfile4_g; /***出力ファイル 4***/
double **picR_g,/***入力画像ファイルデータ***/
**picW_g,/***出力画像ファイルデータ***/
***GazouIN_g,/***元の画像 GazouIN_g[j][サンプル数]***/
***GazouOUT1_g,/***加工後の画像１***/
***GazouOUT2_g,/***加工後の画像２***/
***GazouOUT3_g; /***加工後の画像３***/
/**********************文字列の領域確保*************************************
file_g=getC(50); infile1_g=getC(50); infile2_g=getC(50);
outfile1_g=getC(50); outfile2_g=getC(50); outfile3_g=getC(50);
outfile4_g=getC(50);
/**********************定数読み込み (サンプル数のみ)******************/
read1(file_g, infile1_g, &smpNo_g);
/**********************画像サイズ読み込み************************/
read_size(file_g, infile2_g, &col_g, &row_g, &kaicyo_g, &N_g);
L_g = col_g;
/**************書き込み用ファイルに名前を付ける (以下はファイル名の設定)*********
namefile(file_g, outfile1_g, "_data.dat"); fp1_g=fopen(outfile1_g,"w");
namefile(file_g, outfile2_g, "_out1.pgm"); fp2_g=fopen(outfile2_g,"w");
namefile(file_g, outfile3_g, "_out2.pgm"); fp3_g=fopen(outfile3_g,"w");
namefile(file_g, outfile4_g, "_out3.pgm"); fp4_g=fopen(outfile4_g,"w");
/**********************配列の領域確保 (以下は配列の確保)*************************/
```

```
picR_g =get2DW(L_g, L_g); /***画像を記憶する為の配列***/
picW_g =get2DW(L_g, L_g); /***画像を書き込む為の配列***/
GazouIN_g =get3DW(L_g, L_g, smpNo_g); /***スケーリングされた画像を保存***/
GazouOUT1_g =get3DW(L_g, L_g, smpNo_g); /***加工された画像を保存***/
GazouOUT2_g =get3DW(L_g, L_g, smpNo_g); /***加工された画像を保存***/
GazouOUT3_g =get3DW(L_g, L_g, smpNo_g); /***加工された画像を保存***/
/*********************************************************/
picR(infile2_g, col_g, row_g, picR_g);/***画像読み込み***/
rescale(picR_g, GazouIN_g, col_g, row_g);/***画像データを 0 中心にスケーリング（要
説明）***/
/*********************************************************/
/**↓↓↓↓↓↓↓ここからが一般的な実質的な作業領域↓↓↓↓↓↓↓↓↓↓↓**/
/*********************************************************/
/***底上げした画像を計算***/
for(i1_m=0; i1_m<col_g; i1_m++){
for(i2_m=0; i2_m<row_g; i2_m++){
GazouOUT1_g[i1_m][i2_m][0] = GazouIN_g[i1_m][i2_m][0]+1;/**画像を 1底上げして**/
GazouOUT2_g[i1_m][i2_m][0] = GazouIN_g[i1_m][i2_m][0]+2;/**画像を 2底上げして**/
GazouOUT3_g[i1_m][i2_m][0] = GazouIN_g[i1_m][i2_m][0]-2;/**画像を-2底上げして**/
}
}

N_g = 0;
for(i1_m=0; i1_m<col_g; i1_m++)/***素子数 N_g のカウント**/
for(i2_m=0; i2_m<row_g; i2_m++)
N_g++;

fprintf(fp1_g,"N_g = \t%u\n", N_g);

scale(picW_g, GazouOUT1_g, col_g, row_g);/***画像データを 127 中心にスケーリング***/
picW(outfile2_g, col_g, row_g, kaicyo_g, picW_g);/***加工画像をファイルに記述***/

scale(picW_g, GazouOUT2_g, col_g, row_g);/***画像データを 127 中心にスケーリング***/
picW(outfile3_g, col_g, row_g, kaicyo_g, picW_g);/***加工画像をファイルに記述***/

scale(picW_g, GazouOUT3_g, col_g, row_g);/***画像データを 127 中心にスケーリング***/
picW(outfile4_g, col_g, row_g, kaicyo_g, picW_g);/***加工画像をファイルに記述***/
/*********************************************************/
/**↑↑↑↑↑↑↑ここまでが作業領域↑↑↑↑↑↑↑↑↑↑↑**/
/*********************************************************/
/*********************************************************/
fclose(fp1_g);/**素子数の出力**/ fclose(fp2_g);/**image2.pgm**/
fclose(fp3_g);/**image3.pgm**/ fclose(fp4_g);/**image4.pgm**/
/*************************************************************************************
```

```
return(main_m);
}
```

3　プログラムの解説

3.1　領域確保

　上記のプログラムは

http://juntuzu.net

に掲載してある圧縮ファイルの中にあるので，解凍してそのままの構造を保ったまま main.c をコンパイル欲しい．
　ところで，

・char *file_g;
・file_g=getC(50);

とは著者が作成した関数で

・char file_g[50]

と同じ意味を持つ．なぜこのような面倒な事をするかと言うと，Windows のバージョンによっては，従来の方法ではトータルで１メガバイトまでしか領域を確保出来ない場合があるので，本稿では著者が作成した関数を用いて領域を確保する．
　他にも，

・double **picR_g;
・picR_g =get2DW(L_g, L_g);

も著者が作成した関数で本来なら

・double picR_g[L_g][L_g];

と書く場合と等価なものである．
　３次元配列の場合は，例として

・GazouIN_g =get3DW(L_g, L_g, smpNo_g);

と書く．これは

・double GazouIN_g[L_g][L_g][smpNo_g]

と書くのと等価である．

3.2　関数の説明

・read1(file_g, infile1_g, &smpNo_g)

この関数は，前もってパラメータを設定する場合に用いる関数である．本稿では特に理解する必要は無い．

・read_size(file_g, infile2_g, &col_g, &row_g, &kaicyo_g, &N_g);

この関数は画像ファイルのヘッダを読み込んで，ファイルの性質を知るのに必要な関数である．書式を書くと以下のようになる．

・read_size(ファイル名，ファイルのアドレス，行，列，階調，画像のピクセル数);

この関数を走らせると，画像の入力ファイル名を聞いてくるので，本稿では「lena.pgm」と入力すれば，lena.pgm ファイルの書式が読み込まれる．

・namefile(file_g, outfile1_g, "_data.dat");

このファイルは出力ファイルに名前を付けるファイルである．書式を書けば

・namefile(入力ファイル名，出力ファイル名，"出力ファイル名の語尾に付ける名前");

例えば，

file_g=lena.pgm

だったとすると

outfile1_g は lena_data.dat

となる．

picR(infile2_g, col_g, row_g, picR_g);

picR は入力ファイル infile2_g を 2 次元配列 picR_g に読み込む関数である．書式は

picR(入力ファイル名，行数，列数，入力配列);

である．

3.3　画像の階調

通常グレースケールの画像（白黒画像）では，各ピクセルが 0 から 255 までの値をとる．0 が白で，255 が黒である．しかし画像データを生のデータとしてプログラムで処理する場合には画素値の中心値が 0 である方が好ましい．したがって，本稿では各画素子から 127 だけ引き，20 で割っている．

rescale(picR_g, GazouIN_g, col_g, row_g);

上記の rescale では，picR_gに入っている生の画像データを 127 引き，さらに 20 で割っている．そのデータをGazouIN_gに保存している．この処理された状態で様々な計算を行い，最後にまた127を足して 20 を掛けるという作業をしている．下記の関数がGazouOUT3_g という処理済みのデータに 20 を掛けて 127 を足して picW_g としている．

```
scale(picW_g, GazouOUT3_g, col_g, row_g);
```

最後に

```
picW(outfile4_g, col_g, row_g, kaicyo_g, picW_g);
```

という関数では $picW\_g$ という画像データをファイルに出力している．
以上で，画像の読み書きをマスターしたことになる．

4 課題

・lena.pgm ファイルをコピーしてみよ．
・lena.pgm ファイルの輝度ヒストグラムをとって，gnuplot で表示して見よ．
・次に lena 画像を左右反転，上下反転，原点反転してみよ．
・さらに lena 画像を 1/2 の大きさに変換せよ．
・次に２倍に拡大せよ．
・一様ノイズを入れるプログラムを作成せよ．
・一様ノイズを修復するプログラムを作成せよ．
・その他随時課題を与える

5 参考

word 等を用いて報告書を書くにあたって，目的の Window 画面を Word に貼り付ける方法について説明する．まず，目的の Window をアクティブにした状態で,「Ctrl」＋「Alt」＋「Print Screen」を押すことでクリップボードに保存される．この状態で Word の画面に戻り,「Ctrl」＋「v」を押すことで貼り付けることができる．