

「画像・音声情報処理」中間レポート1

平成28年度の本講義では、計4回の「中間レポート」を課します。今回は1回目です。



■ レポートの目的と内容

➤ 目的：階調画像にヒストグラム平滑化(階調補正)を行なってその効果を確認する。

➤ 内容：

1. **原画像について**：本講義「画像・音声情報処理」の web サイトである <http://nagao-lab.ynu.ac.jp/lecture/ipap.html> に右の画像(org1.pgm)を含む参考ファイルがありますのでお使い下さい。また、自分で何かお好きな画像をフリーソフトの IrfanView を使って、モノクロ階調画像(pgm 形式、バイナリタイプ)で保存して原画像 org2.pgm を作って下さい。カラー画像を IrfanView で表示してから「画像」メニューの「グレースケールに変換」を実行してモノクロ階調画像に変換します。次に「ファイル」メニューの「名前をつけて画像を保存」で、「ファイルの種類」の中の「PGM-Portable Graymap」を選ぶと開く「PNG/PNM/ICO 保存設定」ウィンドウの「圧縮レベル」を 0(圧縮なし)にし、「バイナリデータとして符号化」だけにチェックを入れて保存します。なお、pgmlib.h では 1280×1280 画素以下がデフォルトになっています。それよりも大きなサイズの画像を扱うときは pgmlib.h の中の宣言部分を修正して下さい。
2. **ヒストグラム平滑化のプログラム(heikatsu.c)の作成**：参考テキスト「C 言語による画像処理プログラミング入門」の無償ダウンロードプログラムには含まれていません。裏面の補足説明および雛型のプログラムを参考にして自分で作って下さい。結果的に完成しなくても(つまり、書き掛けでも)、努力点をあげますので、自分だけの力で取り組みましょう。そうしないと自分の勉強になりません。なお、方法について自分だけではどうしてもわからないときは、周囲の人と相談して“アルゴリズム(処理方法)”を確認しても良いですが、他人のプログラムをそのままコピーすることは禁止です(双方を0点にします)。
3. **ヒストグラム平滑化のプログラムの実行**：heikatsu.c をコンパイル・実行して、org1.pgm のヒストグラム平滑化を行った処理画像 out1.pgm と、自分で決めた原画像 org2.pgm とその処理画像 out2.pgm を作りましょう。そして、IrfanView で画面に表示して、どのような画像になったか確認しましょう。
4. **レポートの提出**：レポートは電子メールでお願いします。ソースプログラム(heikatsu.c)、org2.pgm、out1.pgm、out2.pgm を添付して下さい(ファイルを1つのフォルダに入れて zip などでも圧縮して1つのファイルにして添付しても結構です)。電子メールについては次のようにして下さい。



宛先 to: nagao@ynu.ac.jp	← 担当教員(長尾)のメールアドレス
Cc: 自分のメールアドレス	← 確認のため入れて下さい
件名: report1(1444*** 横浜太郎)	← ()の中に自分の学籍番号と名前を入れて下さい
内容:	
学籍番号: 1444***	← 自分の学籍番号
氏名: 横浜太郎	← 自分の氏名
レポートについて:	
1) ソースプログラムについて:	← 数行以内で簡単に説明して下さい。
2) 実験結果について:	← 数行以内で簡単に説明して下さい。
3) 感想・意見など:	← 自由記述。

レポート提出期限：2016年 11月15日(火) 深夜24:00 (=16日の00:00)

※レポートが提出されたら、長尾から「受領通知メール」を返信します(概ね提出後1~2日以内)。レポートを提出したのに「受領通知メール」が届かなかった場合は早めにご連絡下さい。問い合わせが遅い場合は「未提出」

扱いになる場合がありますのでご注意ください。

■ 補足説明（ヒストグラム平滑化）

- 原理：図1に示すように、原画像の濃度ヒストグラム（画素の階調値の頻度分布）を引き伸ばし、分布も揃えて右側のように変換することで、原画像の階調を補正する処理。なお、図1の h は、原画像の総画素数（＝横画素数 × 縦画素数）／256（＝ヒストグラムの横幅）から決定できる。

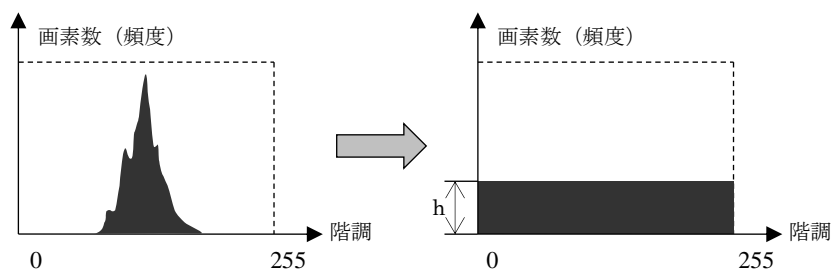


図1 ヒストグラム平滑化処理

- 方法：手順を次に示す。

1. 原画像の最小階調値 f_{min} と最大階調値 f_{max} を求める。
2. for ループで制御変数 i を、 $i=f_{min}$ から f_{max} まで動かしながら、各画素の階調値が変換後にどの階調値になるかを決定する。すなわち、新しい階調値を g とすると、 $g=0$ から始めて、累積が h になったら、 $g=g+1$ として1ずつ増やしていく。つまり、図1の右側の棒グラフを、階調 $g=0$ の一番左側の棒から順に（右側へ）埋めていく。これによって変換後の画像を作ることができる。

- 処理例：このプログラムの処理例を次に示す。



- ヒント：

- ① 上の方法 2. の処理は意外に簡単に書くことができます。ヒントになっていないですが...
- ② main 関数中に処理を書き込めばよく、自分で新たに関数を作らなくても大丈夫です。
- ③ 雛型のサンプルプログラム（次頁、ウェブにアップロード済み）を元にして作ると楽です。
- ④ もしよくわからなければ、情報工学 EP または電子情報システム EP の人に相談しましょう。

レポートについて何か不明な点や相談したいことがあれば、

- 電子メール：nagao@ynu.ac.jp
- 電話：045-339-4131（学内からの内線電話の場合は 4131）
- 長尾教授居室：総合研究棟 S 棟 4 階 S401 室（秘書は S402 室）
- オフィスアワー：毎週火曜日～金曜日の 12:00～13:00（会議などを除く）

まで、お気軽にご相談下さい。ただし、自作したプログラムのバグ（誤り）の修正方法についてのご質問にはお答えできません。



注意！ : org2.pgm としてタレントその他の画像を使う場合は著作権・著作権その他の権利を侵害することがないようにして下さい。例えば、画像処理後の画像を自分のブログに掲載する行為等も違法になります。
あくまでも本レポートのためだけに使用し、外部には絶対に出さないようお願いします。

```
// heikatsu.c このプログラムの名前
// *****
// これは「不完全な」プログラムです. このプログラムの
// ①, ②, ③の部分に適切な記述を補って, 正しく動作す
// るプログラムに変えてから提出して下さい.
// *****
#include<stdio.h>
#include"pgmlib.h"

int main(void)
{
    int h;                // 変換後のヒストグラムの高さ
    int fmin = 255, fmax = 0; // 画像の最低階調値・最大階調値
    int x,y;              // 制御変数
    int g=0;              // 変換後の初期階調値
    int count;            // 各階調 g の画素の数
    int f;                // ③で階調についての for ループを
                        // 作るときの制御変数

    printf("==== ヒストグラム平滑化のプログラム ===¥n");
    printf("原画像を読み込みます¥n");
    load_image( 0, "" );    // ファイル → 画像 No.0   へ読み込み
    copy_image( 0, 1 );    // 画像 No.0 → 画像 No.1   へコピー

    // h (=平滑化後のヒストグラムの高さ) を求める
    // *****
    //
    // ①この部分に適切な記述を書きましょう
    //
    // *****
    printf("ヒストグラムの高さ = %d¥n",h);

    // image[0] の最小階調値 fmin, 最大階調値 fmax を求める
    // *****
    //
    // ②この部分に適切な記述を書きましょう
    //
    // *****
    printf("原画像の最小階調値=%d, 最大階調値=%d¥n",fmin,fmax);

    // 変換する
    // 原画像の最小階調値から最大階調値までを走査しながら,
    // 変換後の画像の階調値を g=0 から順に g=255 まで埋めていく
    // count は, 各 g での画素数のカウンタで, count=h になったら,
    // g の値に 1 を加える.
    // *****
    //
    // ③この部分に適切な記述を書きましょう
    //
    // *****
    printf("ヒストグラム平滑化が終了しました¥n");

    printf("処理後の画像を保存します¥n");
    save_image( 1, "" ); // 画像 No.1 → ファイル
    return 0;
}
```