理工学部 数物・電子情報系学科 情報工学 EP
秋学期講義「画像・音声情報処理」
2018年11月6日(火)

担当：長尾 智晴
総合研究棟 S401 室(内線 4131)
nagao@ynu.ac.jp

「画像・音声情報処理」中間レポート２

平成 30 年度の本講義では，計4回の「中間レポート」を課します．今回が2回目です．

■ レポートの目的と内容

- 目的：簡単なカラーシミュレータの作成．
- 内容：

1. 原画像について：　例えば右に示すように，洋服の色が割とはっきり写ったカラー画像を用意して，フリーソフトの IrfanView を使って ppm 形式(バイナリ)で保存して原画像(org.ppm)にして下さい．

図１　原画像(org.ppm)の例

※ この画像は肖像権・著作権・版権などで保護されています．今回は教育目的のため無断で拝借していますが，本来は使用に当たって許可を取る必要があります．

2. 色変更のプログラム(colorsim.c)の作成：　裏面の手順に従って原画像中の人の洋服の色を様々に変えるプログラムを作って下さい．なお，画像入出力用ヘッダ ppmlib.h とラベリング用ヘッダ label.h を利用すること．結果的に完成しなくても(つまり，書き掛けでも)，努力点をあげるつもりですので，自分だけの力で取り組みましょう．さも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」を確認しても良いですが，他人のプログラムをそっくりそのままコピーすることはやめて下さい．

3. 色変更のプログラムの実行：　colorsim.c をコンパイル・実行して，背景を白，対象色の候補領域を青色で表した画像(mask1.ppm)と，その領域の中で大きさや形などの条件を満たす領域だけをさらに絞り込んで青色にした画像(mask2.ppm)を保存します．続いて，原画像のその領域の画素だけを色変更した画像(out1.ppm，out2.ppm，… など全部で1枚以上)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(colorsim.c)，org.ppm，mask1.ppm，mask2.ppm，out1.ppm，out2.ppm，・・・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 などで圧縮して1つのファイルにして添付しても OK)．電子メールは次のようにして下さい．

宛先 to：nagao@ynu.ac.jp	← 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：report2(16***** 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：16*****	← 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：	← 数行以内程度で簡単に説明して下さい．
2)実験結果について：	← 数行以内程度で簡単に説明して下さい．
3)感想・意見など：	← この課題や講義についてなど，自由記述．

レポート提出期限：２０１８年　１１月２７日(火)　深夜２４：００　(＝２８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(たまにスパムメール扱いになっていることがあります)．また，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下さい．

■ 処理手順

以下に処理手順を示します．

1）処理1：色変更の候補領域の抽出

原画像 org.ppm 中の色を変えたい領域の画素を青（Red=0, Green=0, Blue=255），そうではない画素を白（Red=Green=Blue=255）にした画像を作って mask1.ppm で保存します．図2は図1の原画像で次の条件を満たす画素を青にしたものです：

Red > 1.1*Green　&&　Blue > 1.1*Green　&&　Red > 180

条件式は原画像とその中の色変更したい領域の色によって異なります．ご自分で原画像にした画像に合わせて条件を設定して下さい．上の式はあくまでも参考です．もっと単純あるいは複雑にしないとうまく抽出できないこともあります．

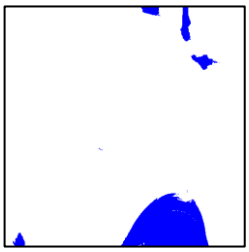

図2　候補領域1
（mask1.ppm）

2）処理2：ラベリング

mask1.ppm 中の背景(白)ではない図形領域に対するラベリングを実行します．

3）処理3：面積で候補を絞り込む

mask1.ppm の青い図形領域には，対象領域以外の似た色の領域が含まれている可能性があります．そこで，さらに候補を絞り込むために，面積が一定範囲内のものだけを残した画像を作って mask2.ppm で保存します．図2中の青い図形の中で面積が画像全体の 1%以上 30%以下のものだけを残した結果を図3に示します．

図3　候補領域2
（mask2.ppm）

4）処理4：対象領域の色を変更して最終的な出力結果を求める

最後に，mask2.ppm の青い画素に対応する原画像中の画素の色を変更し，最終的な出力結果として保存します．名前は out1.ppm，out2.ppm，…などとすること．ここでは簡単に，Red，Green，Blue の階調値に一定のシフト量（例えば−100～+100）を加えるものとし，シフト量は毎回キーボードから入力することにします．なお，シフト後の値が 0 未満になるときは 0，256 以上になるときは 255 にすることをお忘れなく．これによって，例えば図4に例示するように服の色を変化させた画像を作ることができます．

R:100, G:−100, B:−100
（out1.ppm）

R:=−100, G:100, B:−100
（out2.ppm）

R:=−100, G:=−100, B:100
（out3.ppm）

原画像（再出）
（org.ppm）

図4　服の色を変化させた画像の例

5）応用問題：画像中の色指定を RGB 値ではなく HSV 値を用いる

さらに挑戦したい人は，colorsim.c の画像中の色変更を，RGB 値ではなく RGB 値から計算できる HSV 値（H：hue（色相），S:saturation（彩度），V:value（明度））に変更したプログラム（colorsim2.c）を作り，結果について考察して下さい．RGB 値から HSV 値への変更方法はインターネットなどで調べて下さい．

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，お気軽に
- 電子メール：nagao@ynu.ac.jp
- 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）

までご相談下さい．

注意！：実験にタレントさんその他の画像を使う場合は著作権・版権その他の権利を侵害することがないように注意して下さい．例えば，実験に使った画像を勝手に自分のブログなど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．

秋学期講義「画像・音声情報処理」第２回レポート　回答の雛型（サンプル）のプログラム（colorsim.c）
※本講義のウェブページの第2回レポートの関係ファイルに含まれています．

```
// colorsim.c （このプログラムの名前）
#include<stdio.h>   // 標準入出力ヘッダのインクルード
#include"ppmlib.h"  // ppm ファイル用ヘッダのインクルード
#include"label.h"   // ラベリング用ヘッダのインクルード

#define RMIN 0.01   // 領域面積の下限(=1%)
#define RMAX 0.3    // 領域面積の上限(=30%)

int main(void)
{
    int x,y;      // 座標用の制御変数
    int number;   // 孤立領域の総数
    int r,g,b;    // RGB 値
    int i;        // ラベル用の制御変数
    int area;     // 領域の面積
    int count;    // 最終的に残った領域の総数
    int xmin,ymin,xmax,ymax;  // ラベルの領域
    double ratio;  // 面積率
    int color;     // 作業変数

    printf("==== カラーシミュレーター =====¥n");
    printf("原画像(ppm(バイナリ)形式)を読み込みます¥n");
    load_color_image(0,"");    // ファイル → 画像 No.0 への読み込み
    copy_color_image(0,1);     // 画像 No.0 → 画像 No.1 へコピー

    // 処理１：画像 No.1 の目標の色の画素を青色で抽出します
    printf("¥n 処理１：目標の色の画素を青色で抽出します．¥n");
    // ************************************************************************
    // ここに，画像 No.1 の画素の色が，原画像(No.0)の色変更対象領域なら青（R=
    // G=0,B=255），そうでないなら白（R=G=B=255）にする処理を書いて下さい．
    // 条件式は対象とする原画像に依存します．つまり，ここでの選別は，自分で
    // 選んだ原画像だけに有効で，他の画像には使えないもので良いです．
    // ************************************************************************
    printf("原画像の目標色の画素を青色にした画像を mask1.ppm で保存します．¥n");
    save_color_image( 1, "mask1.ppm" );

    // 処理２：ラベリング
    printf("¥n 処理２：目標色の領域のラベリングを行ないます．¥n");
    printf("ラベリング中です．個数が多いと時間がかかります...¥n");
    // ************************************************************************
    // 関数 labeling( ) を使って，画像 No.1 を背景画素の色を白（R=G=B=255)と
    // してラベリングを行ない，孤立領域の総数を変数 number に代入する記述を
    // 補って下さい．たった１行だけです．
    // ************************************************************************
    printf("終了しました．孤立領域の総数＝%d 個¥n",number);
    if ( number == 0 ){
        printf("該当する領域がないので終了します．¥n");
        exit(1);
    }

    // 処理３：面積で選別する
    printf("¥n 処理３：目標色の領域の面積で選別します．¥n");
    printf("面積が範囲外のものは白画素にします.¥n");
    count = number;
    // ラベル=1 から number までを調べる
    for(i=1;i<=number;i++){
        // ************************************************************************
        // まず，ラベル i の領域の面積 area を求めましょう．
        // その際，同時にその領域の x, y の範囲（xmin,ymin)<--->(xmax,ymax)
```

内部でラベル用配列 label[ ][ ]を大域変数として宣言している．

★カラー画像は次に示す4次元配列として扱う．
image[0～2][x][y][0～2]
↑　↑　↑　↑
画像 No.　座標　色(0:R, 1:G, 2:B)
例）画像 No.0 の座標(10,20)の Blue →
image[0][10][20][2]

以下の関数は ppmlib.h 内で宣言されている．
load_color_image( )
copy_color_image( )
init_color_image( )
save_color_image( )

ラスタ走査して調べるのは画像ではなく，label[ ][ ]です．

```
            // も求めておきましょう（後の走査を高速化するためです）．ここで，
            // (xmin,ymin)：ラベル i の領域の左上の座標
            // (xmax,ymax)：ラベル i の領域の右下の座標
            // 次に面積率 ratio を求めましょう．ここでの面積率とは，area を画像
            // 全体の面積で割ったものです．すなわち，0.0～1.0 の実数値になります．
            // 次にその ratio の値の範囲（プログラムの冒頭の define 文で定義され
            // ています．この例では 1%以上 30%以下です）ではない画素を白にします．
            // この処理は既に次に書かれていますので作らなくて結構です．
            // ********************************************************************
            if ( ratio < RMIN || ratio > RMAX ){
                // 面積が該当しない場合は青画素を白画素へ変更する
                count--;
                for(y=ymin;y<=ymax;y++){
                    for(x=xmin;x<=xmax;x++){
                        if ( label[x][y] == i ){
                            // 白画素へ変更する
                            image[1][x][y][0] = 255;
                            image[1][x][y][1] = 255;
                            image[1][x][y][2] = 255;
                        }
                    }
                }
            }
        }
        printf("終了しました．該当した孤立領域の総数＝%d 個¥n",count);
        if ( count > 0 ){
            printf("該当領域を青色にした画像を mask2.ppm で保存します．¥n");
            save_color_image( 1, "mask2.ppm" );
        } else {
            printf("該当する領域が残らなかったので終了します．¥n");
            exit(1);
        }

        // 処理４：色の変更
        printf("¥n 処理４：該当領域の色を変更します．¥n");
        printf("Red   のシフト量 [-100,100] = "); scanf("%d",&r);
        printf("Green のシフト量 [-100,100] = "); scanf("%d",&g);
        printf("Blue  のシフト量 [-100,100] = "); scanf("%d",&b);
        // ********************************************************************
        // 原画像（No.0）をラスタ走査し，対応する No.1 の画素が青のときだけ，No.0
        // の画素の色に上記のシフト量を加えます．0 より小さくなってしまったら 0，
        // 255 より大きくなったら 255 にすることをお忘れなく．
        // これによって，最終的な出力画像が No.0 に作られます．
        // ********************************************************************
        printf("色を変更した画像を保存します．¥n");
        save_color_image( 0,"");

        return 0;  // 正常終了を示す値をシステムに返します
}
```

補足説明:

- このサンプルプログラムの // でコメントアウトされている部分の処理（4箇所）に適切な記述を補うことでプログラムを完成させましょう．
- なお，プログラミングに慣れている人は，このサンプルプログラムを用いずに自分でゼロから作っても結構です．
- 情報工学 EP と電子情報システム EP 以外の EP の人で，どうしても分からない人は私（nagao@ynu.ac.jp）までメールで相談して下さい．場合によってはさらにヒントをお教えします．