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 27 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３

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3回目です．

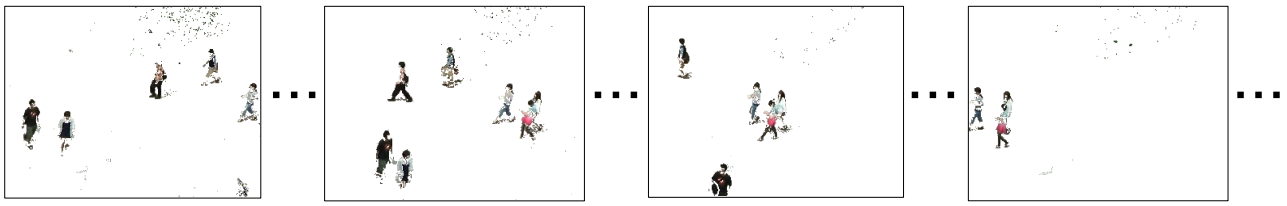
■ レポートの目的と内容

- 目的：カラー動画像に対する処理の作成．
- 内容：
  - 原画像について：
    以前にキャンパス内で撮影した短い動画像を ppm フォーマットの連像静止画像列 org00001.ppm～org00200.ppm にしました．また，このシーンの背景画像 back.ppm を用意しました．今回はこれらを原画像(列)とします．画像のサイズは，いずれも横 320 画素×縦 240 画素です．

背景画像(back.ppm)　org00001.ppm　org00060.ppm　org00120.ppm　org00180.ppm

図１　本レポートにおける原画像

  - 必須問題１：人物領域抽出プログラム１(human1.c)の作成と実行：
    各フレームと背景画像の差分を求め，Red または Green または Blue が 30 以上の階調差がある画素を移動領域とし，それ以外の画素を白にした連続画像 out00001.ppm～out00200.ppm を出力するプログラム human1.c を作って下さい．連番のファイル名の作成方法については，次頁の“必須問題のヒント”を参考にすること．レポートにはソースプログラム human1.c だけを添付すれば良いです．このプログラムを実行すると，全部で 200 枚のファイルが現在のフォルダに生成されるのでご注意して下さい．



out00001.ppm　out00060.ppm　out00120.ppm　out00180.ppm

図２　human1.c によって保存される移動領域だけを抽出した連続画像(全部で 200 枚)の例

  - 必須問題２：人物領域抽出プログラム２(human2.c)の作成と実行：
    必須問題１のプログラム human1.c を少しだけ改良して，図3に示すように 20 フレーム毎(フレーム No.1，No.21，No.41,…，No.181)の人物領域を背景画像に重ねて表示した合成画像 human.ppm を作るプログラム human2.c を作って下さい．レポートにはソースプログラム human2.c と出力画像 human.ppm の両方を添付して下さい．この際の色の差分のしきい値も必須問題１と同様に 30 としましょう．もし余裕があれば，“現在から過去にさかのぼるほど透明度を上げて 1 フレーム毎の人物領域を重ねる”ことにも挑戦してみると良いでしょう(方法を考察するだけでも結構です)．

図３　出力画像 human.ppm

  - 応用問題(やらなくても良い)：特殊映像の画像列を作るプログラム(effect.c)の作成：
    処理の内容はどのようなものでも構いません．自分で撮影した短い動画像から静止画像列を(本資料 p.3 の補足説明を参考にして)作り，それらを原画像列として何等かの処理を加えて静止画像列を出力するプログラム effect.c を作って下さい．effect.c と出力された画像の例を数枚程度メールに添付して説明して下さい．静止画像列は IrfanView で連続して表示することで動画のように表示できます．

● レポートの提出:

レポートは電子メールで提出すること. 必須問題1・必須問題2の報告に必要なファイルを1つのフォルダに入れて zip などで圧縮して添付しても結構です. 応用問題も出来た人はその関連ファイルも添付して下さい. 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:report3(1644*** 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: 1644***	← 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:	← それぞれ数行以内で簡単に説明して下さい.
2)実験結果について:	← それぞれ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(できれば何か書いて下さい).

レポート提出期限: 2018年12月18日(火) 深夜24:00 (=19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(ごくまれにスパムメール扱いになっていることがあります). 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.

■ 必須問題のヒント

必須問題1で行う処理は簡単ですが, プログラム中でファイル名を org00001.ppm～org00200.ppm まで連続的に変えながら画像を読み込み, 処理して out00001.ppm～out00200.ppm で保存する必要があります. このようなことは, 標準ヘッダファイルの string.h 中の文字列操作関数を用いて行うことができます. そこで連番のファイル名を作って返す関数として次の make_filename()を作りました. ファイル名 makefname.c で保存されています.

```
void make_filename( char head[], int keta, int num, char fname[] )
// org00100.ppm などの連続的なファイル名を作成する関数
// head[]:"in","out"などの最初の文字列
// keta:桁数. 上の例では 5
// num:番号(0,1,2,...). 上の例では 100
// fname[]:最終的なファイル名. 上の例では"org00100.ppm"
{
    char buffer[20];   // 作業変数
    int i,length;      // 作業変数

    strcpy( fname, head);       // fname に最初の文字列を代入
    sprintf( buffer,"%d",num ); // num を 10 進数の文字にして buffer へ
    length=strlen(buffer);      // buffer の文字数を調べる
    for(i=1;i<=keta-length;i++){
        strcat( fname, "0" ); // 必要なだけ 0 を追加
    }
    strcat( fname, buffer );   // 数字部分を追加
    strcat( fname, ".ppm");    // 最終的なファイル名
}
```

太字は文字列操作関数

この関数は例えば次のように用います. 1 番目の引数が頭につける文字列, 2 番目が桁数, 3 番目が番号, 4 番目がファイル名を保存する文字型配列です.

【使用例】

```
char   fname[256];

make_filename("org",     5, 100, fname);  → fname に"org00100.ppm" が代入されます.
make_filename("",        3, 100, fname);  → fname に"100.ppm"      が代入されます.
make_filename("out",     4,  23, fname);  → fname に"out0023.ppm"  が代入されます.
make_filename("image", 2,  10, fname);  → fname に"image10.ppm"  が代入されます.
```

この関数を使ったサンプルプログラム（sample.c）（レポート関連ファイル中にあります）を次に示します．このプログラムを変えることで必須問題1・2を作ると良いでしょう．ちなみにこのプログラムは，同じフォルダにある連番ファイル org00001.pm～org00010.ppm を読み込んで，色を反転させた画像を out00001.ppm～out00010.ppm として保存するプログラムです．なお，ppmlib.h の文字表示部をコメントアウトして削除した ppmlib2.h（レポート関連ファイル中にあります）を用いています．

```
// sample.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"ppmlib2.h"        // 画面表示部をカットした ppmlib.h
#include"makefname.c"      // 連番ファイル名の作成

//関数のプロトタイプ宣言（makefname.c 中の関数）
void make_filename( char head[], int keta, int num, char fname[] );

// main 関数の始まり
int main(void)
{
    int i;                 // i:入力ファイル番号
    char fname[256];       // ファイル名用配列（入出力兼用）
    int x,y,col;           // 制御変数

    printf("原画像のフレーム org00001.ppm～org00010.ppm を反転させます¥n");
    for(i=1;i<=10;i++){
        // 連番のファイル名を生成（org00001.ppm～org00010.ppm）
        make_filename( "org", 5, i, fname );    // 5:数字の桁数
        // 画像を No.0 に読み込む
        load_color_image( 0, fname );
        // 画像を反転させる
        for(y=0;y<height[0];y++){
            for(x=0;x<width[0];x++){
                for(col=0;col<3;col++){
                    image[0][x][y][col] = 255 - image[0][x][y][col];
                }
            }
        }
        // 出力ファイル名（out00001.ppm～out00010.ppm）を作る
        make_filename( "out", 5, i, fname );
        // 画像の保存
        save_color_image( 0, fname );
    }
    printf("結果画像を out00001.ppm～out00010.ppm で出力しました¥n");
    return 0;   // プログラムの正常終了
}
```

注意！

makefname.c 中の関数

ppmlib2.h 中の関数

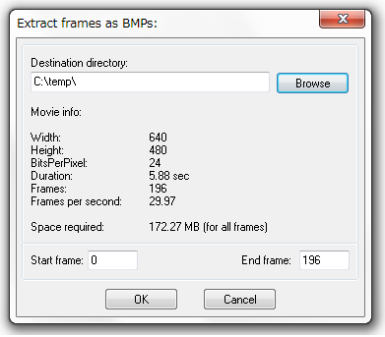
画像 No.0 のカラー画像の色を反転させている

makefname.c 中の関数

ppmlib2.h 中の関数

補足説明 フリーソフト IrfanView を使って動画像ファイル(wmv)を連続 ppm 画像に直す方法

1. 最新版(ver.4.38)をインストールする（すべて英語です）
   （古いバージョンではまだこの機能がサポートされていません．）
2. 元の動画像ファイル（wmv など）をファイルから読み込む．
3. Options の中の Extract All Frames を選ぶ（右図）．
4. 出力先フォルダ，開始・終了フレーム番号を指定すると，それらのフレームが全て bmp 画像列に変換して保存される．
5. Files の Batch Conversion/Rename で，bmp 画像列を ppm 画像列に変換する．その際，連番の指定なども可能．

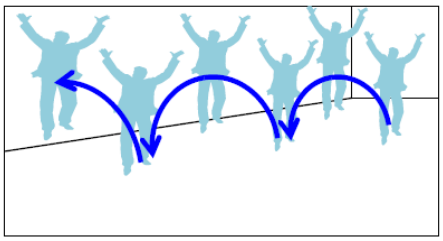


■ 応用問題の補足説明

- 処理内容は自分で考えて下さい．どのような処理でも結構です．できればユニークで面白いものを期待します．元の動画像は何でも結構です．

- どのような処理でも構わないのですが，考えるヒントとして，いくつかの例をご紹介しましょう．

  - 動画像中のフレームを間引く：　例えば，一定間隔で，一定の枚数のフレームを削除するだけでも，時間が飛び飛びになったような面白い効果の動画像を作成することができます．例えば，右図に示すように一定周期でジャンプした動画を撮影し，着地する前後のフレームを間引くと，空中を浮遊しているかのような不思議な動画を作ることができます．



  - 画像の上半分あるいは下半分を折り返した画像にする：　下図のような動画を作ることができます．


YouTube “Tokyo Sky Drive 01”より （これらの画像は著作権等で保護されています）

  - 各フレームの色数を減じる：　各フレームのカラー画像の階調数を減じることで，ポップな感じ？あるいはアニメ風？の動画を作ることができます．本講義で解説したように，R，G，B をそれぞれ単純に一定値で割って新しい階調値にしてからその階調を表現する 256 階調の値を割り振る方法が最も簡単です．例えば R を4階調にしたければ，R = (R / 64) * 64; とすれば良いですね（最初の括弧内は整数割算で「切り捨て」になるので 0〜3 の値になります）．以下同様です．R，G，B を各4階調にした場合は，原画像を4×4×4＝64色数で表現したことになります（なお，原画像の色数は 256^3 色です[1]）．

以上です．これらにこだわらずに自由に発想して下さい．

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
- 電話：045-339-4131（学内からの内線電話の場合は内線 4131）
- 居室：総合研究棟 S 棟 4 階 S401 室（秘書（石井・黒坂）は隣の 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．ただし，プログラムのデバッグに関するご相談には応じかねますのでご注意下さい．

注意！：実験にネット上の動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，実験で作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．

[1] 表示（表現）可能色数のことであり，その画像中に含まれている色の数のことではないことに注意．