3年15組９２番　全　海成(JEON HAESUNG)
コンピュータサイエンス実習A　マルチメディア　三回目　レポート

今回の実習では以前の二回の実習と違う画像処理ライブラリーを使いました。以前はlibgdというCのライブラリーを使いましたが今回はC++のOPENCVと言うライブラリーを使いました。このOPENCVはすごく多いバージョンがあって僕が提出する実行形式をOPENCV3.4.0を使ったコンパイルしました。そのため実行するコンピュータのOPENCVのバージョンと異なる場合実行できないことがあります。そのため実行環境に当てて再度コンパイルをする必要があります。使うコンパイラーはg++で以下のようなコマンドを使って再度コンパイルしてください。

g++ -o [Execute File] [Source File] $(pkg-config opencv --cflags --libs)

必須課題１ではlibgdも使うので最後の部分に-lgdを追加してください。

必須課題１

この課題は以前まで使ったlibgdを利用して画像ファイルを開きOPENCVを使って保存するプログラムを書くことです。libgdとOPENCVで一番違うところはlibgdでは必要な情報を構造体で持ってくるとき先に定義されていたマクロを使いましたがOPENCVではクラスで管理されておりメソッドを使ってアクセスします。以下は実行コマンドの形式です。

./ass1 [Input Image] [Output Filename]

出力画像が入力画像と同じため実行画面でプログラムがうまく働くことを見せます。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ ls
add1       ass1       ass2       ass3       ass4       ass5       at_test.cpp  lena.jpg  string_test.cpp  test.cpp       video_test   video_test.cpp
add1.cpp   ass1.cpp   ass2.cpp   ass3.cpp   ass4.cpp   ass5.cpp   cvlena.jpg   src.JPG   test             time_test.JPG  video_test2
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ ./ass1 lena.jpg dst1.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ identify lena.jpg
lena.jpg JPEG 225x225 225x225+0+0 8-bit sRGB 7.64KB 0.000u 0:00.000
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ identify dst1.JPG
dst1.JPG JPEG 225x225 225x225+0+0 8-bit sRGB 16.9KB 0.000u 0:00.000
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ display dst1.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$
```

任意課題１

OPENCVで画素値にアクセスするためにはatメソッドを使います。しかしこのメソッドはかな時間がかかる方法です。ポインタを利用するとより高速にアクセスができます。この課題ではatを利用したアクセスとポインタを利用したアクセスを両方を実行して比べます。

実験方式は一つの画像ファイルを入力で受けてそれをOPENCVを使って読み込み両方の方法を使って全部の画素に対してアクセスします。それぞれの方法を百回行ってその平均時間を出力します。まず各方法をコードでみせますと以下のようです。

```
for(y=0;y<height;y++) {
  for(x=0;x<width;x++) {
    Vec3b bgr = m.at<Vec3b>(y,x);
  }
}
```

```
for(y=0;y<height;y++) {
  Vec3b *p = m.ptr<Vec3b>(y);
  for(x=0;x<width;x++) {
    Vec3b bgr = p[x];
  }
}
```

下はプログラム実行結果です。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ identify lena.jpg
lena.jpg JPEG 225x225 225x225+0+0 8-bit sRGB 7.64KB 0.000u 0:00.000
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ ./add1 lena.jpg
0.000558 sec when using at
0.000441 sec when using pointer
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ identify time_test.JPG
time_test.JPG JPEG 3264x2448 3264x2448+0+0 8-bit sRGB 5.966MB 0.170u 0:00.179
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$ ./add1 time_test.JPG
0.080631 sec when using at
0.068726 sec when using pointer
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week3$
```

小さい画像と大きい画像に対してプログラムをそれぞれ実行しました。lena.jpg に対しては約 20%ほどポインタを使った方が早いです。time_test.jpg に対しては約 15%ほどポインタを使った方が早いです。

この違いは安全性のためのチェックをするかしないかからくるものだと予想できます。at メソッドはアクセスしようとする座標に対して範囲外なのかチェックしますがポインタを使った方法はそれをしません。

必須課題２

必須課題２は OPENCV を使って何らかの空間フィルタを画像にかけるプログラムを書くことです。先週の課題でフィルタを直接作って適用しました。OPENCV は色んな種類のフィルタを関数として提供するので直接作る必要はありません。ただコード一行でフィルタをかけることができます。この課題では OPENCV が提供するフィルタの中でバイラテラルフィルタを入力画像に適用します。以下は実行コマンドの形式です。

./ass2 [Input Image] [Output Image]

処理前と処理後の画像です。

ノイズは減ったんですが少しボヤけました

必須課題３

この課題は OPENCV を利用してアフィン変換あるいは射影変換を実行することです。僕は画像を横に鏡映するアフィン変換を実行しました。アフィン変換の時使う行列は以下のようです。

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

この行列で a,b,c,d は先週やった線形変換で t は平行移動を指定します。以下は実行コマンドの形式と結果です。

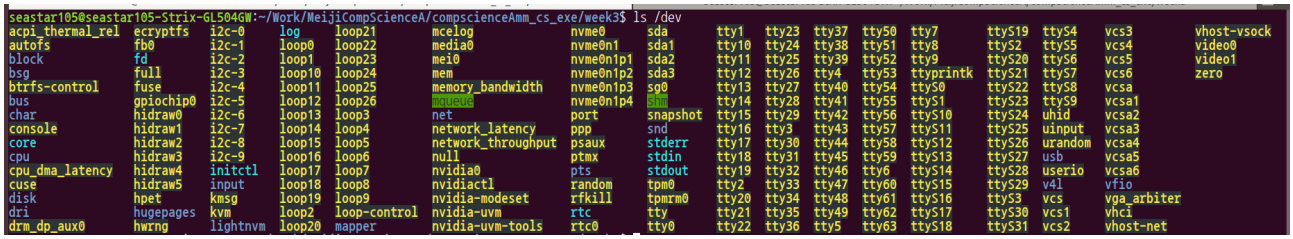
./ass3 [Input File] [Output File]

必須課題４

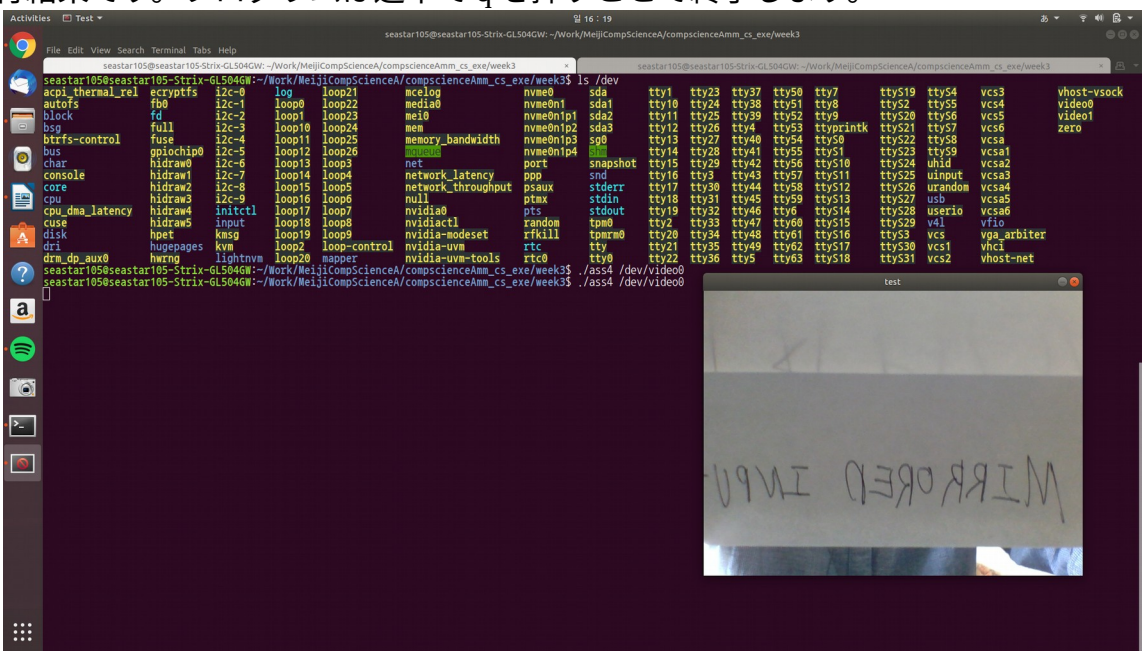
この課題ではカメラから入力をとって何らかの画像処理を行って出力をするプログラムを書くことです。僕が書いたプログラムは必須課題３で実行した画像処理をカメラからの入力画像に対して実行した後出力します。以下は実行コマンドの形式です。

./ass4 [Camera device]

ここで Camera device はつかえるカメラの位置を言います。



上の図を見るとつながってる入出力装置を見えますが video0 と video1 がカメラ装置です。だからプログラムの入力として *‘/dev/video0’* を渡さなきゃちゃんと作動しません。以下は実行結果です。プログラムは途中で q を押すことで終了します。



出力画像で文字を見ると左右反転処理を実行したのを確認できます。

必須課題５
必須課題５はカメラからの入力に何らかの文字列を出力して出すことです。
プログラムの使用方法は４と同じです。
以下は実行結果です。出力する文字列は JHS です。