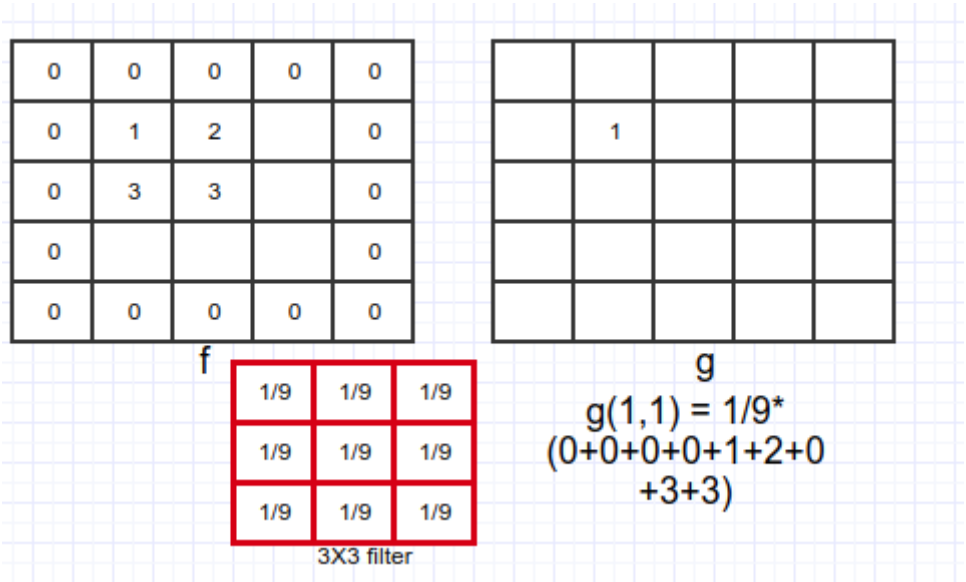
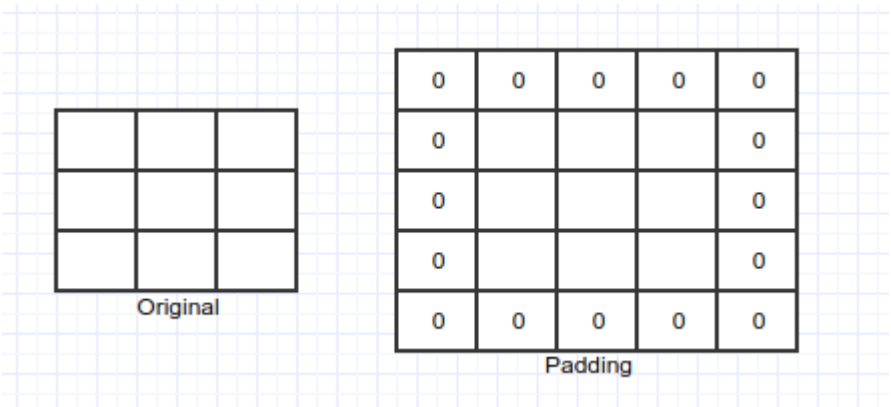
3年15組９２番　全　海成(JEON HAESUNG)
コンピュータサイエンス実習A　マルチメディア　二回目　レポート

必須課題１
以前やった一回目の実習で使ったトーンカーブは一個の画素を入力とし一個の画素を出力に出す関数でした。今回具現したフィルタはそれと似たものです。フィルタはいくつかの画素を入力として(主にある画素を中心とした正四角形の画素)一個の画素を出力として出します。この課題で具現したものはフィルタ野中で一番簡単なものである平均フィルタです。このフィルタは出力対象の画素の座標を中心とした正四角形の領域の画素値の平均値を出力として出します。
出力画像を　f　入力画像を　g　とします。
下の数式を使ってフィルタの出力を求めます。



$$g(i,j)=\sum_{n=-w}^{w}\sum_{m=-w}^{w} f(i+m,j+n)\frac{1}{(2w+1)^2}$$

数式で i+m や j+n と書いてありますがこの数式をそのまま適用すると画像の範囲外に接近する可能性があるのでフィルタの一辺の大きさの半分(w)だけ入力画像の外側に画素を追加し画素値は 0 で設定します。こういう作業を Zero Padding と言います。



次は実行コマンドの形式です。

./ass1 [Input Image] [Output Image] [Filter Size w]

フィルタのサイズとして入力される数は上の式の中にある w です。
以下はプログラムを実行して処理する前の画像と処理後の画像です。w=4

平均フィルタは画像をぼやける効果があり画像のノイズを軽減する効果を持っています。

必須課題２

必須課題 1 では平均フィルタを具現しましたが必須課題 2 では適用するフィルタをテキストファイルで受けてそのフィルタを入力画像に適用した結果を出力に出します。テキストファイルで入力されたフィルタが正四角形じゃなかった場合エラーを出して終了します。
入力されたフィルタを h とすると以下の数式を使って処理を行います。

$$g(i,j)=\sum_{n=-w}^{w}\sum_{m=-w}^{w} f(i+m,j+n)h(m,n)$$

実行コマンドの形式です。

./ass2 [Input Image] [Output Image] [Filter File]

次はソベルフィルタと言うエッジ検出に使うフィルタを適用した結果です。
入力画像は画像処理によく使われるレナさんの画像です。

-1	0	1
-2	0	2
-1	0	1

使ったフィルタは X 軸の画素値の変化度を表してくれるフィルタです。

-1	-2	-1
0	0	0
1	2	1

上のフィルタは Y 軸の画素値の変化度を表してくれるフィルタです。

必須課題３

この課題からは幾何学的変換の中で線形変換を入力画像に対して行うプログラムを書きます。まず線形変換とは一般的に以下の式で表される変換のことを言います。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

座標(x,y)の画素を座標(x’,y’)に移動する変換です。この時画像は正の整数でできた座標系を持っているため変換された数値をそのまま使うのは無理です。そのため上の変換をまず逆変換に変えて使います。

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

こうやって出力対象の座標に逆変換を使って元の画像の座標を求めます。そのあと整数じゃない場合それを補正してその座標の近くの数値を使うのを補間と言います。この補間のやり方にはいろいろありますが必須課題 3,4,5 で使う方法は最近傍補間という方法です。出力画像を g として出力画像の座標(x’,y’)に逆変換をしてえられた座標を(x,y)とします。出力画像の画素値を以下の式を使って求めるのが最近房補間です。

$$g(x', y') = f(\lfloor x+0.5 \rfloor, \lfloor y+0.5 \rfloor)$$

必須課題で具現した変換は拡大縮小です。入力として X 軸にかける倍率と Y 軸にかける倍率を受けてその通り変換をして出力に出します。変換式は以下のようです。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad \frac{1}{(S_x S_y)} \begin{pmatrix} S_y & 0 \\ 0 & S_x \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$

以下は実行コマンドの形式です。

./ass3 [Input Image] [Output Image] [X ratio] [Y ratio]

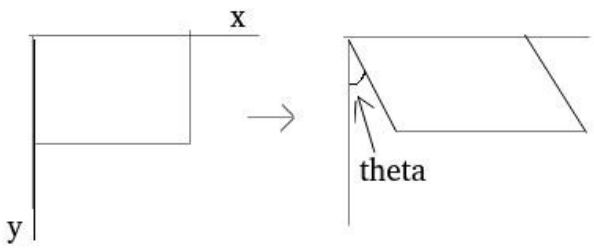
処理前と処理後の画像をワードプロセッサ上で見せても違いがはっきり分かるものではないのでターミナル上で実行して画像ファイルの情報を確認した図を入れます。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ ls
alpha_test2.JPG  ass1.c  ass3.c  ass5     dog.JPG   dst4_1.JPG  dst.JPG       dst_opt3.JPG  filter.txt  opt1    opt2.c  opt4    opt5.c      src.JPG
alpha_test.jpg   ass2    ass4    ass5.c   dst1.JPG  dst4.JPG    dst_opt1.JPG  dstX2.JPG     Lenna.JPG   opt1.c  opt3    opt4.c  sobelX.txt
ass1             ass2.c  ass4.c  cat.jpg  dst2.JPG  dst5.JPG    dst_opt2.JPG  dstY2.JPG     libgd       opt2    opt3.c  opt5    sobelY.txt
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ gcc -o ass3 ass3.c -lgd -lm
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ ./ass3 src.JPG dst3.JPG 0.5 0.5
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ identify src.JPG
src.JPG JPEG 1024x683 1024x683+0+0 8-bit sRGB 322KB 0.010u 0:00.010
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ identify dst3.JPG
dst3.JPG JPEG 512x341 512x341+0+0 8-bit sRGB 41.3KB 0.000u 0:00.000
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$
```

結果ファイルの dst3.JPG の情報を見ると 1024x683 だった画像度が 512x341 に変わったのを確認できます。下は処理前と処理後の画像です。


必須課題４

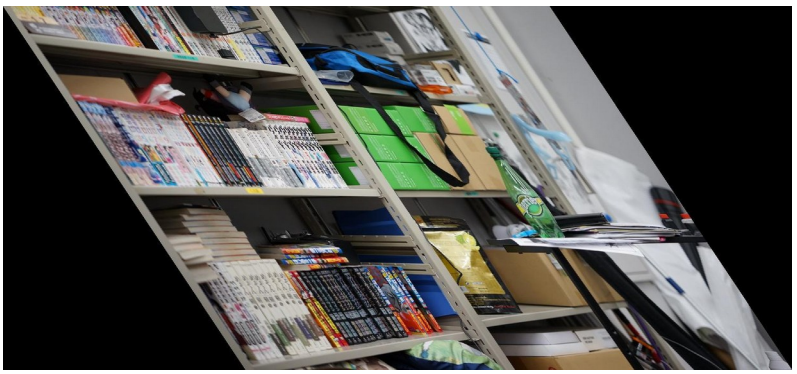
この問題で行う変換はスキューと言うもので画像を平行四辺形にする変換です。この課題では x 軸方向に傾けます。以下の図のようにします。



変換式は以下のとおりです。

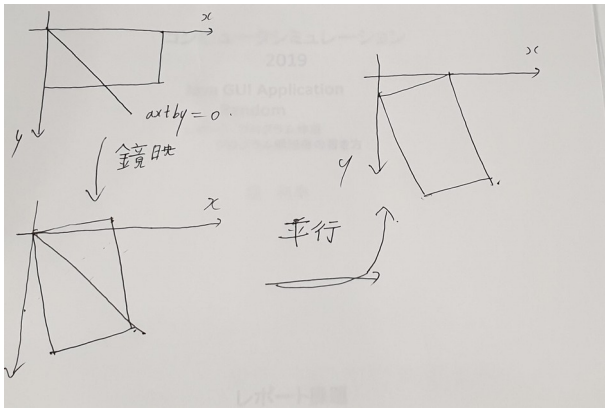
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad \begin{pmatrix} 1 & -b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \quad b = \arctan(\theta)$$

式で使われるシータは入力で受けます。45 度スキュー処理をする前と後の画像です。

必須課題５

この課題は画像を任意の直線に対して鏡映する課題です。鏡映する場合正の座標系から抜け出すことがありえます。その場合抜け出した分だけ平行移動をして補正する必要があります。以下の図はその過程を手書きで書いたものです。



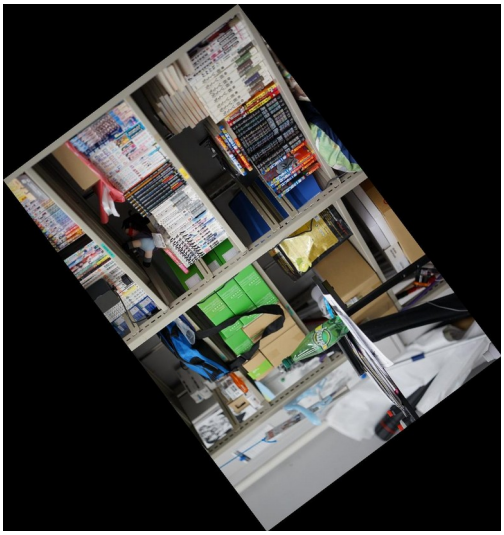
直線の方程式は ax+by=0 で a,b は入力で受けます。

変換式は以下のようです。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{1}{(m^2+1)} \begin{pmatrix} 1-m^2 & 2m \\ 2m & m^2-1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad m = -\frac{a}{b}$$

鏡映なので逆変換も同じです。

以下は a=1,b=-1 の場合と a=1,b=-2 の場合の画像です。

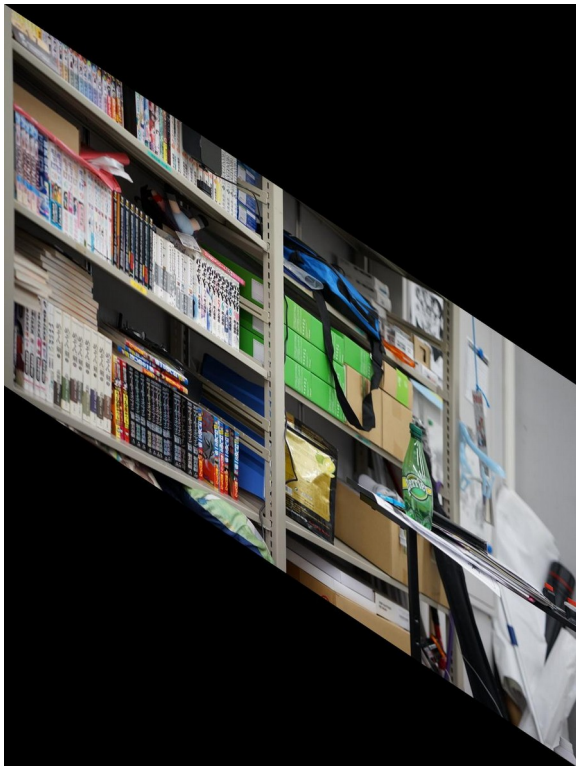


任意課題１、３

必須課題３、４、５で使った補間法は最近房補間でしたがこの任意課題ではバイリニア補間方法を使ったものです。ここでは補間方法の数式だけを書いておきます。

$$\begin{aligned} g(x',y') = {} & (\lfloor x \rfloor+1-x)(\lfloor y \rfloor+1-y) f(\lfloor x \rfloor, \lfloor y \rfloor) \\ & +(\lfloor x \rfloor+1-x)(y-\lfloor y \rfloor) f(\lfloor x \rfloor, \lfloor y \rfloor+1) \\ & +(x-\lfloor x \rfloor)(\lfloor y \rfloor+1-y) f(\lfloor x \rfloor+1, \lfloor y \rfloor) \\ & +(\lfloor x \rfloor-x)(\lfloor y \rfloor-y) f(\lfloor x \rfloor+1, \lfloor y \rfloor+1) \end{aligned}$$

任意課題２
必須課題４でやったスキューをｙ軸方向にしてバイリニア補間を使った課題です。
以下は処理前と処理後の画像です。



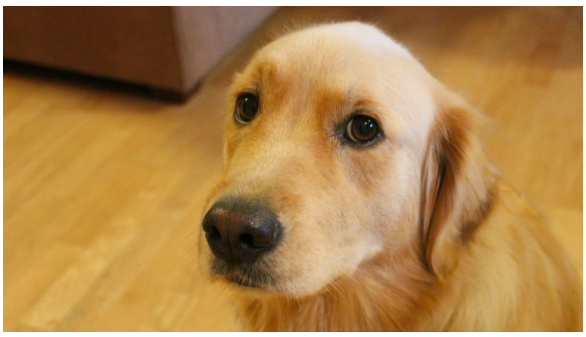
任意課題４
アルファブレンディングとは二つの画像を一つに重ねる処理で以下の数式を使って出力の値を求めます。

$$g(i,j)=\alpha f_1(i,j)+(1-\alpha)f_2(i,j)$$

実行コマンドは以下の形式です。

./opt4 [Input Image 1] [Input Image 2] [Output Image] [Alpha]

以下は　$\alpha=0.5$　の場合の処理前の画像と処理後の画像です。



アルファの値がおかしいときや入力画像がそれぞれに違う画像度を持っている場合エラーを出して終了します。

任意課題５
この課題はアルファの変わりに出力枚数を入力として受けます。それからアルファが 0 だった場合から 1 の場合までアルファを順番に変えながら入力されたプレフィックスを利用して出力します。
実行コマンドは以下の形式です。

./opt5 [Input File 1] [Input File 2] [Prefix] [# of Output]

実行画面と結果です。

```
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ ls
alpha_test2.JPG  ass1.c  ass3    ass4.c  cat.jpg   dst2.JPG    dst4.JPG    dst5.JPG      dst_opt2.JPG  dstX2.JPG   Lenna.JPG  opt1.c  opt3    opt4.c  sobelX.txt
alpha_test.jpg   ass2    ass3.c  ass5    dog.JPG   dst3.JPG    dst5_1.JPG  dst.JPG       dst_opt3.JPG  dstY2.JPG   libgd      opt2    opt3.c  opt5    sobelY.txt
ass1             ass2.c  ass4    ass5.c  dst1.JPG  dst4_1.JPG  dst5_2.JPG  dst_opt1.JPG  dst_opt4.JPG  filter.txt  opt1       opt2.c  opt4    opt5.c  src.JPG
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ ./opt5 cat.jpg dog.JPG out 5
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$ ls
alpha_test2.JPG  ass2    ass4    cat.jpg   dst3.JPG    dst5_2.JPG    dst_opt2.JPG  dstY2.JPG   opt1    opt3    opt5       out02.JPG  sobelY.txt
alpha_test.jpg   ass2.c  ass4.c  dog.JPG   dst4_1.JPG  dst5.JPG      dst_opt3.JPG  filter.txt  opt1.c  opt3.c  opt5.c     out03.JPG  src.JPG
ass1             ass3    ass5    dst1.JPG  dst4.JPG    dst.JPG       dst_opt4.JPG  Lenna.JPG   opt2    opt4    out00.JPG  out04.JPG
ass1.c           ass3.c  ass5.c  dst2.JPG  dst5_1.JPG  dst_opt1.JPG  dstX2.JPG     libgd       opt2.c  opt4.c  out01.JPG  sobelX.txt
seastar105@seastar105-Strix-GL504GW:~/Work/MeijiCompScienceA/compscienceAmm_cs_exe/week2$
```





ドキュメンタリーとかでこういう画面転換を見たことがあると思いますがその時使うのがアルファブレンディングです。
参考文献
講義資料