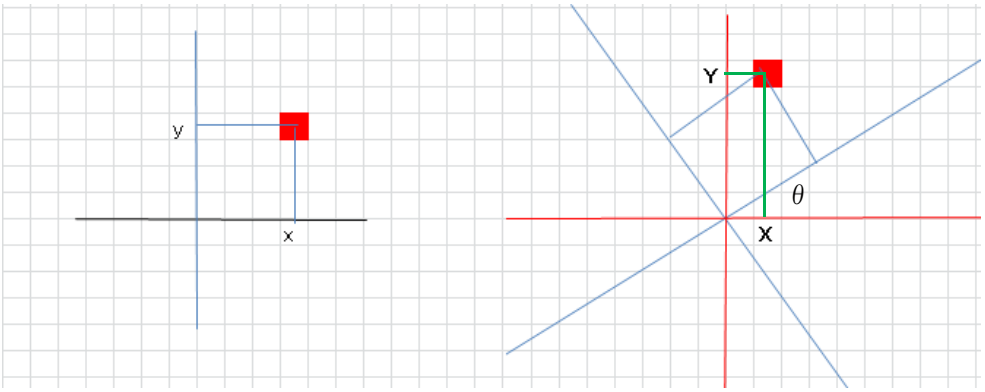
メディアプログラミング演習―第３回（第１テーマ３日目）―

演習３：画像の回転

与えられた画像を，角度θ回転させた画像を生成する．次図３－１を参照し解説する



(a)元画像のピクセル座標　　(b)目的画像のピクセル座標

図３－１　回転の考え方

(注意)以下、(x,y)および(X,Y)は異なる（小文字／大文字）ので注意のこと。

演習３－１：正方向写像

元画像(a)のピクセル(x,y)は，原点を中心にθ度回転した場合，目的画像(b)の，式(3-1)で表現されるピクセル(X,Y)に移動する．

X=x*cosθ -y*sinθ　　Y=x*sinθ +y*cosθ　　(3-1)

回転の中心を(x0,y0)とすると次式となる。

X=(x-x0)*cosθ -(y-y0)*sinθ +x0　　(3-2)

Y=(x-x0)*sinθ +(y-y0)*cosθ +y0

よって，すべての元ピクセル(x,y)の値を，回転後の新ピクセル（X,Y）にコピーすれば，回転した画像を得ることができると考えられる．

画像の中心で回転させる場合は，画像中心の座標は，(w/2, h/2)である（w は幅，h は高さ）ので，以下となる。

```
int X=int((x-w/2)*cosθ -(y-h/2)*sinθ +w/2);
int Y=int((x-w/2)*sinθ +(y-h/2)*cosθ +h/2);
```

しかし、回転させると、はみ出す部分があるので、はみ出た場合、コピーしない処理が必要となる。以下の通りである。

```
if( (X>=0)&&(X<w)&&(Y>=0)&&(Y<h) )  ＜－範囲内かの判定
{       int OldPos = x + y*w;     ＜－　元画像におけるピクセル(x,y)
        int NewPos = X + Y*w;      ＜－目的画像におけるピクセル(X,Y)
        img_out.pixels[NewPos] = img_in.pixels[OldPos];
}
```

これらを、すべての元画像の(x,y)に対して繰り返せばよい。これを参考にプログラム(sample3-1)を完成させなさい。

演習３－１の結果画像をよく見ると，抜けている（コピーされていない）ピクセルがある．この理由を考察しなさい。

演習３－２：逆方向写像

そこで，以下のように考える．すなわち，演習３－１では，すべての元画像のピクセルを，新画像にコピーするようにした．ここでは，逆に，すべての新画像のピクセル(X,Y)に対して，それが元画像のどのピクセル(x,y)に対応するかを求め，そのピクセルの値を新ピクセルにコピーするように作り換える．

さて，(x,y)から(X,Y)と求めるのが式(3-2)であるので，(X,Y)から(x,y)を求めるためには，式(3-2)を，x=～，　y=～　の式に変形する必要がある．

よって、下記をすべての新画像の(X,Y)で繰り返せばよい。

```
int x= <<< >>>  ;  <=この部分は，変形した式を参考にする．
int y= <<< >>>  ;  <=          上に同じ
if( (x>=0)&&(x<w)&&(y>=0)&&(y<h) )  /* 範囲検証は元画面座標で */
{       int OldPos = x + y*w;
        int NewPos = X + Y*w;
        img_out.pixels[NewPos] = img_in.pixels[OldPos];
}
```