# 영상처리 HW5

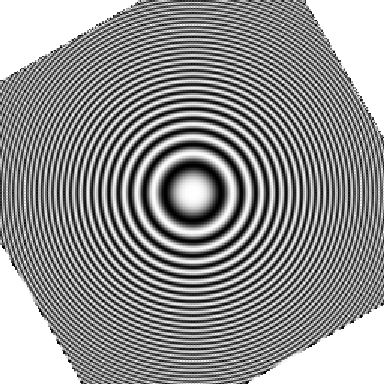
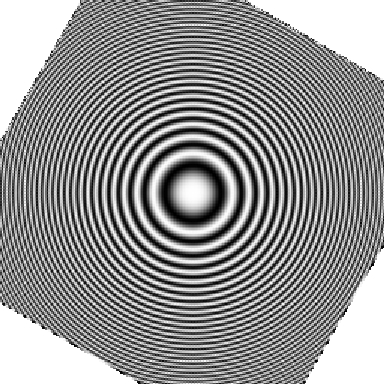
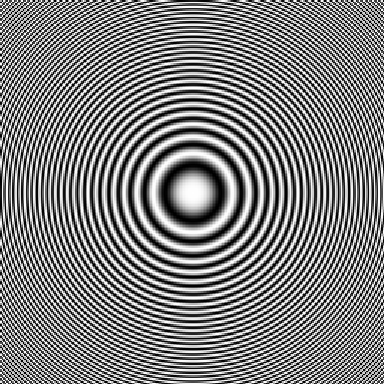
16010980 이우석

## 출력 결과

* lena.pgm / lena\_30angle.pgm / lena\_60angle.pgm



* czp.pgm / czp\_30angle.pgm / czp\_60angle.pgm



## 코드

void bilinear\_rotation(image\_ptr buffer, char\* fileout, int rows, int cols,

float angle, int type)

{

unsigned long x, y; /\* loop indices for columns and rows \*/

unsigned long index; /\* index into line buffer \*/

unsigned long source\_index; /\* address of source pixel in image buffer \*/

unsigned char\* line\_buff; /\* output line buffer \*/

int new\_rows, new\_cols; /\* values of rows and columns for new image \*/

unsigned line; /\* number of pixels in one scan line \*/

FILE\* fp; /\* output file pointer \*/

unsigned long X\_Source, Y\_Source; /\* x and y address of source pixel \*/

pixel\_ptr color\_buff; /\* pointer to a color image in memory \*/

int NW, NE, SW, SE, result;

float EWweight, NSweight, EWtop, EWbottom;

double radian = getRadian(angle); // radian 값을 받을 변수

unsigned long x\_center, y\_center; // 이미지의 중앙 좌표를 받을 변수

/\* open new output file \*/

if ((fp = fopen(fileout, "wb")) == NULL)

{

printf("Unable to open %s for output\n", fileout);

exit(1);

}

// 중앙 부분의 좌표

x\_center = cols / 2;

y\_center = rows / 2;

/\* print out the portable bitmap header \*/

// fprintf(fp, "P%d\n%d %d\n255\n", type, new\_cols, new\_rows);

fprintf(fp, "P%d\n%d %d\n255\n", type, cols, rows);

if (type == 5) /\* PGM file \*/

line = cols;

else /\* PPM file \*/

{

line = cols \* 3;

color\_buff = (pixel\_ptr)buffer;

}

line\_buff = (unsigned char\*)malloc(line);

for (y = 0; y < rows; y++)

{

index = 0;

for (x = 0; x < cols; x++)

{

// 원본 파일의 좌표

X\_Source = (unsigned long)(cos(radian) \* ((double)x - x\_center) + sin(radian) \* ((double)y - y\_center) + (double)x\_center);

Y\_Source = (unsigned long)((-1) \* sin(radian) \* ((double)x - x\_center) + cos(radian) \* ((double)y - y\_center) + (double)y\_center);

if (type == 5) /\* PGM \*/

{

/\* bilinear interpolation \*/

// duplicate the border edges

// X\_Source, Y\_Source 가 원본 이미지의 가로, 세로 크기를 넘지 않는다면,

if (0 <= X\_Source && X\_Source < cols && 0 <= Y\_Source && Y\_Source < rows)

{

NW = buffer[Y\_Source \* cols + X\_Source];

if (X\_Source + 1 >= cols && Y\_Source + 1 >= rows)

{

NE = NW;

SW = NW;

SE = NW;

}

else if (X\_Source + 1 >= cols)

{

NE = NW;

SW = buffer[(Y\_Source + 1) \* cols + X\_Source];

SE = SW;

}

else if (Y\_Source + 1 >= rows)

{

NE = buffer[Y\_Source \* cols + X\_Source + 1];

SW = NW;

SE = NE;

}

else

{

NE = buffer[Y\_Source \* cols + X\_Source + 1];

SW = buffer[(Y\_Source + 1) \* cols + X\_Source];

SE = buffer[(Y\_Source + 1) \* cols + X\_Source + 1];

}

// 가중치를 계산할 때에도,

EWweight = (cos(radian) \* ((double)x - x\_center) + sin(radian) \* ((double)y - y\_center) + (double)x\_center) - (float)X\_Source;

NSweight = ((-1) \* sin(radian) \* ((double)x - x\_center) + cos(radian) \* ((double)y - y\_center) + (double)y\_center) - (float)Y\_Source;

EWtop = NW + EWweight \* (NE - NW);

EWbottom = SW + EWweight \* (SE - SW);

result = EWtop + NSweight \* (EWbottom - EWtop);

line\_buff[index++] = result;

}

// X\_Source, Y\_Source 가 원본 이미지의 가로, 세로 크기를 넘는다면,

else

{

// 흰색으로 표현함.

line\_buff[index++] = 255;

}

}

else /\* PPM \*/

{

line\_buff[index++] = color\_buff[source\_index].r;

line\_buff[index++] = color\_buff[source\_index].g;

line\_buff[index++] = color\_buff[source\_index].b;

}

}

fwrite(line\_buff, 1, line, fp);

}

fclose(fp);

}