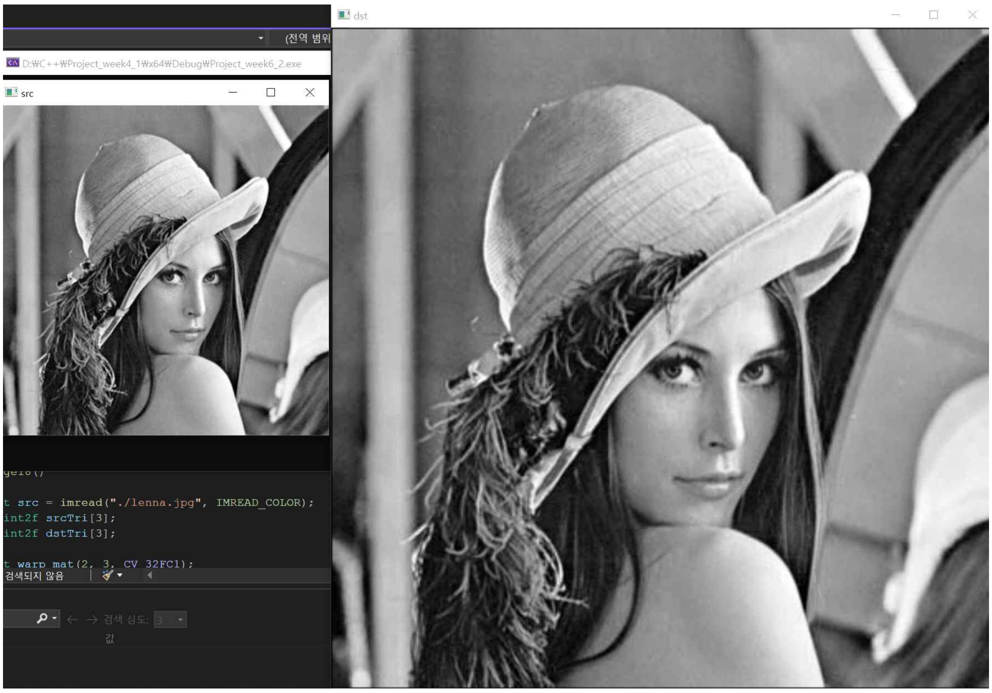
영상처리 실제 6주차 실습_기하학적 변환

2023254015 장욱진

```
#include <opencv2/opencv.hpp>
using namespace std;
using namespace cv;

float Lerp(float s, float e, float t) {
    return s + (e - s) * t;
}
float Blerp(float c00, float c10, float c01, float c11, float tx, float ty) {
    return Lerp(Lerp(c00, c10, tx), Lerp(c01, c11, tx), ty);
}
float GetPixel(Mat img, int x, int y)
{
    if (x > 0 && y > 0 && x < img.cols && y < img.rows)
        return (float)(img.at<uchar>(y, x));
    else
        return 0.0;
}

void page10()
{
    Mat src = imread("./lenna.jpg", IMREAD_GRAYSCALE);
    Mat dst = Mat::zeros(Size(src.cols * 2, src.rows * 2), src.type());
    for (int y = 0; y < dst.rows; y++) {
        for (int x = 0; x < dst.cols; x++) {
            float gx = ((float)x) / 2.0;
            float gy = ((float)y) / 2.0;
            int gxi = (int)gx;
            int gyi = (int)gy;
            float c00 = GetPixel(src, gxi, gyi);
            float c10 = GetPixel(src, gxi + 1, gyi);
            float c01 = GetPixel(src, gxi, gyi + 1);
            float c11 = GetPixel(src, gxi + 1, gyi + 1);
            int value = (int)Blerp(c00, c10, c01, c11, gx - gxi, gy - gyi);
            dst.at<uchar>(y, x) = value;
        }
    }
}

int page18()
{
    Mat src = imread("./lenna.jpg", IMREAD_COLOR);
    Point2f srcTri[3];
    Point2f dstTri[3];

    Mat warp_mat(2, 3, CV_32FC1);
    Mat warp_dst;

    warp_dst = Mat::zeros(src.rows, src.cols, src.type());

    srcTri[0] = Point2f(0, 0);
    srcTri[1] = Point2f(src.cols - 1.0f, 0);
    srcTri[2] = Point2f(0, src.rows - 1.0f);
    dstTri[0] = Point2f(src.cols * 0.0f, src.rows * 0.33f);
    dstTri[1] = Point2f(src.cols * 0.85f, src.rows * 0.25f);
    dstTri[2] = Point2f(src.cols * 0.15f, src.rows * 0.7f);
```

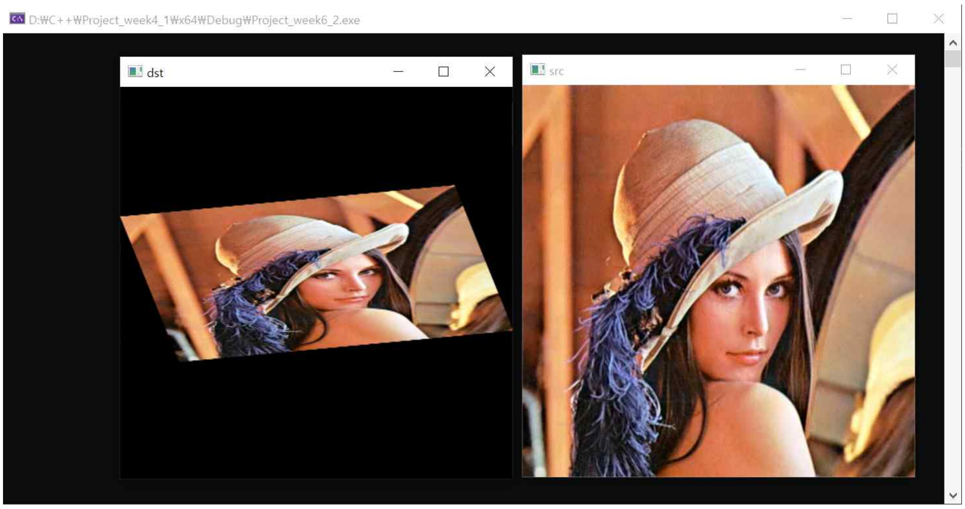
```
	warp_mat = getAffineTransform(srcTri, dstTri);
	warpAffine(src, warp_dst, warp_mat, warp_dst.size());

	imshow("src", src);
	imshow("dst", warp_dst);

	waitKey(0);
	return 0;
}

void page23()
{
	Mat src = imread("d:/book.jpg");
	Point2f inputp[4];
	inputp[0] = Point2f(30, 81);
	inputp[1] = Point2f(274, 247);
	inputp[2] = Point2f(298, 40);
	inputp[3] = Point2f(598, 138);
	Point2f outputp[4];
	outputp[0] = Point2f(0, 0);
	outputp[1] = Point2f(0, src.rows);
	outputp[2] = Point2f(src.cols, 0);
	outputp[3] = Point2f(src.cols, src.rows);
	Mat h = getPerspectiveTransform(inputp, outputp);
	Mat out;
	warpPerspective(src, out, h, src.size());
	imshow("Source Image", src);
	imshow("Warped Source Image", out);
	waitKey(0);

}

int main()
{
	page10();
	page18();
	page23();

	return 0;
}
```

결과화면

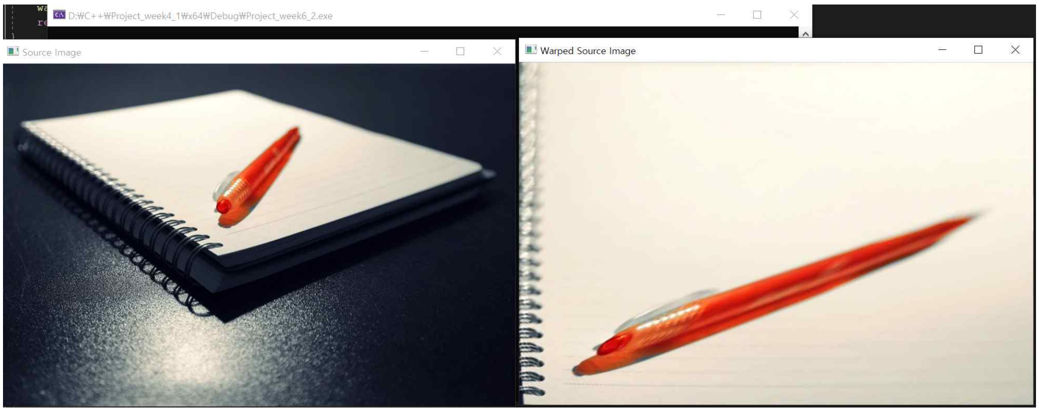


```
t src = imread("./lenna.jpg", IMREAD_COLOR);
int2f srcTri[3];
int2f dstTri[3];

t warp mat(2, 3, CV 32FC1);
```

<page10 결과화면>



<page18 결과화면>



<page23 결과화면>