

10-공간필터링 8page

Visual Studio Code interface showing the implementation of a spatial filter (blur) using OpenCV. The code is in `6주차_10_8page.cpp`.

```

1  #include <opencv2/opencv.hpp>
2  using namespace cv;
3  using namespace std;
4
5  void filter(Mat img, Mat& dst, Mat mask)
6  {
7      dst = Mat(img.size(), CV_32F, Scalar(0));
8      Point h_m = mask.size() / 2;
9
10     for (int i = h_m.y; i < img.rows - h_m.y; i++) {
11         for (int j = h_m.x; j < img.cols - h_m.x; j++) {
12             float sum = 0;
13             for (int u = 0; u < mask.rows; u++) {
14                 for (int v = 0; v < mask.cols; v++) {
15                     int y = i + u - h_m.y;
16                     int x = j + v - h_m.x;
17                     sum += mask.at<float>(u, v) * img.at<uchar>(y, x);
18                 }
19             }
20             dst.at<float>(i, j) = sum;
21         }
22     }
23 }
24
25 int main()
26 {
27     Mat image = imread("C:/Users/Chan's Victus/Documents/class/Project/image/filter_blur.jpg",
28                       IMREAD_GRAYSCALE);
29     CV_Assert(image.data);
30
31     float data[] = {
32         1 / 9.f, 1 / 9.f, 1 / 9.f,
33         1 / 9.f, 1 / 9.f, 1 / 9.f,
34         1 / 9.f, 1 / 9.f, 1 / 9.f
35     };
36     Mat mask(3, 3, CV_32F, data);
37     Mat blur;
38     filter(image, blur, mask);
39     blur.convertTo(blur, CV_8U);
40
41     imshow("image", image); imshow("blur", blur);
42     waitKey();
43     return 0;
44 }

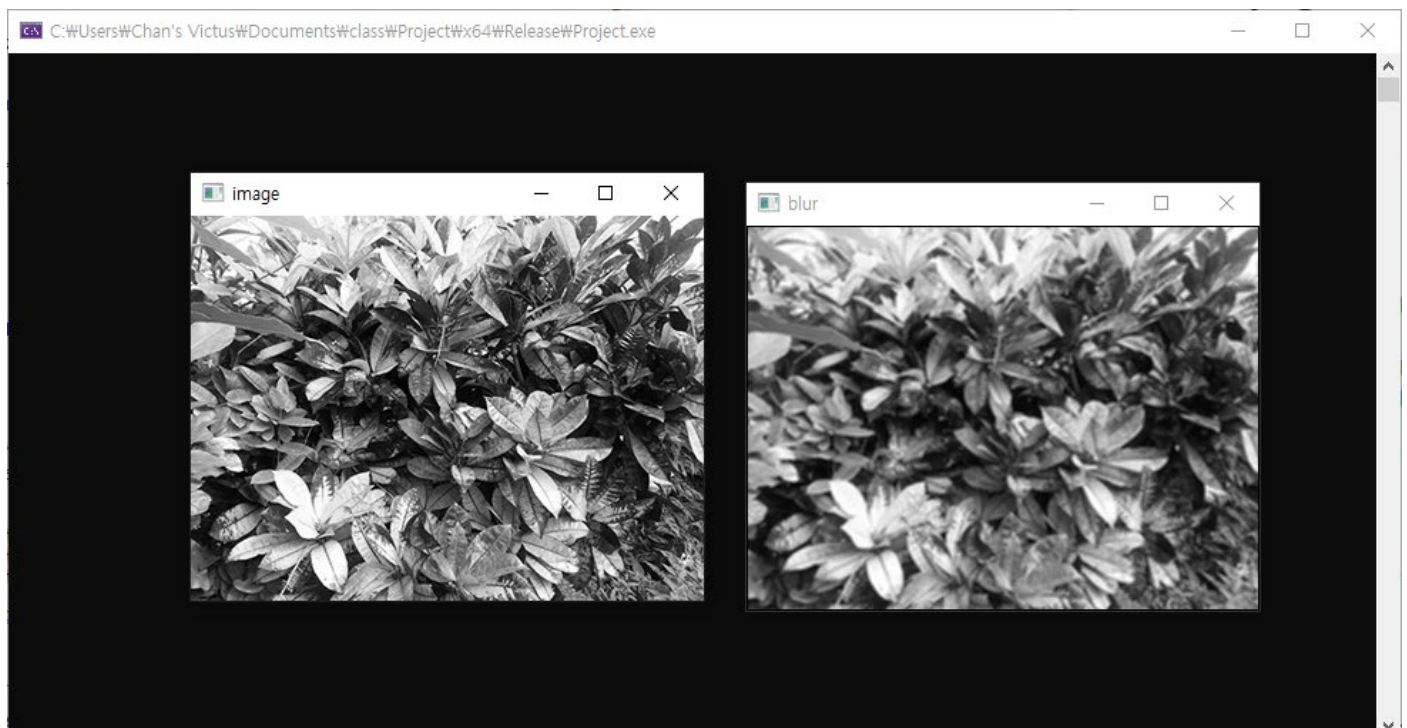
```

The output window shows the execution results:

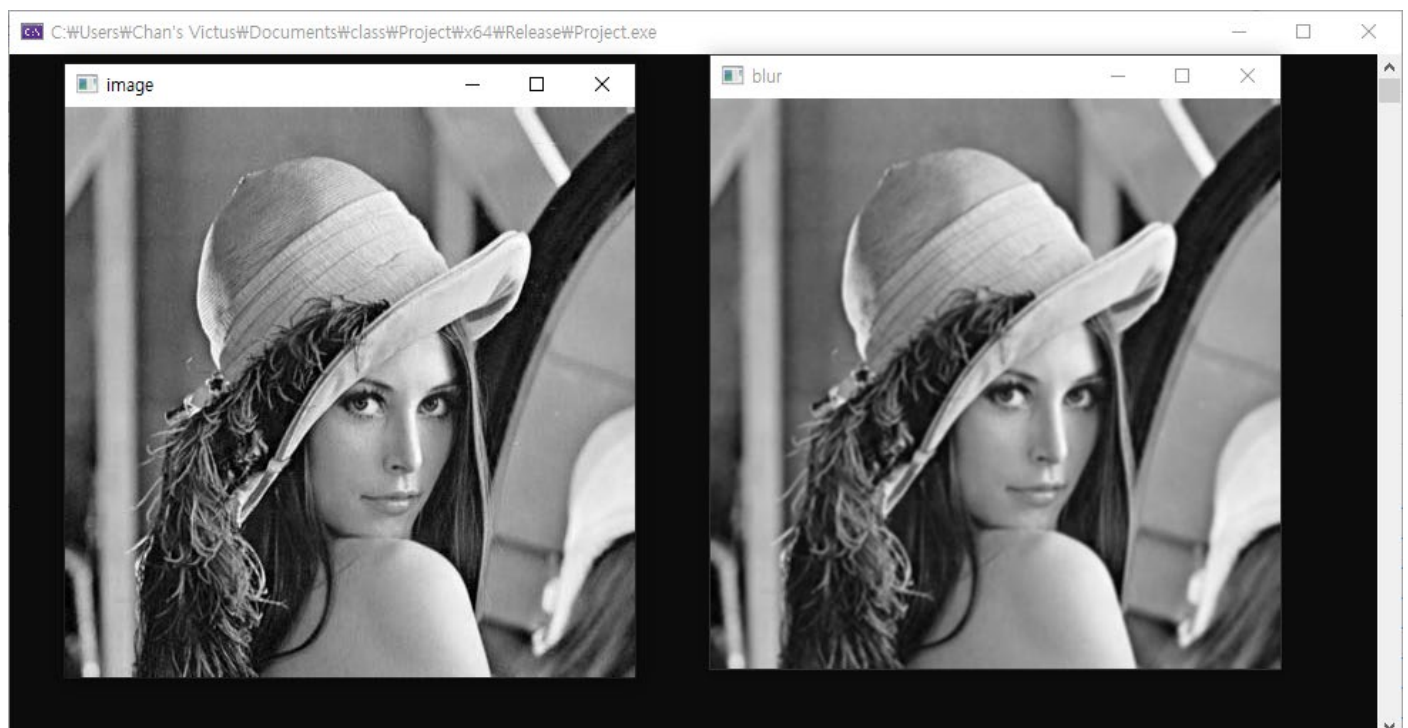
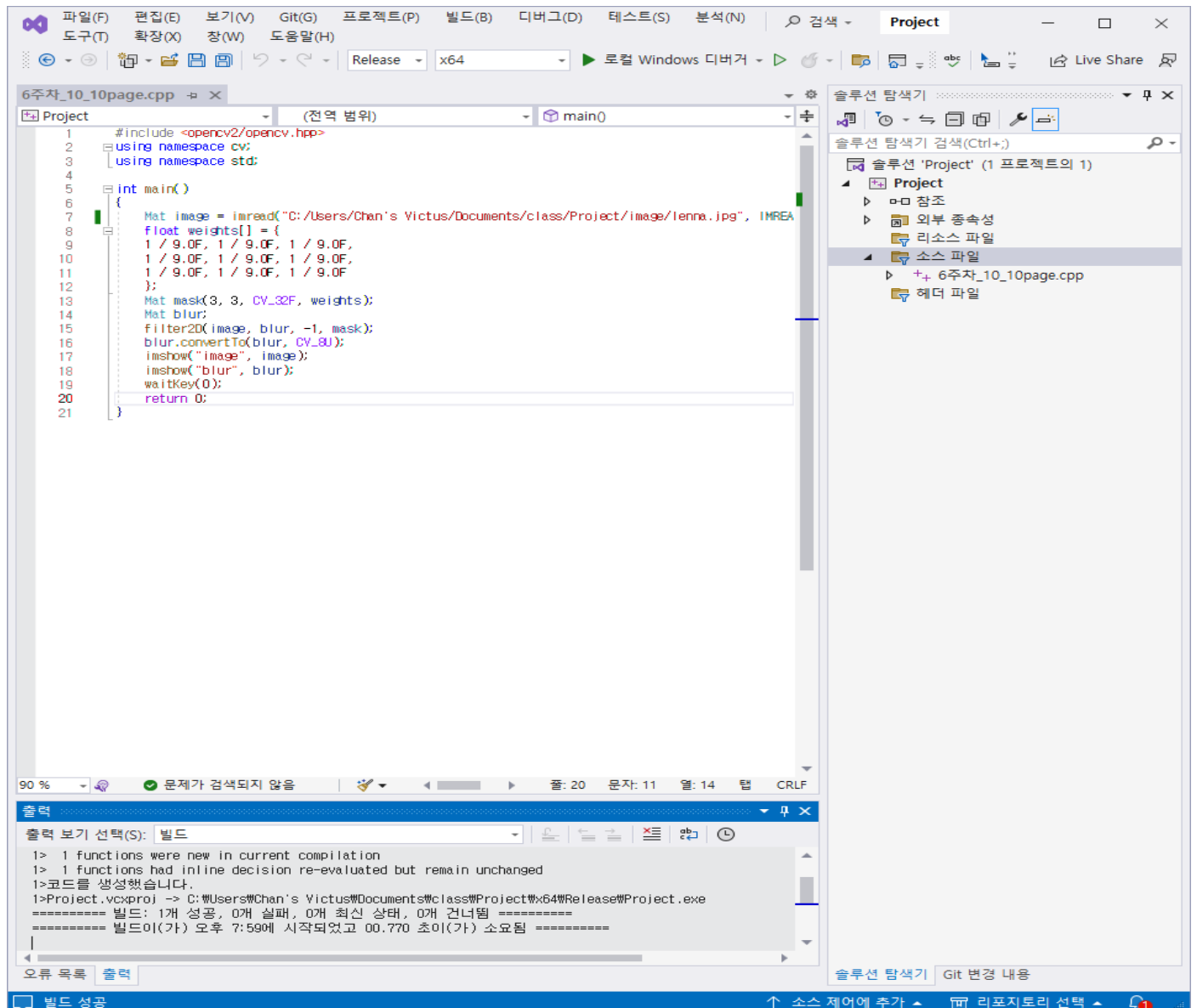
```

출력 보기 선택(S): 빌드
1> 0 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
1> "Project.vcxproj" 프로젝트를 빌드했습니다.
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 7:57에 시작되었고 02:397 초이(가) 소요됨 =====

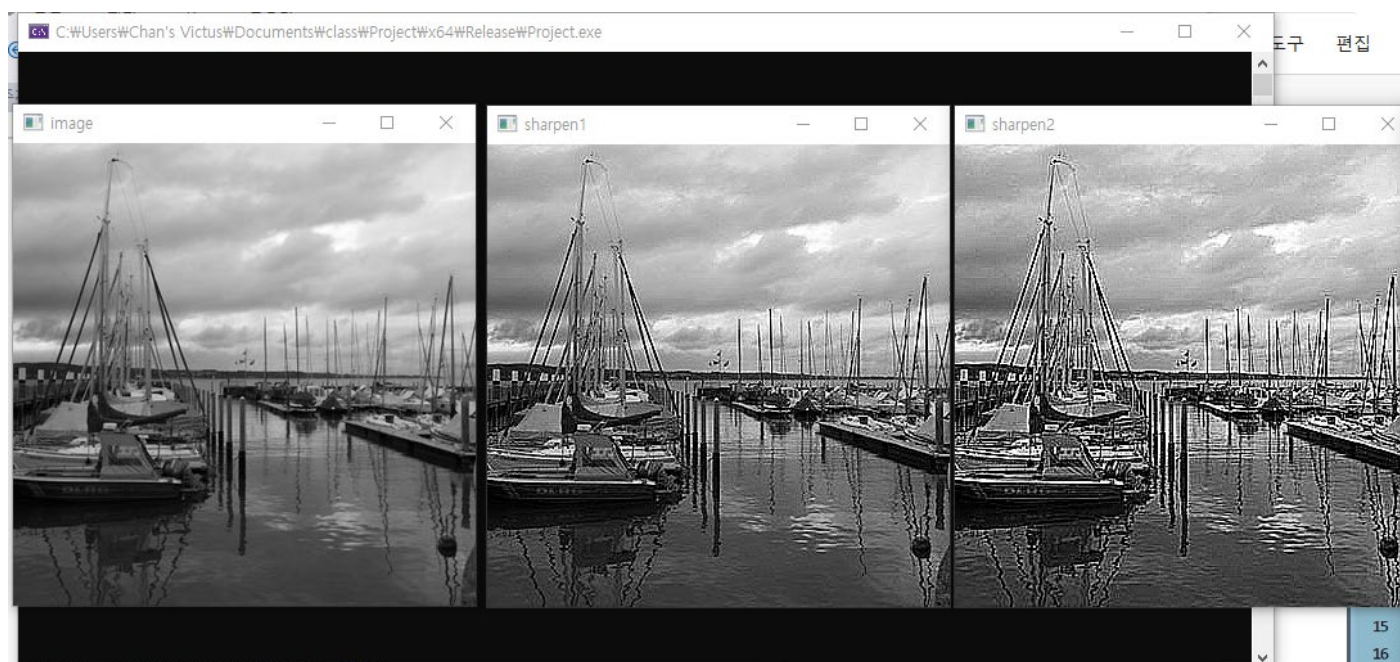
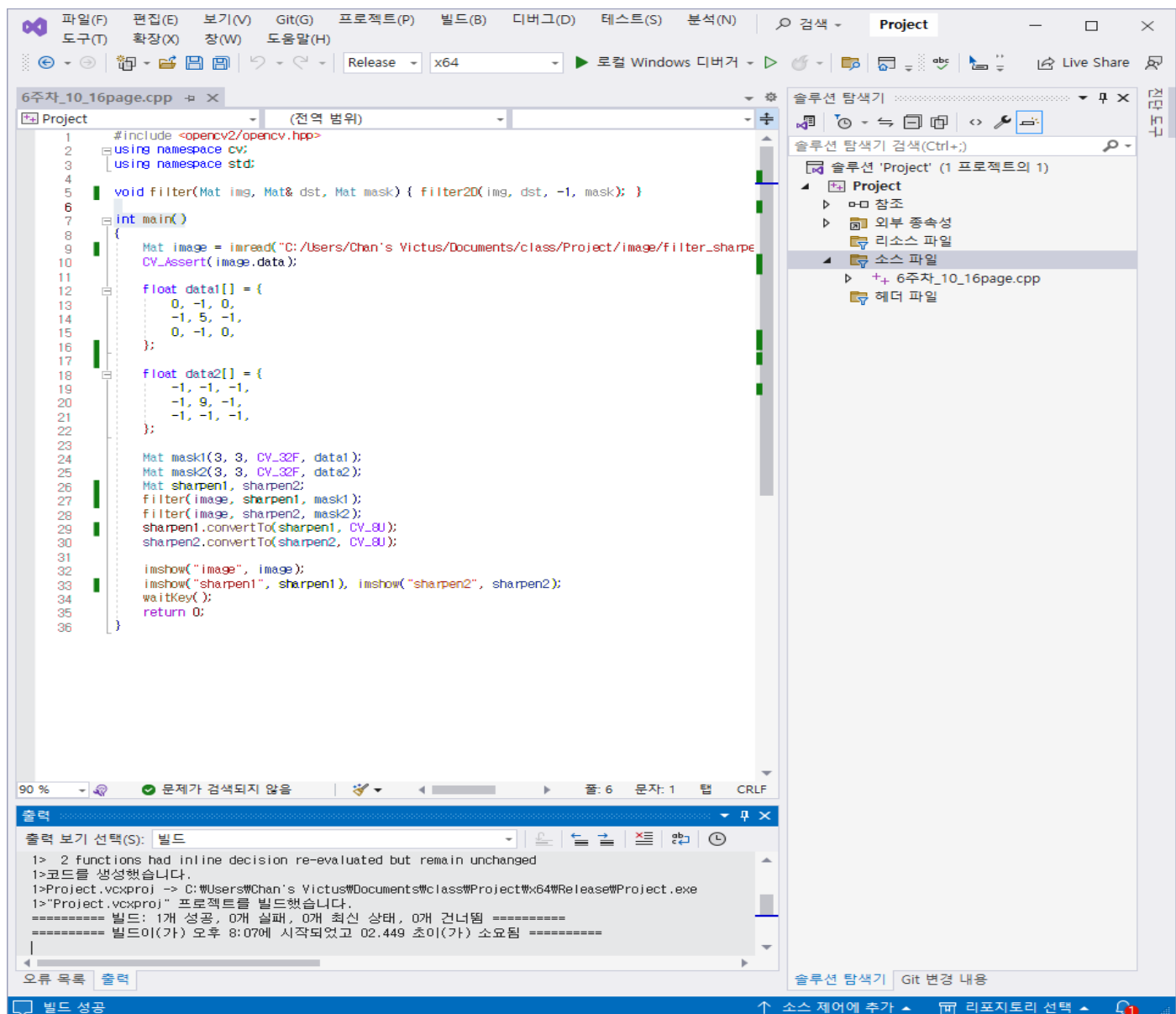
```



10-공간필터링 10page



10-공간필터링 16page



10-공간필터링 21page

Visual Studio Code interface showing the implementation of spatial filtering (median blur) on an image.

6주차_10_21page.cpp

```

1 #include <opencv2/opencv.hpp>
2 using namespace cv;
3 using namespace std;
4
5 int main()
6 {
7     Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/city1.jpg", IMF_
8     if (src.empty()) { return -1; }
9     Mat dst;
10    Mat noise_img = Mat::zeros(src.rows, src.cols, CV_8U);
11    randu(noise_img, 0, 255); // noise_img 의 모든 화소를 0 부터 255 까지의 난수로 채움
12    Mat black_img = noise_img < 10; // noise_img 의 화소값이 10 보다 작으면 1이되는 black_
13    Mat white_img = noise_img > 245; // noise_img 의 화소값이 245 보다 크면 1이되는 white_
14    Mat src1 = src.clone();
15    src1.setTo(255, white_img); // white_img 의 화소값이 1 이면 src1 화소값을 255 로 한다
16    src1.setTo(0, black_img); // black_img 의 화소값이 1 이면 src1 화소값을 0 으로 한다
17    medianBlur(src1, dst, 5);
18    imshow("source", src1);
19    imshow("result", dst);
20    waitKey(0);
21    return 0;
22 }
23

```

출력 (Output):

```

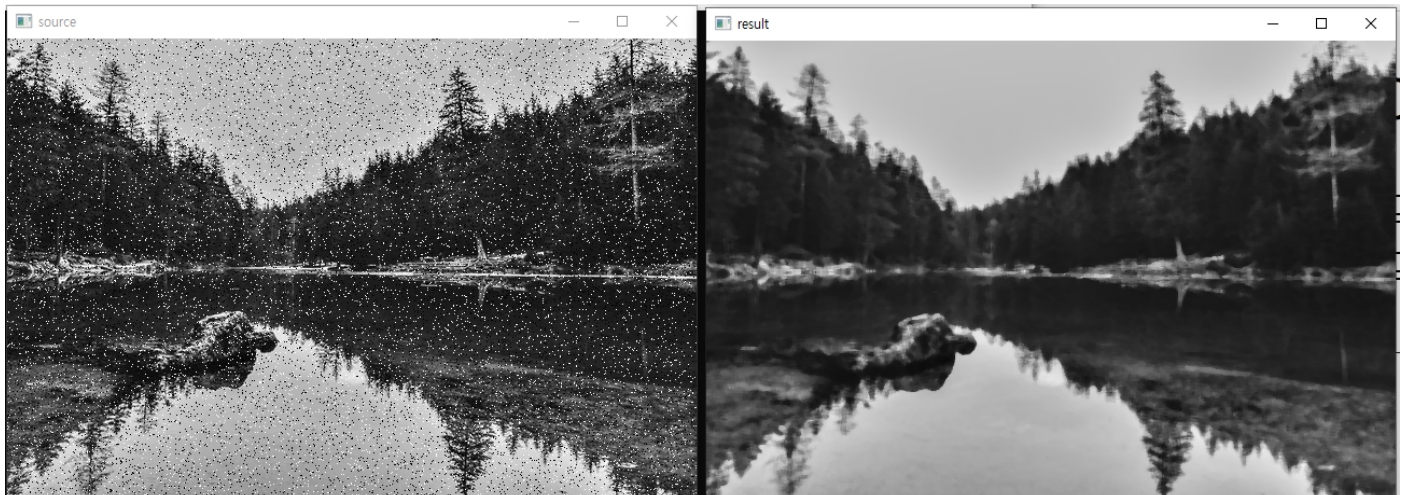
1> 8 functions were new in current compilation
1> 1 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 8:09에 시작되었고 01.015 초이(가) 소요됨 =====

```

솔루션 탐색기 (Solution Explorer):

- Project (1 프로젝트의 1)
- 참조
- 외부 종속성
- 리소스 파일
- 소스 파일
 - 6주차_10_21page.cpp
 - 헤더 파일

빌드 성공



10-공간필터링 27page

Visual Studio Code interface showing the C++ code for spatial filtering (6주차_10_27page.cpp) and the output window.

Code (6주차_10_27page.cpp):

```

1  #include <opencv2/opencv.hpp>
2  using namespace cv;
3  using namespace std;
4
5  void differential(Mat image, Mat& dst, float data1[], float data2[])
6  {
7      Mat dst1, mask1(3, 3, CV_32F, data1);
8      Mat dst2, mask2(3, 3, CV_32F, data2);
9
10     filter2D(image, dst1, CV_32F, mask1);
11     filter2D(image, dst2, CV_32F, mask2);
12     magnitude(dst1, dst2, dst);
13     dst.convertTo(dst, CV_8U);
14
15     convertScaleAbs(dst1, dst1);
16     convertScaleAbs(dst2, dst2);
17     imshow("dst1 - 수직 마스크", dst1);
18     imshow("dst2 - 수평 마스크", dst2);
19 }
20
21 int main()
22 {
23     Mat image = imread("C:/Users/Chan's Victus/Documents/class/Project/image/edge_test1.jp
24     CV_Assert(image.data);
25
26     float data1[] = {
27         -1, 0, 1,
28         -1, 0, 1,
29         -1, 0, 1
30     };
31     float data2[] = {
32         -1, -1, -1,
33         0, 0, 0,
34         1, 1, 1
35     };
36     Mat dst;
37     differential(image, dst, data1, data2);
38     imshow("image", image);
39     imshow("프러워 이미지", dst);
40     waitKey();
41     return 0;
42 }

```

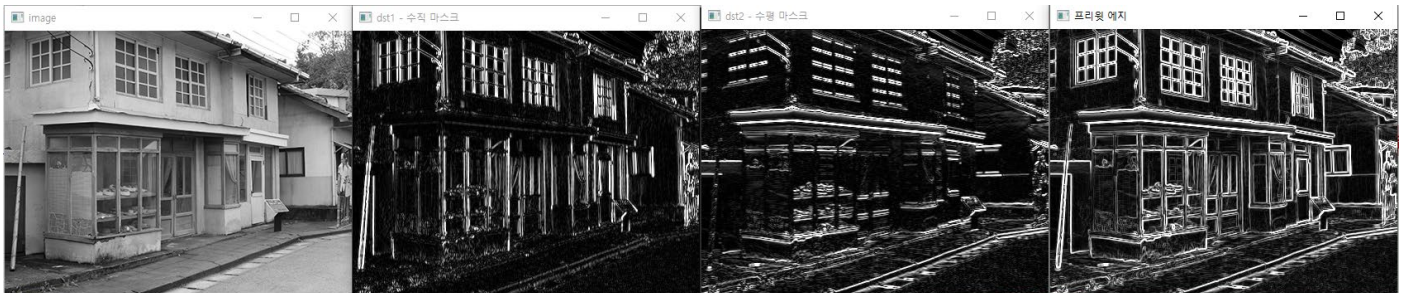
Output Window:

```

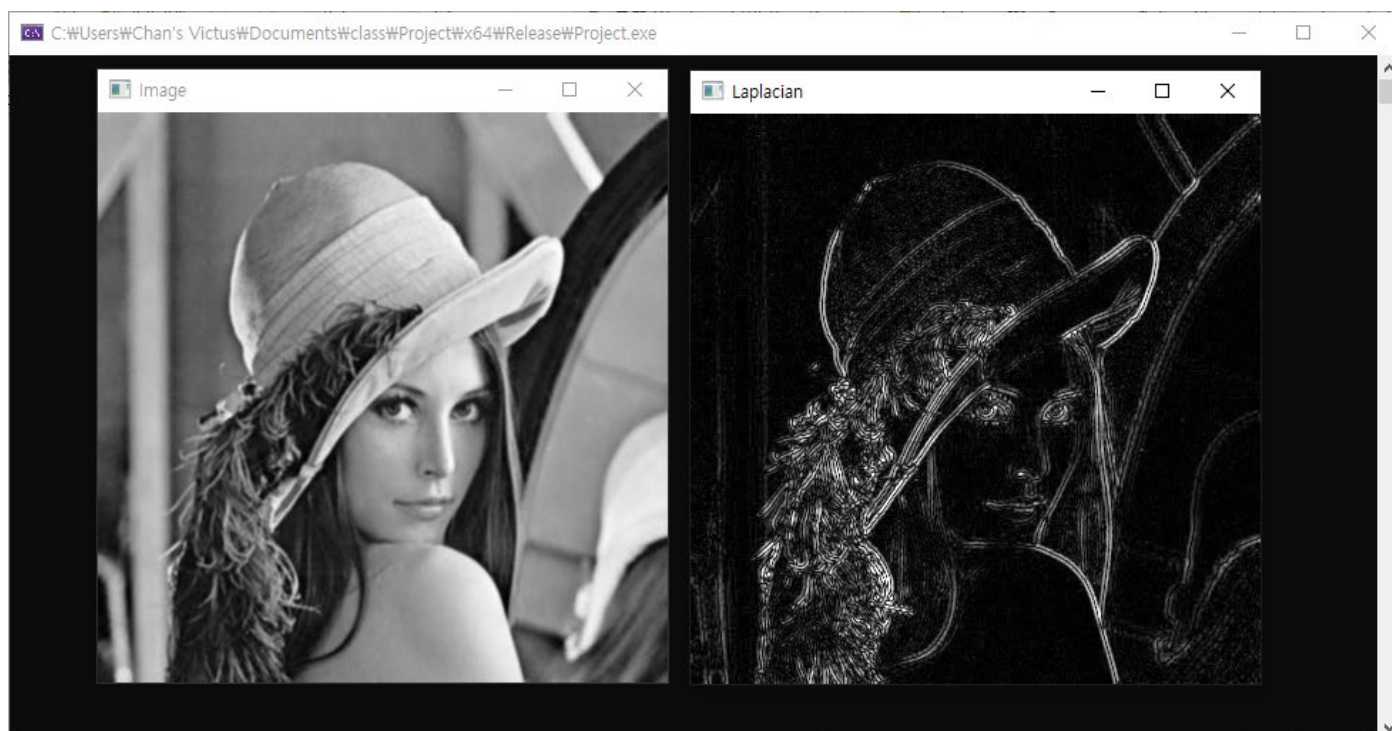
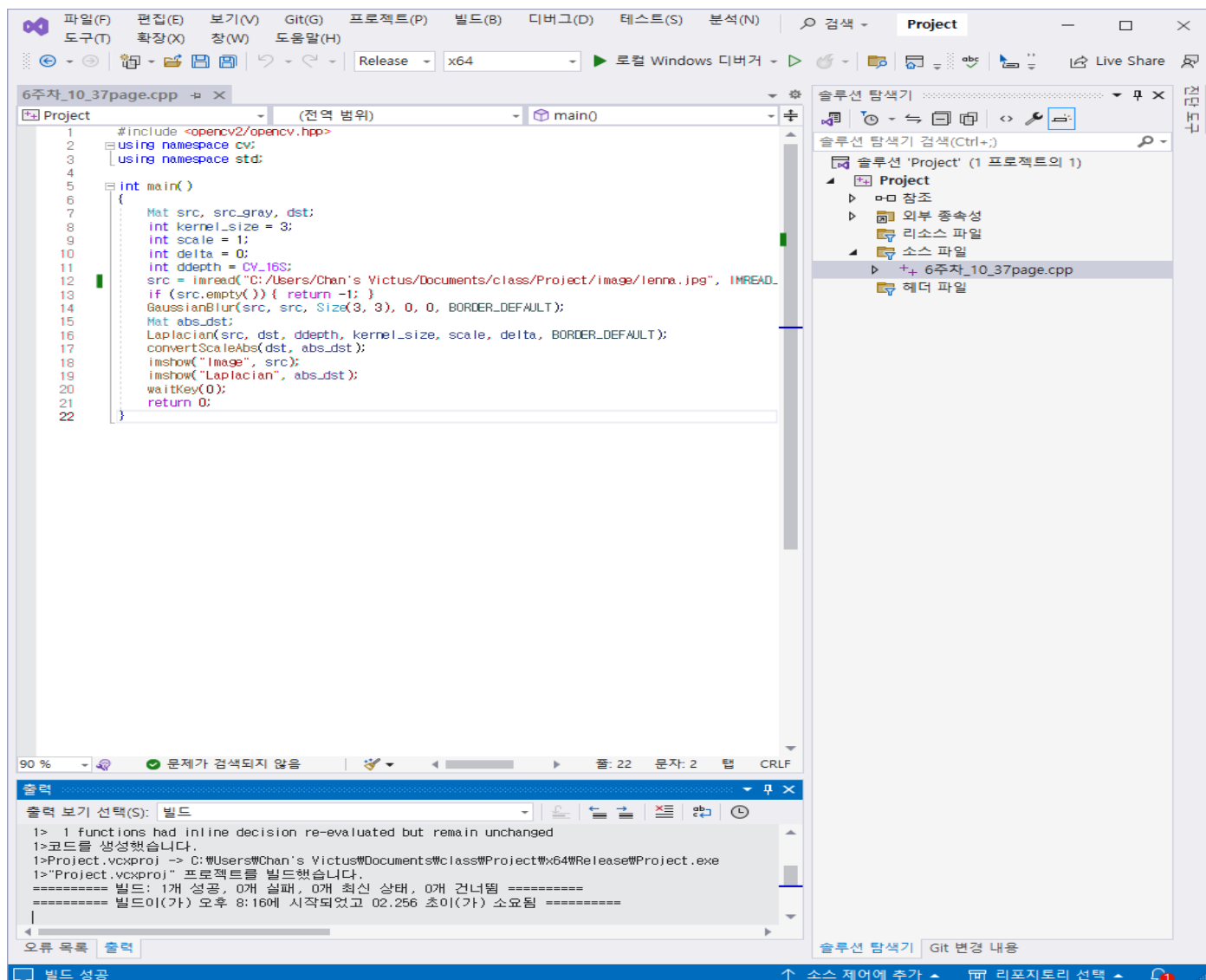
1> 2 functions were new in current compilation
1> 1 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 8:15에 시작되었고 01.173 초이(가) 소요됨 =====

```

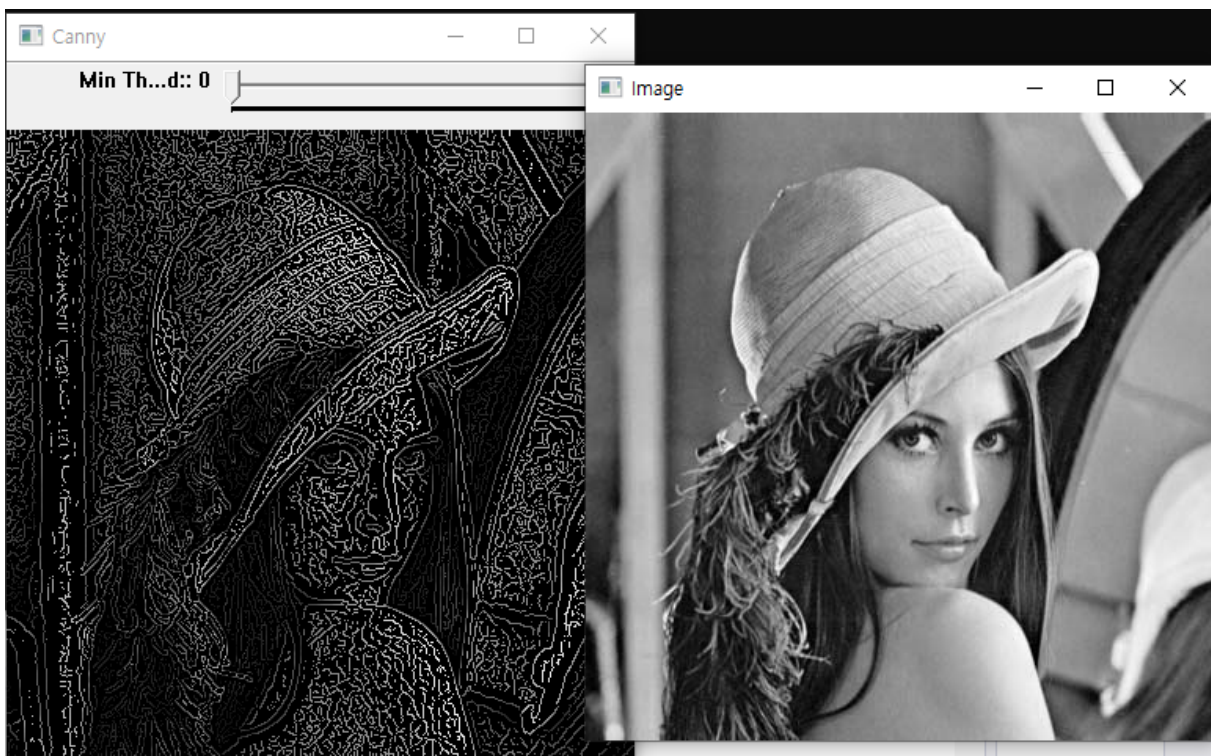
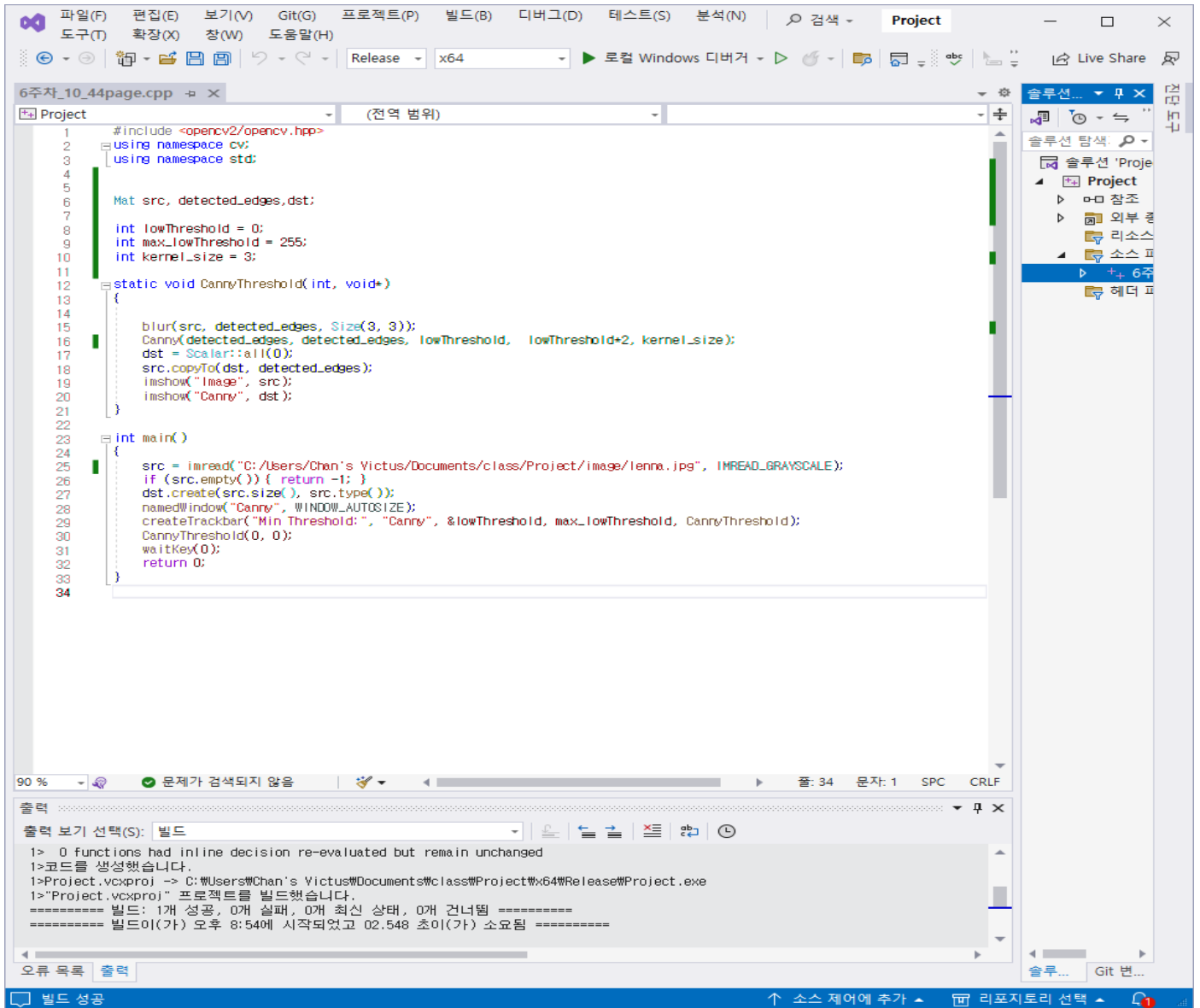
Build status: 빌드 성공 (Build Success)



10-공간필터링 37page



10-공간필터링 44page



11-기하학적 변환 10page

Visual Studio Code interface showing the implementation of bilinear interpolation in C++.

6주차_11_10page.cpp

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

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;
}

int main() {
    Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/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;
        }
    }

    imshow("Bilinear Interpolated Image", dst);
    waitKey(0);
    return 0;
}
```

출력

```
1> 0 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\6주차_11_10page\Release\Project.exe
1> "Project.vcxproj" 프로젝트를 빌드했습니다.
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 8:30에 시작되었고 02.748 초이(가) 소요됨 =====
```



11-기하학적 변환 18page

Visual Studio Code interface showing the C++ code for image warping and the output console.

```

1 #include <opencv2/opencv.hpp>
2 using namespace cv;
3 using namespace std;
4
5 int main()
6 {
7     Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/lenna.jpg", IMREAD_COLOR);
8     Point2f srcTri[3];
9     Point2f dstTri[3];
10    Mat warp_mat(2, 3, CV_32FC1);
11    Mat warp_dst;
12    warp_dst = Mat::zeros(src.rows, src.cols, src.type()); srcTri[0] = Point2f(0, 0);
13    srcTri[1] = Point2f(src.cols - 1.0f, 0);
14    srcTri[2] = Point2f(0, src.rows - 1.0f);
15    dstTri[0] = Point2f(src.cols * 0.0f, src.rows * 0.33f); dstTri[1] = Point2f(src.cols *
16    imshow("src", src);
17    imshow("dst", warp_dst);
18    waitKey(0);
19    return 1;
20 }
21

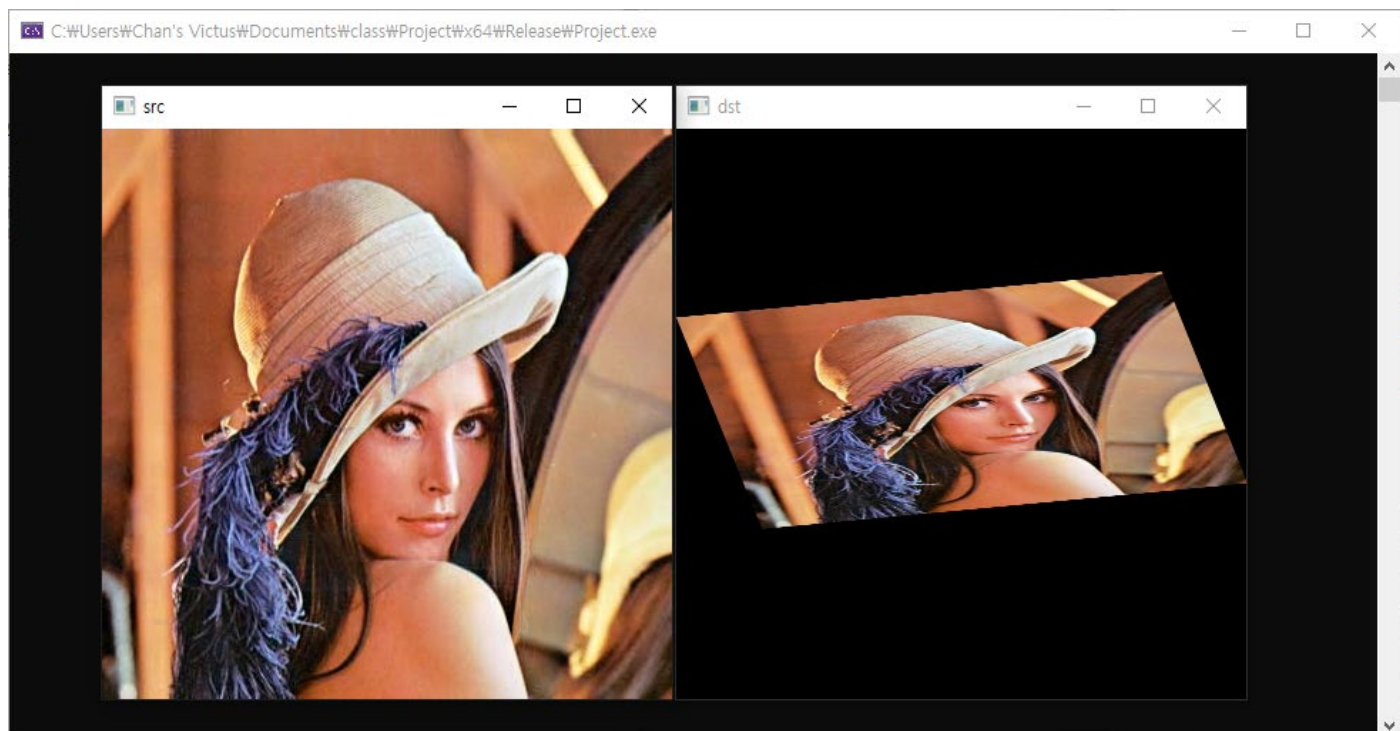
```

Output Console (출력):

```

1> 1 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
1> "Project.vcxproj" 프로젝트를 빌드했습니다.
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 8:31에 시작되었고 02.323 초이(가) 소요됨 =====

```



11-기하학적 변환 23page

Visual Studio Code interface showing the C++ code for image warping using OpenCV.

```

1 #include <opencv2/opencv.hpp>
2 using namespace cv;
3 using namespace std;
4
5 int main()
6 {
7     Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/book.jpg");
8     Point2f inputp[4];
9     inputp[0] = Point2f(30, 81);
10    inputp[1] = Point2f(274, 247);
11    inputp[2] = Point2f(298, 40);
12    inputp[3] = Point2f(598, 138);
13    Point2f outputp[4];
14    outputp[0] = Point2f(0, 0);
15    outputp[1] = Point2f(0, src.rows);
16    outputp[2] = Point2f(src.cols, 0);
17    outputp[3] = Point2f(src.cols, src.rows);
18    Mat h = getPerspectiveTransform(inputp, outputp);
19    Mat out;
20    warpPerspective(src, out, h, src.size());
21    imshow("Source Image", src);
22    imshow("Warped Source Image", out);
23    waitKey(0);
24 }

```

Build Output:

```

1> 0 functions were new in current compilation
1> 1 functions had inline decision re-evaluated but remain unchanged
1> 코드를 생성했습니다.
1> Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
===== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뛴 =====
===== 빌드이(가) 오후 8:33에 시작되었고 01.038 초이(가) 소요됨 =====

```

