

# 영상처리 실제 - 11주차 실습

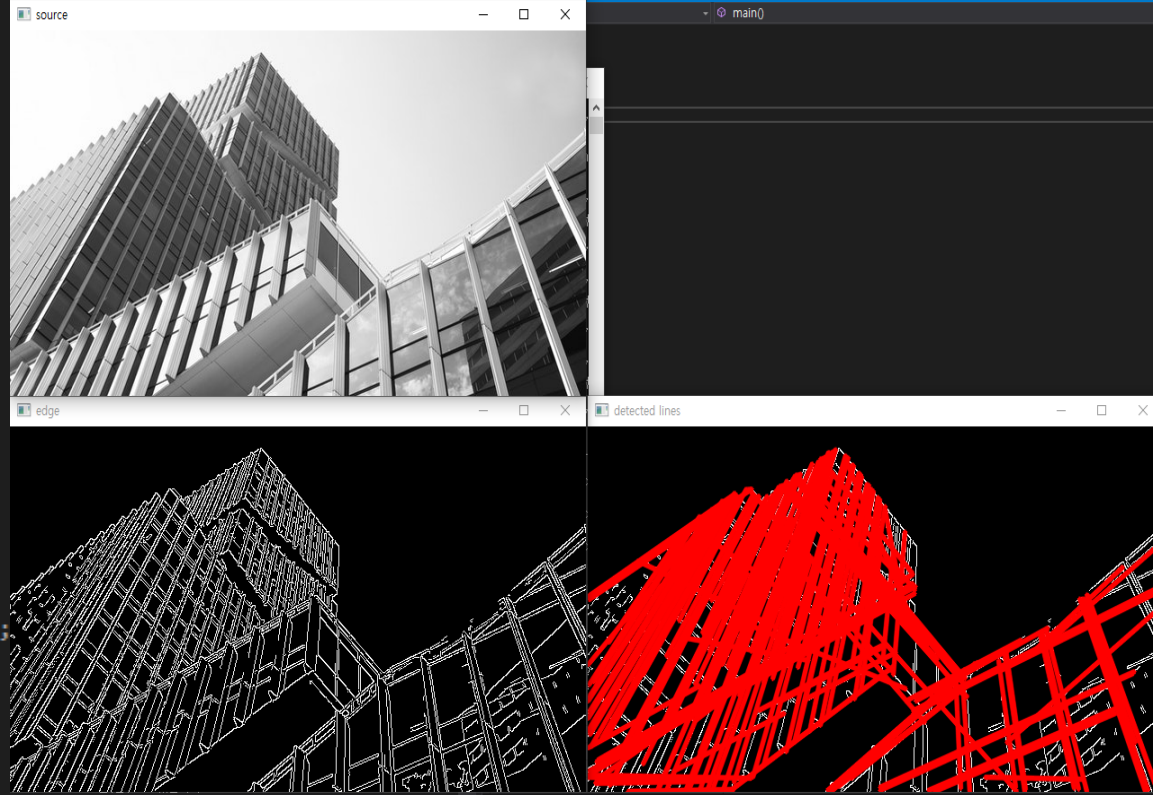
## : 16.특징추출(1) – p.22 ~ 23

```
//16 - 특징추출 - p.22
#ifdef 1
Mat src = imread("D:\\999.Image\\building.jpg", 0);
if (src.empty())
{
    cout << "can not open " << endl;
    return -1;
}

Mat dst, cdst;
Canny(src, dst, 100, 200);
imshow("edge", dst);
cvtColor(dst, cdst, COLOR_GRAY2BGR);

vector<Vec4i> lines;
HoughLinesP(dst, lines, 1, CV_PI / 180, 50, 100, 20);
for (size_t i = 0; i < lines.size(); i++)
{
    Vec4i l = lines[i];
    line(cdst, Point(l[0], l[1]), Point(l[2], l[3]), Scalar(0, 0, 255), 3, LINE_AA);
}

imshow("source", src);
imshow("detected lines", cdst);
waitKey();
#endif
```



## : 16.특징추출(1) – p.27 ~ 28

```
//16 - 특징추출 - p.27
#ifdef 1
Mat src, src_gray;

src = imread("D:\\999.Image\\plates.jpg", 1);
imshow("src", src);
// 그레이스케일로 변환한다.
cvtColor(src, src_gray, COLOR_BGR2GRAY);

// 가우시안 블러링 적용
GaussianBlur(src_gray, src_gray, Size(9, 9), 2, 2);

vector<Vec3f> circles;

// 원을 검출하는 허프 변환
HoughCircles(src_gray, circles, HOUGH_GRADIENT, 1, src_gray.rows / 8, 200, 50, 0, 0);

// 원을 영상 위에 그린다.
for (size_t i = 0; i < circles.size(); i++)
{
    Point center(cvRound(circles[i][0]), cvRound(circles[i][1]));
    int radius = cvRound(circles[i][2]);
    circle(src, center, 3, Scalar(0, 255, 0), -1, 8, 0); // 원의 중심을 그린다
    circle(src, center, radius, Scalar(0, 0, 255), 3, 8, 0); // 원을 그린다.
}

imshow("Hough Circle Transform", src);
waitKey();
#endif
```

