

영상처리

과제 HW#5

학과 : 소프트웨어학과

학번 : 2016039016

이름 : 유수형



HW#5-1. Edge Detector

문제 정의 : 아래와 같은 영상을 Sobel 연산자, Laplacian 연산자, Marr-Hildreth 에지 검출 알고리즘을 사용하여 edge를 검출하고, 각각의 결과와 어떻게 다른지 비교하시오.

```
Mat dstImageGy;
normalize(abs(dstGy), dstImageGy, 0, 255, NORM_MINMAX, dtype); // normalize에 대한 dstGy절대값
threshold(abs(dstGy), dstImageGy, thresh, 255, THRESH_BINARY);
imshow("dstImageGy", dstImageGy);

Mat dstMag;
magnitude(dstGx, dstGy, dstMag); // dstGx, dstGy 절대값에 대한 합
Mat dstImageGxy;
normalize(dstMag, dstImageGxy, 0, 255, NORM_MINMAX, dtype);
threshold(dstMag, dstImageGxy, thresh, 255, THRESH_BINARY);
imshow("dstImageGxy", dstImageGxy);
```

- ✓ dstImageGy와 같이 dstImageGx도 구한 절대 값의 합을 통해 Gxy의 결과를 얻을 수 있다. 수직에 대한 edge와 수평에 대한 edge의 합이라고 볼 수 있다.

```
Mat lapImage0; //Laplacian 연산자
Laplacian(srcImage, lapImage0, CV_32F, ksize);
imshow("Laplacian", lapImage0); //Laplacian 연산자만 적용한 결과
```

```
Mat blurImage;
GaussianBlur(srcImage, blurImage, Size(ksize, ksize), 0.0); // srcImage를 15 x 15의 가우시안 필터로 스무딩

Mat lapImage;
Laplacian(blurImage, lapImage, CV_32F, ksize); // 가우시안 스무딩 된 영상에 대해 라플라시안 필터링 수행

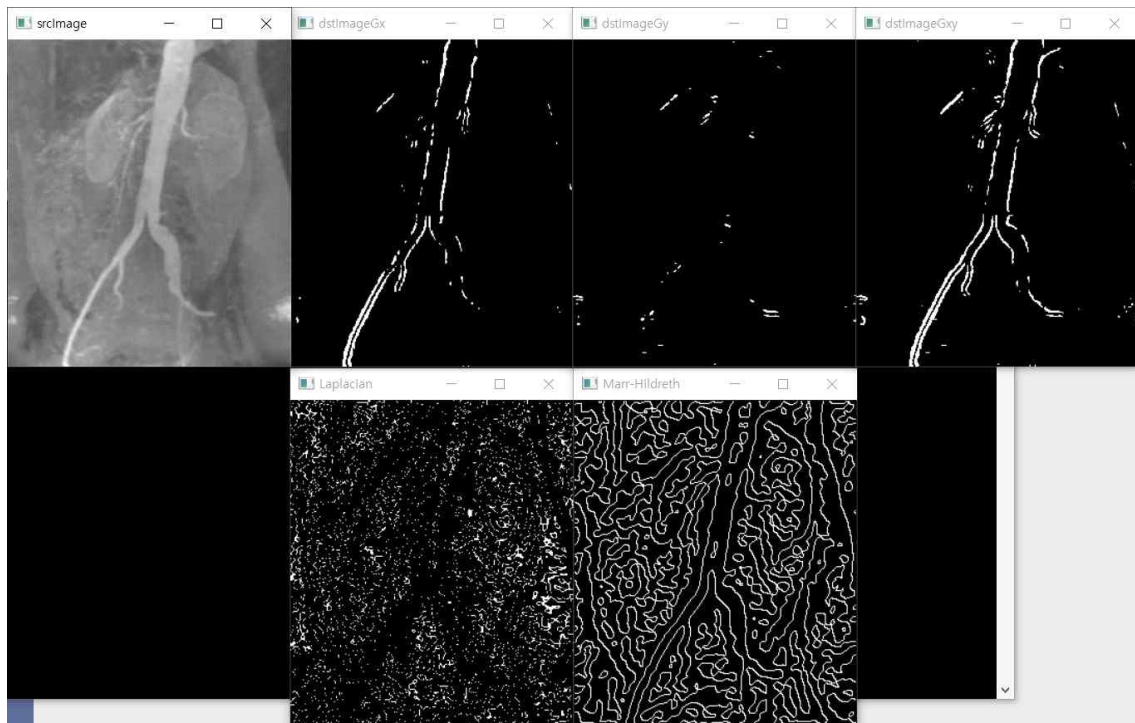
Mat dstImage;
ZeroCrossing(lapImage, dstImage, 10); // 라플라시안 결과에 대해 영교차점 체크

imshow("Marr-Hildreth", dstImage); //Marr-Hildreth 에지 검출 알고리즘을 적용한 결과
```

- >> 먼저 laplacian연산자를 통해 영상에 대해 적용시키고 lapImage0로 생성시킨다.
가우시안 필터로 blurImage를 생성하고 Laplacian의 결과에 대해 영교차점을 확인

```
Mat zeroCross(src.size(), CV_32F, Scalar::all(0));
add(zeroCrossH, zeroCrossV, zeroCross); // zeroCross = zeroCrossH + zeroCrossV
threshold(zeroCross, dst, th, 255, THRESH_BINARY);
```

- ✓ 0교차점 확인하는 과정에서 0교차점은 zeroCrossH+zeroCrossV. 가로, 세로방향에 대한 이웃을 조사하고 부호가 결정되는 0교차 위치를 검출하는 방식이고 이 차이에 대한 절대값을 통해 부호 변경 위치를 edge로 변경하여 반영



HW#5-2. 대각선 검출

아래와 같은 영상에 대하여 45° , -45° 방향의 대각선 edge를 검출하고자 한다.

Sobel 연산자를 변형하여 적절한 마스크를 만들어 작용해 보시오.

```
double maskdata[3][3] = { // 45도 마스크 생성
    {2, -1, -1},
    {-1, 2, -1},
    {-1, -1, 2}
};
```

```
double maskdata1[3][3] = { // -45도 마스크 생성
    {-1, -1, 2},
    {-1, 2, -1},
    {2, -1, -1}
};
```

sobel 연산자 변형을 위해 45° , -45° 방향의 대각선 edge검출을 위한 마스크를 생성한다.

```
filter2D(srcImage, dstImage1, -1, mask);
imshow("45 degree", dstImage1);
```

생성한 마스크에 대해서 filter2D를 이용하여 필터링한다. (-45° 도 같은 방법으로 필터링)

