|  |  |
| --- | --- |
| Baboo.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |
| boats.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |

|  |  |
| --- | --- |
| Boats\_512\_high.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |
| lena.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |
| Lena\_512\_low.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |
| Lena\_512\_noise.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |

|  |  |
| --- | --- |
| pepper.png<결과영상> | |
| 임계값  128 |  |
| 임계값  50 |  |
| 임계값  200 |  |

|  |  |
| --- | --- |
| Lena\_256.png | |
| Threshold value:128 |  |
| Threshold value:50 |  |
| Threshold value:200 |  |

<결과 토의>

일반적으로 threshold value가 크면 edge는 적어지고 threshold value가 작으면 edge는 많아진다. 소벨에서는 x 필터(수직 마스크)와 y 필터 즉(수평 마스크)의 절댓값의 합을 구한 gradient로 가장자리를 인식하게 된다. 즉, 이전값과 현재값 픽셀의 차이, 수직 마스크와 수평 마스크의 절댓값의 합의 크기가 threshold value 보다 크면 255 가 되고 작으면 0 이 되는 것이다. Boats\_512\_high.png를 보면 high frequency 성분만 있으므로 수직, 수평 픽셀들의 차이가 별로 없으므로 임계값이 50 일때가 edge를 200,128 일때보다 잘 추출해준다. lena\_512\_low.png를 보면 low frequency 성분만 있으므로 수직,수평 픽셀들의 차이가 별로 없으므로 임계값이 50 일때가 200,128보다 edge를 더 잘 추출해준다. lena\_512\_noise.png를 보면 noise는 high frequency 성분이므로 임계값이 200 일 때가 noise 성분을 잘 차단해 주는 것을 볼 수 있다. 왜냐하면 noise는 high freqeuency이므로 edge가 될 가능성이 높다. 따라서 threshold가 클수록 edge를 줄여주기 때문에, high frequency인 noise가 제거 됨을 볼 수 있다. 따라서 소벨은 잡음에 강한 편 인 것을 알 수 있다. baboo.png와 boats.png는 high frequency가 많으므로 threshold value가 200일 때도 다른 이미지에 비해서 edge가 추출됨을 알 수 있다. 왜냐하면 high frequency 성분이 많고 low frequency 성분이 적으므로 threshold value값이 200이어도, high frequency 성분이 많아서, edge로 인식 할 수 있다. lena.png와 pepper.png는 low frequency와 high frequency가 골고루 있으므로 threshold value 값이 작을 수록 경계선의 굵기가 굵어 지며 외곽선이 많아지는 것을 볼 수 있다. threshold value 값이 커지면 즉 200일 때 경계선이 얇아지며 외곽선이 적어지는 것을 볼 수 있다. threshold value 값에 따른 보여지는 일반적인 영상 이미지 결과를 정리하자면, Threshold value 값이 작을수록, 영상 외곽선이 뚜렷해 지지며 에지가 많아진다. 이는 이상적으로 필요한 에지 보다 더 많이 검출 될 수 있다. 또한 검출한 에지가 필요 이상으로 두꺼워져 객체를 휠씬 더 식별 하기 어렵게 만들 수 있다. 높은 경계 값의 경우는 모든 중요한 외곽선에 속하는 에지를 정의하는 것으로 즉, 이상 치로 간주되는 모든 에지를 배제한다. 하지만 흐릿하지만 장면의 중요한 외곽선에는 확실히 에지가 보인다. 즉, Sobel 낮은 경계 값 영상은 에지를 잘 검출한 듯하지만 진해지지만, 반대로 Sobel 높은 경계 값 영상은 에지를 정확히 찾지 못했다**.** 따라서 소벨 가지고는 영상의 edge를 잘 추출하기가 어려우므로 다른 추가적인 연산이 필요함을 알 수 있다. lena\_256.png는 low frequency가 high frequency 보다 많으므로 lena.png에 비해서 경계선이 굵다는 것을 알 수 있다. 왜냐하면 low frequency가 많아서 high frequency랑 차이가 많이 나므로, edge가 많이 발생하기 때문에 threshold value가 200 일 때도 lena.png에 비해서 edge가 많은 것을 알 수 있다.

<프로그램 코드>

echo off;

clear all;

i=imread('pepper.png');

h=fspecial('sobel'); //가로 방향을 강조는 3 by 3, sobel 필터를 생성, 수평 마스크

h\_t=h'; // 세로 방향을 강조는 3 by 3을 얻기 위해서 전치함, 수직 마스크

b1=imfilter(i,h\_t); //h\_t를 이용해서 i를 필터링 함(세로 방향 필터)

b2=imfilter(i,h); //h를 이용해서 i를 필터링 함 (가로 방향 필터)

b1=abs(b1); // b1의 절댓값을 구함

b2=abs(b2); //b2의 절댓값을 구함

gradient=b1+b2; //b1+b2를 통해 gradient의 크기를 구함

[row,col]=size(gradient); // gradient의 row,col 크기를 구함

for x=1:row

for y=1:col

if gradient(x,y)>128

thr\_img(x,y)=255;

else

thr\_img(x,y)=0;

end

end

end

// 행,열 방향으로 gradient를 threshold value에 따라서 edge를 검출함

thr\_img=uint8(thr\_img);

// 255,0으로 이루어진 thr\_img를 8비트의 부호 없는 정수형으로 바꿈

figure;

subplot(121); imshow(i,[0 255]); title('원본 이미지');

//원본이미지를 보여줌

subplot(122); imshow(thr\_img,[0 255]); title('edge detection한 이미지');

//edge detection한 이미지를 보여줌

설명:

Fspecial 함수를 통해 sobel 3 by 3 수평 필터를 얻고 이를 전치하여 수직 필터를 얻는다. 각각의 2개의 필터 마스크를 통해서 읽어온 이미지를 imfilter 함수를 사용해서 필터링 한 후, 구 한 값을 각각 절댓값을 취하여 더 해서 gradient 크기를 구한다. gradient의 row, col 크기 만큼 threshold value에 따라서 edge를 판별 한 후, 이를 uint8를 통해서 부호 없는 8비트의 정수형으로 만들어서 sobel을 이용한 edge detection 한 영상을 얻는다.