HW 03 – REPORT

소속 : 정보컴퓨터공학부

학번 : 202055536

이름 : 민예진

1. 서론

1. 실습 목표

Canny Edge Detection의 과정을 직접 구현하고 이를 통해 이미지의 edge를 추출하는 것을 목표로 한다.

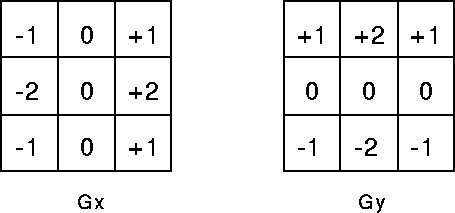
1. 이론적 배경 기술
   1. Edge Detection

평면, depth, 색깔, 밝기 등의 요소가 급격하게 바뀌는 지점을 추적하면 edge를 찾을 수 있다. 급격하게 바뀌는 지점은 변화량을 통해 감지할 수 있으며 2D 이미지의 edge를 구하기 위해 Image derivative filter를 사용하여 변화량을 구할 수 있다.

* 1. Canny Edge Detection
     1. Smoothing – Noise 제거

이미지에 noise가 있게 되면 edge 부근에서 보이는 급격한 변화량을 감지하기 어렵다. Noise를 제거하기 위해 Gaussian Filter를 사용하여 Blurring을 한다.

* + 1. Image Gradient, Orientation 값 얻기



Sobel convolution kernel

* + - * + Sobel convolution kernel : edge를 추출하기 위해 사용하는 필터

**Image Gradient, Orientation 수식**

Sobel convolution kernel을 사용하여 x축, y축 방향의 변화량을 얻는다. 이 값을 바탕으로 Image Gradient, Orientation을 위 식을 이용하여 얻는다.

* + 1. Non-maximum suppression

Sobel convolution kernel로 얻은 Edge는 여러 개의 픽셀로 표현이 되어 얇은 edge를 얻는 과정이 필요하다. Gradient image의 orientation을 이용하여 orientation 방향의 edge중 max값을 edge로 추출한다. 이 과정에서 orientation 방향의 점을 얻기 위해 interpolation하는 과정이 필요할 수도 있다.

* + 1. Thresholding edges and lining (hysteresis)
       1. Double thresholding

위 과정에서 얻는 edge 중 strong edge를 얻기 위해 2개의 임계값을 정한 후 strong edge와 weak edge로 구분한다. 픽셀의 값을 R이라 하고 임계값 중 큰 값을 T, 작은 값을 t라 할 때 Strong edge는 R > T를 만족하는 edge이고 weak edge는 t < R < T 를 만족하는 edge이다. R < t 인 경우는 edge에서 제외한다.

* + - 1. Hysteresis

strong edge와 연결된 weak edge를 찾는다. 연결되지 않은 edge는 edge에서 제외한다.

2. 본론

1. Noise reduction

HW2의 gaussconvolve2d 함수를 이용하여 원본 grayscale 이미지를 blurring한다.

**실행결과**

|  |  |
| --- | --- |
|  |  |
| 원본 이미지 | 가우시안 blurring 진행 |

1. Finding the intensity gradient of the image

Sobel filter 원본 이미지와 convolution 하여 x방향 이미지와 y방향 이미지를 얻은 후, 이 결과를 통해 서론의 수식을 이용하여 Image gradient와 theta 값을 얻는다. 이때 gradient는 numpy의 hypot을 이용하고 theta는 numpy의 arctan2를 이용한다.

**실행결과**



1. Non-Maximum Suppression
   1. angle 변환

numpy의 degrees 함수를 통해 theta의 radian 값을 degree 값으로 바꾼다.

* 1. 0, 45, 90, 135도를 기준으로 gradient의 max값 구하기

|  |  |
| --- | --- |
|  |  |
| 좌표 기준 | 위 사진의 좌표를 기준으로 각도당 이웃 좌표를 나타낸 수식 |

0도, 45도, 90도, 135도를 기준으로 이웃한 점을 구한 후, 이웃 점과 현재 gradient의 최댓값을 구해 edge를 구한다.

**실행결과**



1. Double threshold

임계값 T\_high, T\_low를 통해 strong edge에는 255, weak edge에는 80의 값을 부여한다.

**실행결과**



1. Edge Tracking by hysteresis

dfs를 통해 strong edge와 연결된 weak edge를 찾은 후 80의 값을 255의 값으로 바꾸고 strong edge와 연결되지 않은 weak edge는 0의 값을 바꾼다.

**실행결과**



3. 결론

Canny Edge Detection의 Finding the intensity gradient of the image, Non-Maximum Suppression, Double threshold, Edge Tracking by hysteresis과 같은 4가지의 세부 과정을 직접 구현해보았다. 이 과정을 통해 섬세한 edge를 얻는 과정을 직접 코드로 표현할 수 있는 시간이었다.

Canny Edge Detection 이외의 또 다른 edge detection 방식에 대해 알아보자

**Sobel Edge**

해당 실습에서 Canny Edge Detection에서 x축 방향과 y축 방향의 edge를 검출하기 위해 사용한 필터로 x, y 모든 방향의 edge를 추출하는 방식이다. 잡음에 강한 편이나 intensity 변화 구간이 촘촘하거나 복잡한 영상일 경우 edge detection의 효과가 떨어진다.

**Prewitt Edge**

텍스트이(가) 표시된 사진

자동 생성된 설명

**Prewitt Operator – 위키백과**

Sobel filter와 결과가 비슷한 방식으로 응답시간이 다소 빠르다. 하지만 Sobel filter에 비해 밝기 변화에 대한 weight를 적게 부여하여 edge가 덜 부각된다. 대각선 방향의 edge보다 수직, 수평 방향의 edge에 더 민감하게 반응한다.

**Robert Edge**

**텍스트, 시계이(가) 표시된 사진

자동 생성된 설명**

**Robert Operator – 위키백과**

Sobel, Prewitt filter에 비해 매우 빠른 계산 속도를 보여주며 edge detection에 좋은 효과를 보인다. 하지만 Sobel, Prewitt filter에 비해 edge가 훨씬 가늘며, noise에 매우 민감하다.