

## 숙제 #2 (40점)

### Edge detection + Boundary tracing (DP)

EEE6478 컴퓨터 비전

120180048 김태훈

\* Python을 이용하여 실험을 진행하였으며, 각 영상은 모두 512 x 512 pixel의 흑백 영상이다.

a) 교재에 있는 Algorithm 6.4와 Algorithm 6.5를 구현하여 세 가지 영상에 대하여 실험을 진행하였다. 각 Algorithm의 구현에 앞서 에지 검출을 위하여 Sobel 연산자를 사용하였다.

-1	0	+1
-2	0	+2
-1	0	+1

x filter

+1	+2	+1
0	0	0
-1	-2	-1

y filter

X축을 기준으로 하는 Sobel 연산자와 Y축을 기준으로 하는 Sobel 연산자를 각 pixel을 중심으로 하는 3x3 행렬에 곱하여 각 방향의 변화량을 측정한 뒤, 이들의 절대 값의 합을 구하면 해당 pixel의 에지 크기(edge magnitude)를 구할 수 있다.

단, Sobel 연산 만으로는 에지 방향(edge direction)을 알 수 없기 때문에 이를 위하여 추가적인 연산을 수행하였다. X filter의 연산 결과를  $G_x$ , Y filter의 연산 결과를  $G_y$ 라고 할 때, 해당 에지의 기울기의 방향은  $\theta = \text{atan}\left(\frac{G_y}{G_x}\right)$  를 통해 각도로서 표현할 수 있다. 이를 이용하여  $-67.5^\circ$  이하나

$67.5^\circ$  이상은 수직 방향,  $27.5^\circ \sim 67.5^\circ$ 는  $y=x$  방향,  $0^\circ \sim 27.5^\circ$ 는 수평 방향,  $-27.5^\circ \sim -67.5^\circ$ 는  $y = -x$  방향으로 설정하여 Algorithm 6.4의 step 1에서 필요한 8-connectivity에 따른 edge direction을 계산하였다. Canny 에지 검출을 고안한 Canny에 따르면 Algorithm 6.5의 threshold 값 선정 시 high threshold와 low threshold의 비율은 2:1에서 3:1 정도가 적당하다고 한다. 이를 토대로 여러 threshold에 대하여 반복적으로 실험한 결과, low threshold 30, high threshold 90으로 설정하였을 때 세 영상 모두 최선의 결과를 얻을 수 있었다.

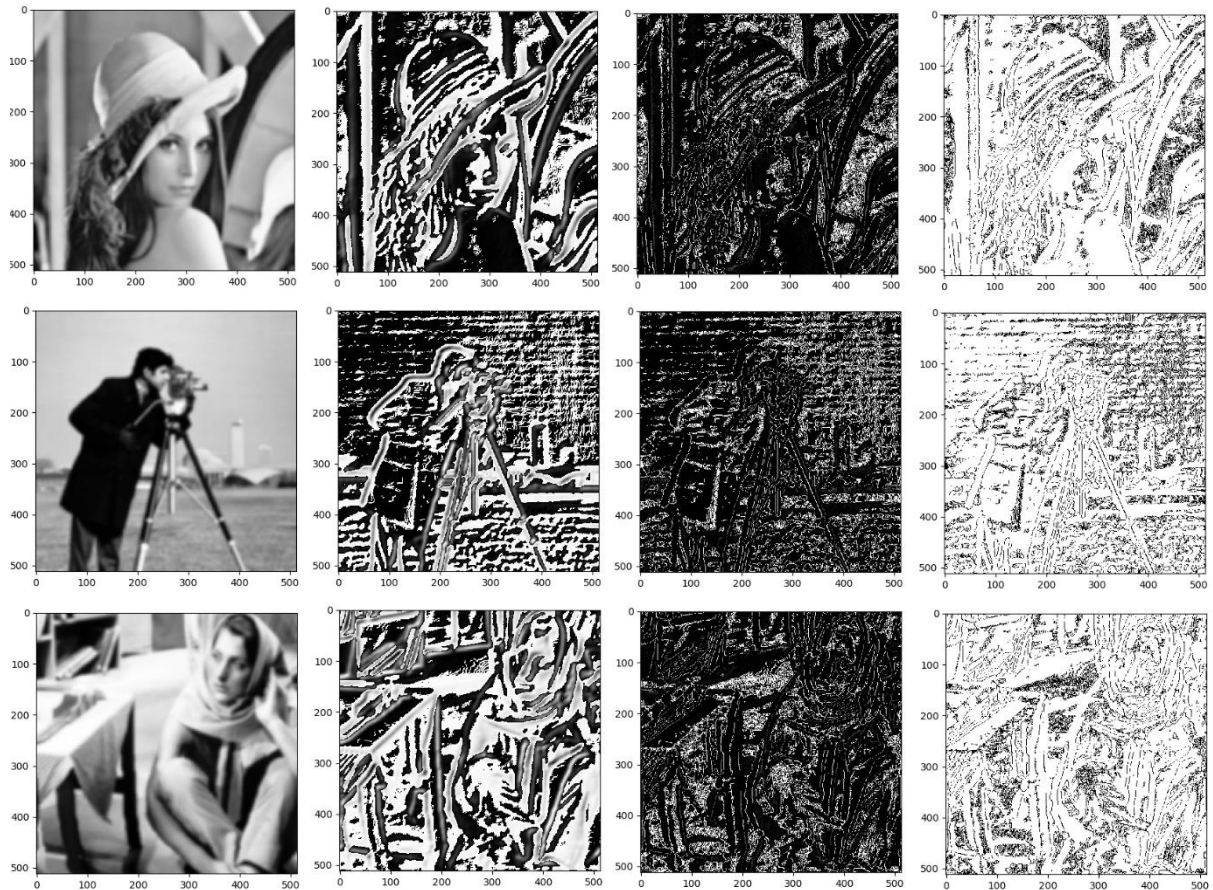
그러나 세 영상 모두 평탄한 부분을 찾아보기 힘들어 영상 원본을 그대로 사용한 경우 불필요한 에지가 너무 많이 검출되는 문제가 발생하였다. 이를 해결하기 위하여 각 영상에 sigma 3의 가우시안 필터를 적용하여 영상을 전처리 한 뒤 에지 검출을 수행하였다. 각 영상에 대한 실험 결과는 다음과 같다.

\* 각 영상 별 에지 개수/비율

영상 1: 40374개(15.4%)

영상 2: 46968개(17.9%)

영상 3: 44849개(17.1%)



**b)** Algorithm 6.12의 구현을 Algorithm 6.5의 결과에 적용하기 위해서는 몇몇 파라미터에 대한 정의가 필수적이다.

에지 graph의 구축을 위해서는 각 에지의 magnitude와 direction에 대한 정의가 이루어져야 한다. 실험에는 a)에 서술한 대로 sobel 연산자를 이용하여 구한 magnitude와 direction을 사용하였다. 이를 토대로 에지 direction에 따라 모든 에지를 연결한 graph를 구축하였다.

Graph를 구축하는 과정에서 에지를 무분별하게 연결할 경우 loop가 발생하고 graph가 너무 복잡해질 가능성이 있다. 이를 막기 위하여 각 node를 연결할 시에는 에지 direction을 중심 축으로 하여  $-45^{\circ} \sim 45^{\circ}$  범위 내에 존재하는 에지 만을 채택하여 연결하였다.

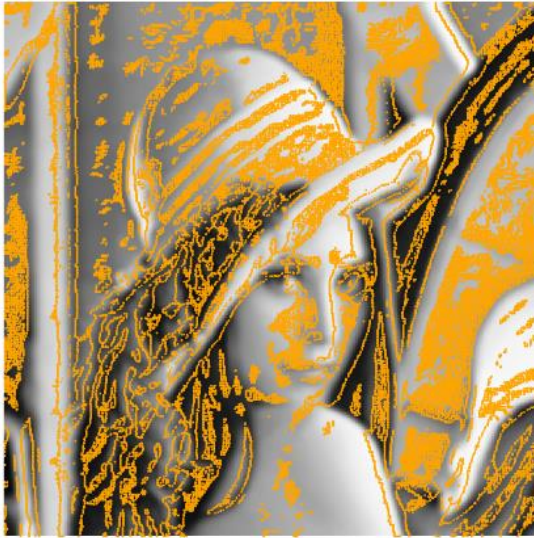
graph node 별 partial path cost로는 실제 pixel 간 L2 norm을 이용하였다. 또한 첫번째 graph 계층의 initial cost는 각 node의 partial path cost로 설정하였다.

Graph 계층의 깊이가 너무 깊어지면 연산 시간이 너무 오래 걸리기 때문에 Graph의 최대 계층수는 500으로 설정하였다. 또한 border가 너무 복잡해지는 것을 막기 위하여 loop가 발생할 시 해당 node는 조기에 탐색을 중단시키는 pruning을 수행하였다.

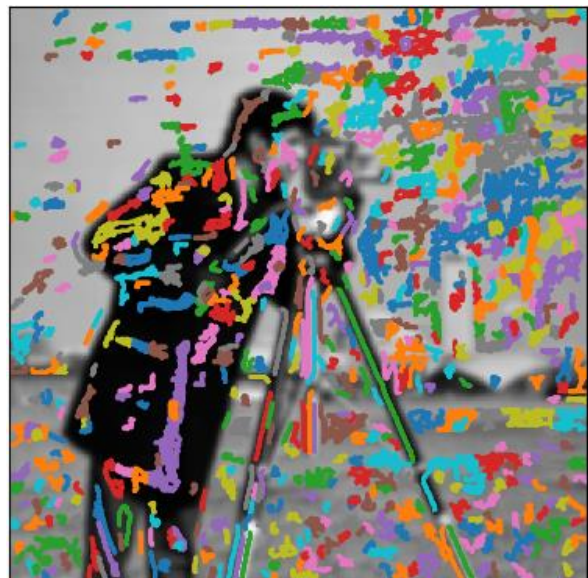
최종 결과로 나온 다수의 optimal path 중에서 길이가 100 이하인 path는 무시하였다. 이는 영상의 실제 border가 아닌, 잡음에 의해 발생한 path를 무시하기 위함이다.

이와 같은 전제 조건을 바탕으로 Algorithm 6.5를 통해 정제된 에지에 대한 boundary tracing 결과는 다음과 같다.

[영상 1]

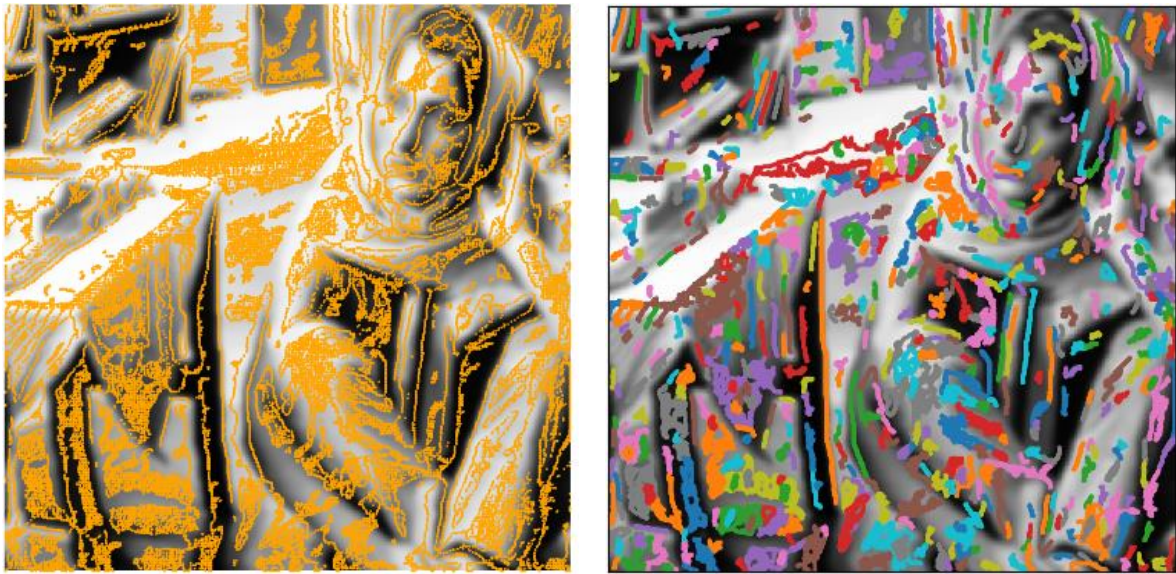


[영상 2]





[영상 3]



Algorithm 6.5를 활용하여 불필요한 에지를 최대한 줄이고, 가우시안 필터를 적용하여 영상의 잡음을 최대한 줄이고자 하였음에도 불구하고 불필요한 border가 많이 검출되었다. 이는 원본 영상 자체가 너무 복잡하기 때문인 것으로 보인다.