

에지 검출과 응용

에지 검출

에지는 픽셀 값이 급격하게 변경되는 부분이므로, 픽셀 값의 변화율을 이용하여 에지를 찾을 수 있다. 영상에서 에지(edge)는 한쪽 방향으로 픽셀 값이 급격하게 바뀌는 부분을 가리킨다. 즉, 어두운 영역에서 갑자기 밝아지거나 또는 반대로 밝은 영역에서 급격하게 어두워지는 부분을 에지라고 한다. 일반적으로 객체와 배경의 경계, 또는 객체와 다른 객체의 경계에서 에지가 발생한다. 그러므로 영상에서 에지를 찾아내는 작업은 객체의 윤곽을 알아낼 수 있는 유용한 방법이며 다양한 컴퓨터 비전 시스템에서 객체 판별을 위한 전처리로 에지 검출이 사용되고 있다.

기본적으로 영상에서 에지를 찾아내려면 픽셀 값의 변화율을 측정하여 변화율이 큰 픽셀을 선택해야 합니다. 수학에서 함수 또는 데이터의 변화율을 미분이라고 한다. 정확하게 기술하면 함수의 미분이란 주어진 함수의 순간 변화율을 의미하며, 1차원 연속 함수 $f(x)$ 의 미분은 다음과 같이 정의합니다.

$$f' = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

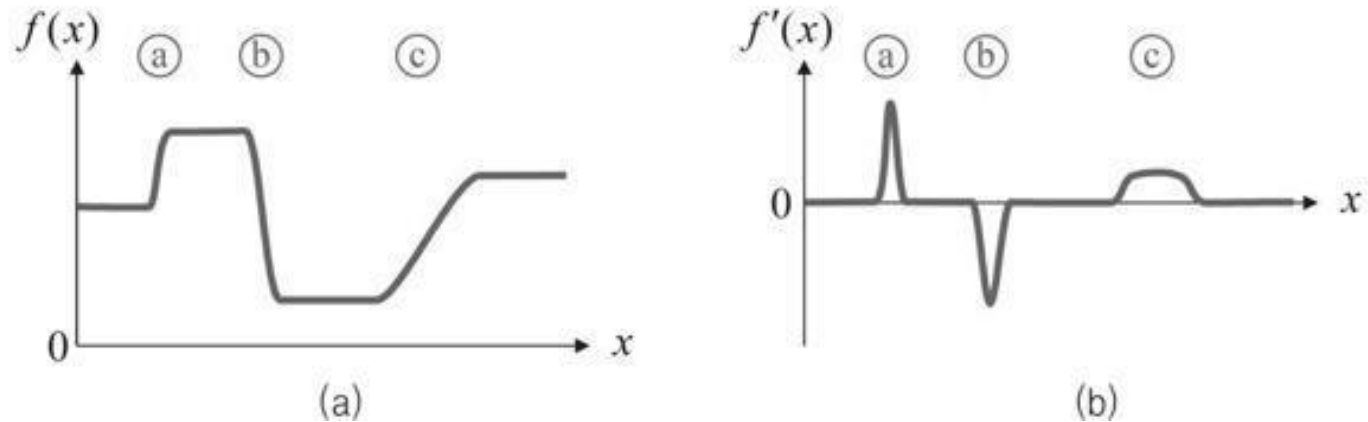
함수 값이 증가하는 위치 : 미분값이 0보다 큰 양수로 나타남

함수 값이 감소하는 위치 : 미분값이 0보다 작은 음수로 나타남

함수값이 일정한 구간 : 미분값이 0에 가까운 값을 가짐

에지 검출

그림은 1차원 연속 함수 $f(x)$ 의 값 변화에 따른 미분 $f'(x)$ 를 나타낸 것이다. 함수 $f(x)$ 값이 급격하게 바뀌는 부분을 찾기 위해서는 함수의 미분 $f'(x)$ 값이 0보다 훨씬 크거나 또는 훨씬 작은 위치를 찾아야 한다.



영상은 2차원 평면 위에 픽셀 값이 정형화되지 않은 상태로 나열되어 있는 형태이므로 미분 공식을 적용할 수 없다. 영상으로부터 미분을 계산하려면 두 가지 특징을 고려해야 한다. 하나는 영상이 2차원 평면에서 정의된 함수라는 점이고, 두 번째는 영상이 정수 단위 좌표에 픽셀이 나열되어 있는 이산함수라는 점이다.

에지 검출

영상과 같이 일련의 데이터가 순서대로 나열되어 있는 경우에는 미분 근사화 방법을 이용하여 변화량을 측정할 수 있다. 미분 근사는 다음 세 가지 방법을 주로 사용한다.

- **전진 차분**(forward difference) $\frac{dI}{dx} \cong \frac{I(x+h) - I(x)}{h}$ → 자기 바로 앞에 있는 픽셀에서 자기 픽셀 값을 뺀 형태
- **후진 차분**(backward difference) $\frac{dI}{dx} \cong \frac{I(x) - I(x-h)}{h}$ → 자기 픽셀에서 바로 뒤에 있는 픽셀 값을 뺀 형태
- **중앙 차분**(centered difference) $\frac{dI}{dx} \cong \frac{I(x+h) - I(x-h)}{2h}$ → 자신을 제외하고 바로 앞과 뒤에 있는 픽셀 값을 이용

세 가지 미분 근사 방법 중 **중간값 차이**를 이용하는 방법이 이론적으로 근사화 오류가 가장 적으며, 실제 영상에서 미분을 계산할 때에도 널리 사용된다.

에지 검출

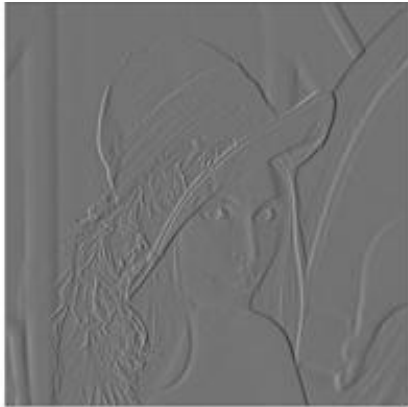
영상에서 에지를 찾기 위해서는 영상을 가로 방향과 세로 방향으로 각각 미분해야 한다. 2차원 영상 $I(x,y)$ 에 대하여 x 축과 y 축 방향에 대한 각각의 편미분을 중앙 차분 방법으로 근사화하면 다음과 같다.

$$I_x = \frac{\partial I}{\partial x} \cong \frac{I(x+1, y) - I(x-1, y)}{2}$$

$$I_y = \frac{\partial I}{\partial y} \cong \frac{I(x, y+1) - I(x, y-1)}{2}$$



(a)



(b)



(c)

→ 회색 영역은 픽셀 값 변화가 작은 영역이고, 흰색 또는 검은색으로 표현된 픽셀은 픽셀 값이 급격하게 바뀌는 부분이다.

실습1

에지 검출

2차원 공간에서 정의된 함수 $f(x,y)$ 의 x축 방향 미분과 y축 방향 미분을 한꺼번에 벡터로 표현한 것을 그래디언트라고 하고, 다음과 같이 표기한다.

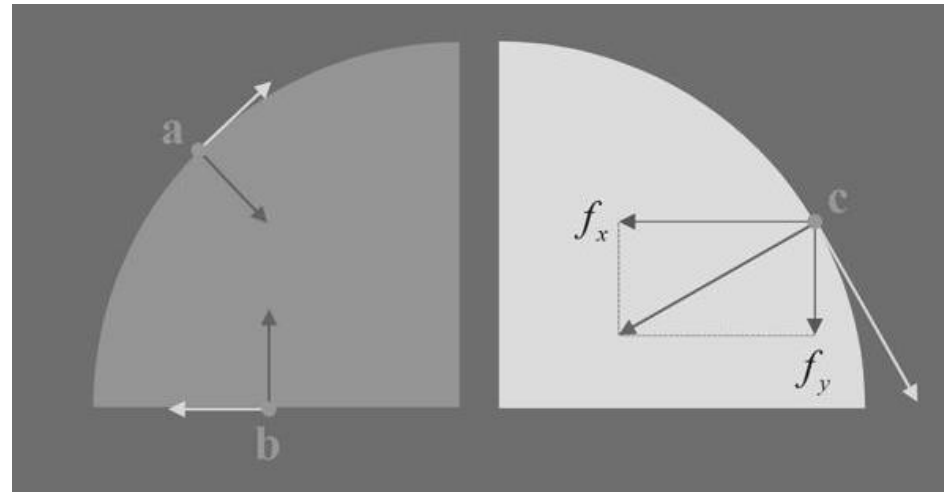
$$\nabla f = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = f_x \mathbf{i} + f_y \mathbf{j}$$

그래디언트는 벡터이기 때문에 크기와 방향 성분으로 표현할 수 있다. 그래디언트 벡터의 방향은 변화 정도가 가장 큰 방향을 나타내고, 크기는 변화율 세기를 나타내는 척도로 생각할 수 있다.

에지 검출

아래 그림은 객체와 배경 경계상의 세 점 a, b, c를 선택하고, 각 점에서의 그래디언트 벡터를 화살표로 나타낸 것이다. 화살표의 길이는 그래디언트의 크기를 나타내고, 화살표의 방향은 그래디언트 벡터의 방향을 나타낸다.

- 벡터의 크기는 밝기 차이가 클 수록 크게 나타나므로 점 a, b의 화살표보다 점 c에서 화살표 길이가 길게 나타난다.
- 벡터의 방향은 해당 위치에서 밝기가 가장 밝아지는 방향을 가리킨다.
- 점 c에서 가장 밝은 화살표는 그래디언트 벡터와 수직인 방향을 표시한 것이며, 이를 에지의 방향이라고 부른다.



마스크 기반 에지 검출

가장 널리 사용되고 있는 미분 마스크는 소벨 필터(Sobel filter) 마스크이다.

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

(a)

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

(b)

(a)는 x축 방향으로의 편미분을 구하는 소벨 마스크, (b)는 y축 방향으로 편미분을 구하는 소벨 마스크이다.

마스크 기반 에지 검출 - 소벨 마스크

OpenCV는 소벨 마스크를 이용하여 영상을 미분하는 Sobel()함수를 제공한다. Sobel()함수는 3x3 소벨 마스크 또는 확장된 형태의 큰 마스크를 이용하여 영상을 미분한다.

```
void Sobel(InputArray src, OutputArray dst, int ddepth,  
           int dx, int dy, int ksize = 3, double scale = 1, double delta = 0,  
           int borderType = BORDER_DEFAULT);
```

Src : 입력 영상

Dst : 출력 영상

Ddepth : 출력 영상의 깊이

Dx : x방향 미분 차수

Dy : y방향 미분 차수

Ksize : 소벨 커널의 크기

Scale : 필터링 연산 후 추가적으로 곱할 값

Delta : 필터링 연산 후 추가적으로 더할 값

borderType : 가장자리 픽셀 확장 방식

마스크 기반 에지 검출 - 소벨 마스크

x방향으로 편미분한 결과를 dx행렬에, y방향으로 편미분한 결과를 dy행렬에 저장하려면 다음과 같이 코드를 작성한다.

```
Mat src = imread("lenna.bmp", IMREAD_GRAYSCALE);
```

```
Mat dx, dy;
```

```
Sobel(src, dx, CV_32FC1, 1, 0);
```

```
Sobel(src, dy, CV_32FC1, 0, 1);
```

마스크 기반 에지 검출 - 샤프 필터 마스크

OpenCV는 널리 사용되고 있는 소벨 마스크 외에도 샤프 필터 마스크를 이용한 미분 연산도 지원한다. 샤프 필터는 소벨 마스크보다 정확한 미분 계산을 수행하는 것으로 알려져 있다.

```
void Scharr(InputArray src, OutputArray dst, int ddepth,  
            int dx, int dy, double scale = 1, double delta = 0,  
            int borderType = BORDER_DEFAULT);
```

Src : 입력 영상

Dst : 출력 영상

Ddepth : 출력 영상의 깊이

Dx : x방향 미분 차수

Dy : y방향 미분 차수

Ksize : 소벨 커널의 크기

Scale : 필터링 연산 후 추가적으로 곱할 값

Delta : 필터링 연산 후 추가적으로 더할 값

borderType : 가장자리 픽셀 확장 방식

마스크 기반 에지 검출 - magnitude() 함수

Sobel() 또는 Scharr() 함수를 이용하여 x 방향으로 미분과 y 방향으로 미분을 각각 계산하여 행렬에 저장한 후, 두 미분 행렬을 이용하여 그래디언트 크기를 계산할 수 있다. OpenCV는 2차원 벡터의 x 방향 좌표와 y 방향 좌표를 이용하여 **벡터의 크기를 계산하는 magnitude() 함수**를 제공한다.

```
void magnitude(InputArray x, InputArray y, OutputArray magnitude);
```

X : 벡터의 x좌표를 나타내는 실수 행렬 또는 벡터

Y : 벡터의 y좌표를 나타내는 실수 행렬 또는 벡터

Magnitude : 벡터의 크기를 나타내는 실수 행렬 또는 벡터

마스크 기반 에지 검출 - phase() 함수

x 방향으로 미분과 y 방향으로 미분이 저장된 두 개의 행렬이 있을 때, 그래디언트의 방향을 계산하고 싶다면 phase() 함수를 사용할 수 있습니다. phase() 함수 원형은 다음과 같습니다.

```
void phase(InputArray x, InputArray y, OutputArray angle, bool angleInDegrees = false);
```

X : 벡터의 x좌표를 나타내는 실수 행렬 또는 벡터

Y : 벡터의 y좌표를 나타내는 실수 행렬 또는 벡터

Angle : 벡터의 방향을 나타내는 실수 행렬 또는 벡터

angleInDegrees : 이 값이 true면 각도(degree) 단위를 사용하고, false이면 라디안(radian)단위를 사용한다.

실습2

Sobel()함수를 사용하여 영상에서 그래디언트를 계산하고, 그래디언트 크기를 이용하여 에지를 검출하는 실습.
그래디언트 크기가 특정 임계값보다 큰 픽셀을 에지로 검출해본다.

`Mat edge = mag > 150;`

→ mag 행렬 원소값이 150보다 크면 255로, 작으면 0으로 설정하여 에지를 판별한다. 임계값은 150.
임계값을 너무 낮추면 잡음의 영향도 에지로 검출될 수 있으므로 주의해야 한다.



캐니 에지 검출기

소벨 마스크 기반 에지 검출 방법은 그래디언트 크기만을 기준으로 에지 픽셀을 검출하기 때문에 임계값에 민감하고, 에지 픽셀이 두껍게 표현되는 문제점이 있다. 1986년 캐니는 에지 검출을 최적화 문제 관점으로 접근함으로써 소벨 에지 검출 방법의 단점을 해결할 수 있는 방법을 제시했다.

소벨 에지 검출 방법이 단순히 그래디언트 크기만을 이용하여 에지를 찾는 방법이라면, 캐니 에지 검출기는 그래디언트의 크기와 방향을 모두 고려하여 더 정확한 에지 위치를 찾을 수 있다. 또한 에지는 서로 연결되어 있는 가능성이 높다는 점을 고려하여 그래디언트 크기가 다소 약하게 나타나는 에지도 놓치지 않고 찾을 수 있다.

캐니 에지 검출기 - 가우시안 필터링

캐니 에지 검출기의 첫 번째 과정은 **가우시안 필터링**이다.

캐니 에지 검출기의 첫 번째 단계에서 가우시안 필터를 적용하는 이유는 **영상에 포함된 잡음을 제거하기 위함이다.** 다만 가우시안 필터링에 의해 영상이 부드러워지면서 에지의 세기도 함께 감소할 수 있기 때문에 적절한 표준편차를 선택하여 가우시안 필터링을 수행해야 하며, 영상에 포함된 잡음이 심하지 않다면 가우시안 필터링은 생략할 수 있다.



캐니 에지 검출기 - 영상 그래디언트 구하기

캐니 에지 검출기의 두 번째 과정은 **영상의 그래디언트**를 구하는 작업이다.

캐니 에지 검출기에서 그래디언트 계산은 보통 3×3 소벨 마스크를 사용한다. 다만 소벨 에지 검출 방법이 오직 그래디언트 크기만을 이용하여 에지를 탐색하였다면, **캐니 에지 검출기는 좀 더 정확한 에지를 찾기 위해 그래디언트 방향도 함께 고려한다.** 그러므로 가로 방향과 세로 방향으로 각각 소벨 마스크 필터링을 수행한 후, 그래디언트 크기와 방향을 모두 계산해야 한다.



캐니 에지 검출기 - 비최대 억제

에지 검출을 위해 단순히 그래디언트 크기가 특정 임계값보다 큰 픽셀을 선택할 경우, 에지 근방의 여러 픽셀이 한꺼번에 에지로 선택될 수 있다. **에지가 두껍게 표현되는 현상을 방지하기 위해** 캐니 에지 검출기에서는 비최대 억제(non-maximum suppression) 과정을 사용한다. 비최대 억제는 **그래디언트 크기가 국지적 최대(local maximum)인 픽셀만을 에지 픽셀로 설정하는 기법**이다. 상대적으로 국지적 최대가 아닌 픽셀은 에지 픽셀에서 제외하기 때문에 비최대 억제라는 용어를 사용한다.

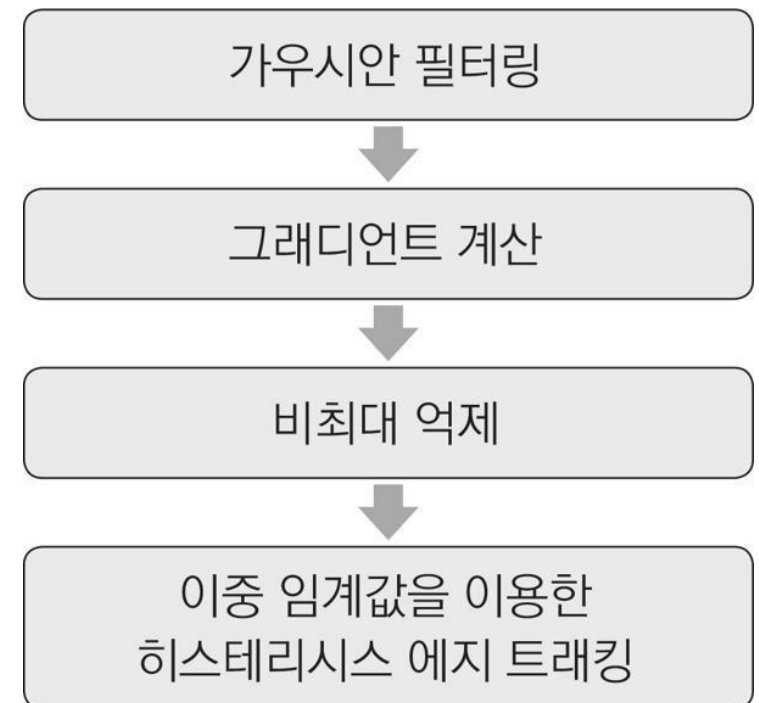
캐니 에지 검출기의 비최대 억제 과정에서는 **그래디언트 벡터의 방향과 같은 방향에 있는 인접 픽셀끼리만** 국지적 최대 검사를 수행한다. 결과적으로 비최대 억제를 수행함으로써 **가장 변화율이 큰 위치의 픽셀만 에지로 검색된다.**



캐니 에지 검출기 - 이중 임계값을 이용한 히스테리시스 에지 트래킹

그래디언트 크기가 특정 임계값보다 크면 에지 픽셀로, 작으면 에지가 아닌 픽셀로 판단하는 경우, 임계값을 조금만 조절해도 에지 픽셀 판단 결과가 크게 달라질 수 있다. 즉, 하나의 임계값을 사용할 경우 이분법으로 결과가 판단되기 때문에 환경 변화에 민감해질 수 있다. 이러한 문제를 보완하기 위해 **캐니 에지 검출기에서는 두 개의 임계값을 사용한다.**

캐니 에지 검출기에서 사용하는 두 개의 임계값 중에서 **높은 임계값을 T_{High} , 낮은 임계값을 T_{Low} 라고 표기한다.** 만약 **그래디언트 크기가 T_{High} 보다 크면 이 픽셀은 최종적으로 에지로 판단하고, 그래디언트 크기가 T_{High} 보다 작으면 에지 픽셀이 아니라고 판단한다.** 그래디언트 크기가 T_{Low} 와 T_{High} 사이인 픽셀은 에지일 수도 있고 에지가 아닐 수도 있다고 판단하며, 이런 픽셀에 대해서는 추가적인 검사를 수행한다.



캐니 에지 검출기 - 이중 임계값을 이용한 히스테리시스 에지 트래킹

OpenCV에서 캐니 에지 검출 알고리즘은 Canny()함수에 구현되어 있다.

```
void Canny(InputArray image, OutputArray edges,  
           double threshold1, double threshold2,  
           int apertureSize = 3, bool L2gradient = false);
```

```
void Canny(InputArray dx, InputArray dy, OutputArray edges,  
           double threshold1, double threshold2,  
           bool L2gradient = false);
```

Threshold1 : 낮은 임계값 지정

Threshold2 : 높은 임계값 지정

실습3

Canny()함수를 사용하여 영상에서 에지를 검출해본다.

`Canny(src, dst1, 50, 100);` → 낮은 임계값을 50, 높은 임계값을 100으로 설정하고 결과를 dst1에 저장

`Canny(src, dst2, 50, 150);` → 낮은 임계값 ≥ 50 , 높은 임계값을 150으로 설정하고 결과를 dst2에 저장



임계값을 낮출 수록 에지로 판별되는 픽셀이 더 많아지므로, dst1영상에서 더 많은 에지 픽셀이 검출된 것을 확인할 수 있다.