영상 공간 필터링 – 2

- 미분필터, 기타 필터

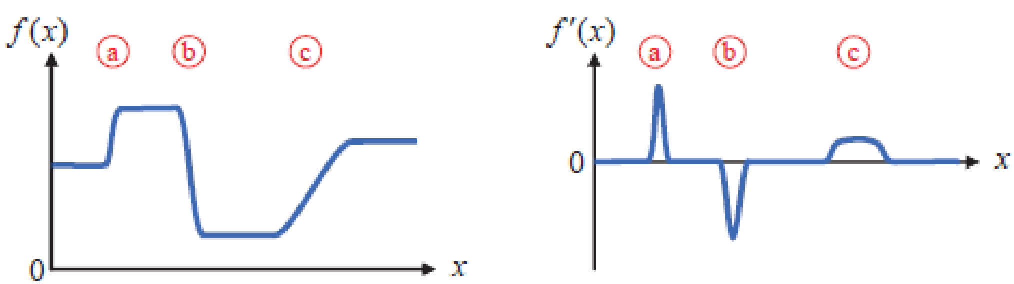
미디어기술콘텐츠학과
강호철

미분 필터링

- 미분과 경사도
  - 함수 또는 데이터의 변화율
  - 함수의 순간 변화율

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

- 앞 수식에서 Δx는 x의 변화량을 의미함
- x 의 변화량이 무한히 0에 가까워질 때의 함수 값 변화량을 미분이라고 함
- 함수 값이 증가하는 위치에서는 함수의 미분 값이 0보다 큰 양수로 나타남
- 함수 값이 감소하는 위치에서는 함수의 미분 값이 0보다 작은 음수를 갖게 됨
- 함수 값이 일정한 구간에서는 함수의 미분이 0에 가까운 값을 가짐



(a) (b)

미분 필터링

- 미분과 경사도
  - 1차 미분과 차분

$$f'(x) = \frac{\partial f(x)}{\partial x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$= f(x+1) - f(x) \text{ , if } h = 1$$

  - 2차 미분과 차분

$$f''(x) = \frac{\partial^2 f(x)}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

미분 필터링

- 영상에서의 1차 미분
  - 차분의 종류
    - 전진 차분(forward difference): $\frac{dI}{dx} \cong \frac{I(x+h)-I(x)}{h}$
    - 후진 차분(backward difference): $\frac{dI}{dx} \cong \frac{I(x)-I(x-h)}{h}$
    - 중앙 차분(centered difference): $\frac{dI}{dx} \cong \frac{I(x+h)-I(x-h)}{2h}$

미분 필터링

- 영상에서의 경사도
  - 2차원 함수의 편미분

$$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

Sobel

$$g_x = \frac{\partial f(x,y)}{\partial x} = [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)]$$

$$g_y = \frac{\partial f(x,y)}{\partial y} = [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)]$$

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

(a)

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

(b)

미분 필터링

- 영상에서의 경사도
  - 크기와 방향

$$mag(\nabla f(x,y)) = \sqrt{(g_x^2 + g_y^2)} = \sqrt{(\frac{\partial f(x,y)}{\partial x})^2 + (\frac{\partial f(x,y)}{\partial y})^2}$$

$$\approx |\frac{\partial f(x,y)}{\partial x}| + |\frac{\partial f(x,y)}{\partial y}|$$

$$\theta = angle(\nabla f(x,y)) = \mathrm{atan}(\frac{g_y}{g_x})$$

미분 필터링

- 영상에서의 2차 미분
  - 라플라시안 (Laplacian)

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y}$$

$$= [f(x+1,y) + f(x-1,y) - 2f(x,y)] + [f(x,y+1) + f(x,y-1) - 2f(x,y)]$$

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

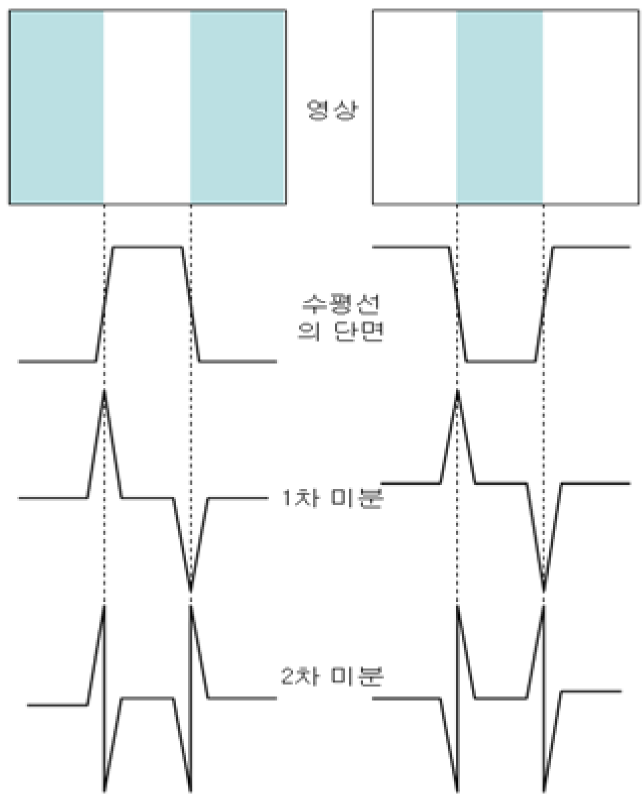
$$\frac{\partial^2 f(x,y)}{\partial^2 x} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f(x,y)}{\partial^2 y} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

미분 필터링

- 영상에서의 2차 미분
  - 라플라시안 (Laplacian)
    - 1차 미분은 엣지 존재 여부 파악
    - 2차 미분은 엣지 밝기 변화 파악 가능
    - 에지 위치에 zero-crossing
    - 밝기 값이 점차적으로 변화되는 영역에 대해서는 반응을 보이지 않음
    - 잡음에 민감



미분 필터링

- 실습

Sobel Filter

```
cv2.Sobel(src, ddepth, dx, dy[, dst[, ksize[, scale[, delta[,borderType]]]]])
```

: 지정한 축방향으로의 디지털 형태의 미분을 구현. 1차미분을 통하여 특정한 Edge를 검출하는 Filter이다.

parameter

- src: input
- ddept: output image depth
- dx: x에 대한 미분 차수
- dy: y에 대한 미분 차수
- ksize: Sobel kernel 크기 (1,3,5,7만 가능)

example

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

수직 엣지 검출

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

수평 엣지 검출

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

대각 엣지 검출1

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

대각 엣지 검출2

출처: https://wjddyd66.github.io/opencv/OpenCV(5)/#sobel-filter1

미분 필터링

- 실습

Sobel Filter2(gradient orientation)

`cv2.cartToPolar(x, y[, magnitude[, angle[, angleInDegrees]]])` 는
2D vector의 magnitude계산및 각도도 Return하는 함수이다.
결과 angleM에서 색깔은 각각의 의미와 같다.

- 빨강: magnitude의 값 중 angle = 0 인곳
- 초록: magnitude의 값 중 angle = 90 인곳
- 파랑: magnitude의 값 중 angle = 180 인곳
- 노랑: magnitude의 값 중 angle = 270 인곳

출처: https://wjddyd66.github.io/opencv/OpenCV(5)/#sobel-filter1

미분 필터링

- 실습

Laplacian Filter

`cv2.Laplacian(src, ddepth[, dst[, ksize[, scale[, delta[, borderType]]]]])` : 1차 미분을 사용하는 Sobel Filter에 비하여 영상 내에 blob이나, 섬세한 부분을 더 잘 검출하는 경향을 보인다. 이러한 특징 때문에 **2차 미분은 주로 영상개선에 사용하며 1차 미분은 특징검출**에 사용된다.

example(3 x 3, ksize=1)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

출처: https://wjddyd66.github.io/opencv/OpenCV(5)/#sobel-filter1

미분 필터링

- 실습

`cv2.magnitude(x, y[, magnitude])` : 2D vector의 magnitude계산

parameter

- x: floating point1
- y: floating point2
- manitude: $dst = \sqrt{x^2 + y^2}$

`cv2.normalize(src[, dst[, alpha[, beta[, norm_type[, dtype[, mask]]]]]])`

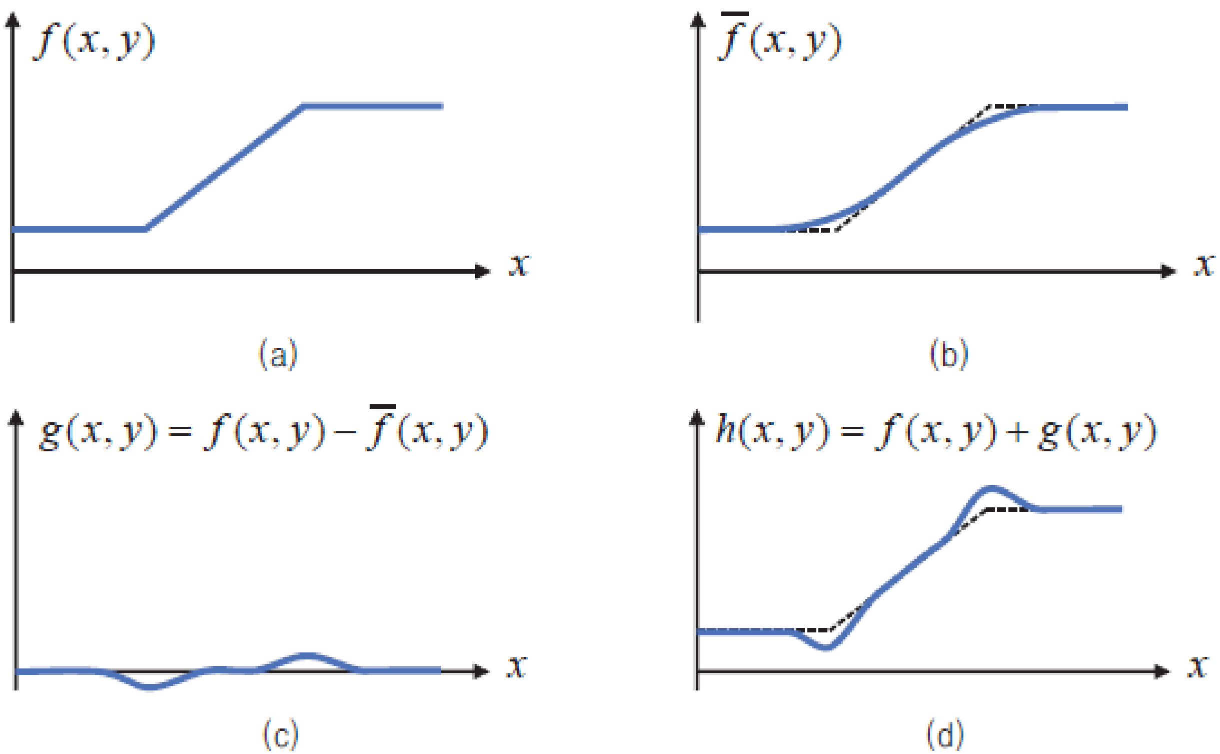
parameter

- src: input
- alpha: norm value to normalize to or the lower range boundary
- beta: upper range boundary
- normType: normalization type

출처: https://wjddyd66.github.io/opencv/OpenCV(5)/#sobel-filter1

샤프닝 필터링

- 언샤프 마스크 필터
  - 블러링이 적용된 날카롭지 않은 영상을 이용하여 날카로운 영상으로 만드는 필터링



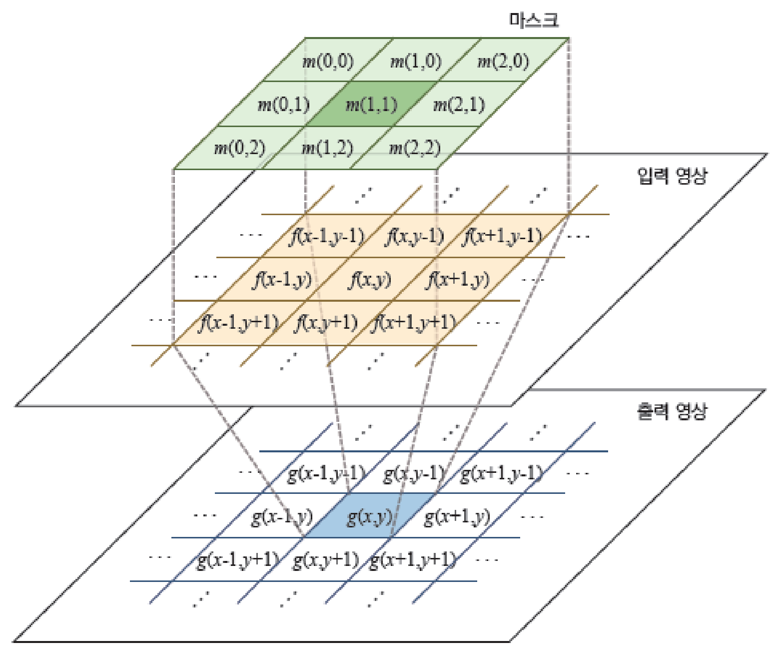
샤프닝 필터링

- 언샤프 마스크 필터
  - 실습

일반적인 필터링

- 컨볼루션 연산
  - 영상과 마스크의 합성곱

$$dst(x,y) = \sum_{s=0}^{\ker \neq l.cols-1} \sum_{t=0}^{\ker \neq l.rows-1} kernel(s,t)\ src(x+s-anchor.x,\ y+t-anchor.y)$$



일반적인 필터링

- 영상과 마스크의 합성곱
  - OpenCV 함수

`cv2.filter2D(src, ddepth, kernel[, dst[, anchor[, delta[, borderType]]]])` : Kernel을 통하여 Image를 Filtering하는 방법

`cv2.sepFilter2D(src, ddepth, kernelX, kernelY[, dst[, anchor[, delta[, borderType]]]])` : image의 x, y를 각각의 kernel을 통하여 Filtering하는 방법

parameter

- kenelX: image x-축에 적용할 kernel
- kenelY: image y-축에 적용할 kernel

출처: https://wjddyd66.github.io/

일반적인 필터링

- LoG 필터링
  - Laplacian of Gaussian

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1-\frac{x^2+y^2}{2\sigma^2}]\exp(-\frac{x^2+y^2}{2\sigma^2})$$

  - 왜?

일반적인 필터링

- 실습

화이트 보드

영상처리 프로그래밍 기초

- Python으로 배우는 OpenCV 프로그래밍
  - 김동근 지음
  - 가메출판사, 2018
- OpenCV4 로 배우는 컴퓨터 비전과 머신러닝
  - 황선규 지음
  - 길벗, 2019