

# Open CV를 이용한 Feature Detector 구현

---

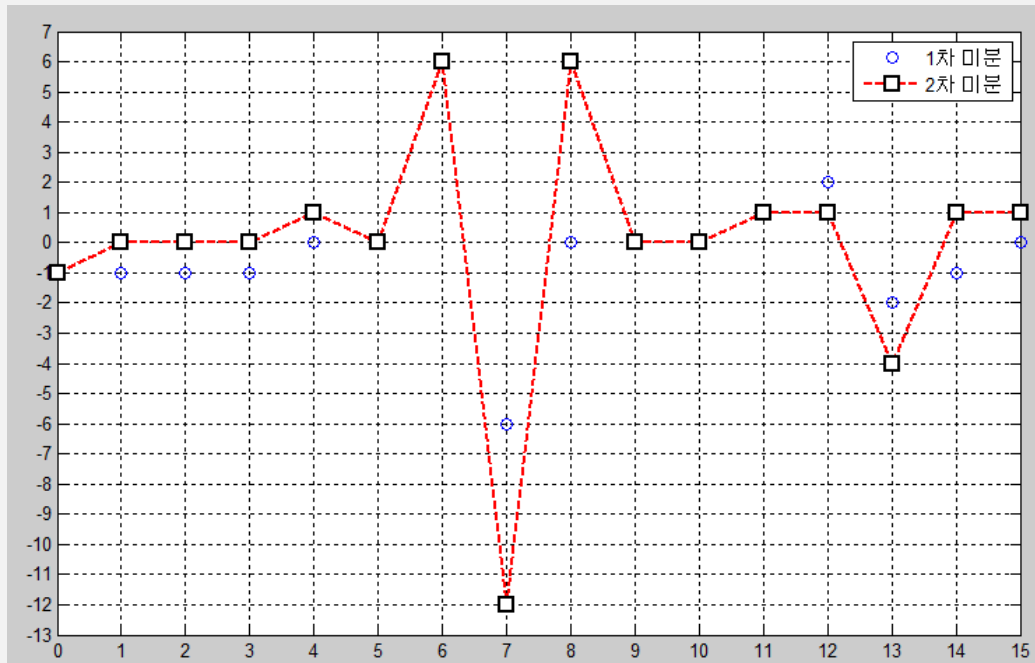
# ▶ Index

- Laplacian
- Gaussian
- Canny Edge

# Laplacian

## 라플라시안 연산을 이용한 필터링

영상띠	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	...
1차미분	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0			
2차미분	-1	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1			



# Laplacian

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

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

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

$$\therefore f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

# Laplacian

- Open CV 함수

입력 영상

출력 영상

The destination  
image depth

```
cv::Laplacian(InputArray src, OutputArray dst, int ddepth,  
              int ksize=1, double scale=1, double delta=0,  
              int borderType=BORDER_DEFAULT )
```

커널 크기

# Laplacian



<원본 이미지>



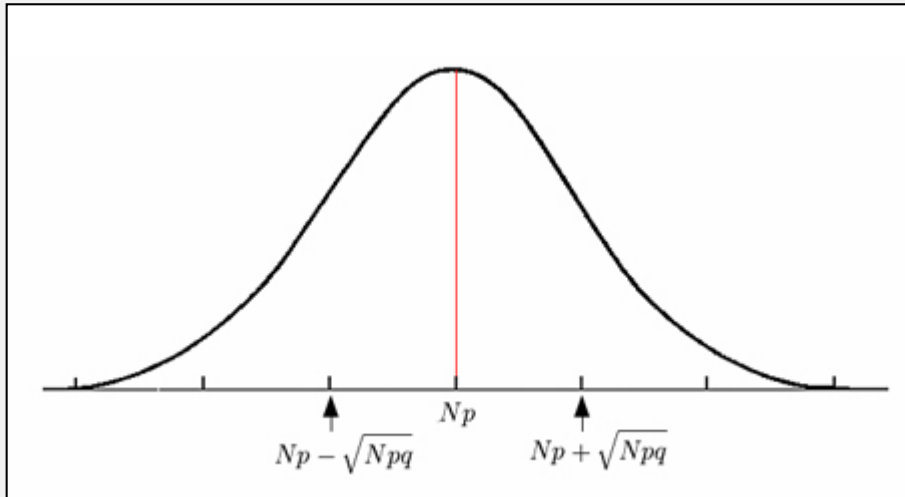
<Laplacian 이미지>

# Gaussian

- 가우시안을 이용한 블러효과

<Gaussian function>

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\mu = \text{평균}, \sigma = \text{표준편차})$$



<Ideal LPF>

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

$D_0 = \text{Cutoff freq.}$

<Gaussian LPF>

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$

# Gaussian

- LoG(Laplacian of Gaussian) – Marr & Hildreth (1980)

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

<Gaussian function>

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\mu = \text{평균}, \sigma = \text{표준편차})$$

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

$$= \frac{\partial}{\partial x} \left[ -\frac{x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right] + \frac{\partial}{\partial y} \left[ -\frac{y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right]$$

$$= \left[ \frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[ \frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\therefore \nabla^2 G(x, y) = \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



# Gaussian

## ● Open CV 함수

입력 영상

출력 영상

커널 사이즈

```
cv::GaussianBlur(InputArray src, OutputArray dst, Size ksize,  
double sigmaX, double sigmaY=0,  
int borderType=BORDER_DEFAULT )
```

시그마

# Gaussian



<원본 이미지>



<LoG 이미지>

# Canny Edge

- Canny Edge 알고리즘 - John F. Canny (1986)
  1. Gauss 필터를 이용한 스무딩
  2. 기울기 크기 영상과 각도 영상을 계산
  3. 기울기 크기 영상에 비 최대 억압을 적용
  4. 에지를 검출하고 연결하기 위해 이력 문턱치 처리

# Canny Edge

- 이력 문턱치 처리(Hysteresis thresholding)



$$g_{NH}(x, y) = g_N(x, y) \geq T_H$$

$$g_{NL}(x, y) = g_N(x, y) \geq T_L$$

\*초기에  $g_{NL}(x, y), g_{NH}(x, y)$  0으로 설정

$$g_{NL}(x, y) = g_N(x, y) - g_{NH}(x, y)$$

↖	↑	↗
←	$p$	→
↙	↓	↘

- (1)  $g_{NH}(x, y)$ 에서 방문되지 않은 그 다음 에지 화소  $p$ 를 찾아낸다
- (2) 8연결성으로  $p$ 에 연결된  $g_{NL}(x, y)$ 의 모든 약한 화소들을 유효에지 화소로 표시한다.
- (3)  $g_{NH}(x, y)$ 에 있는 0이 아닌 모든 화소들이 방문되었다면 단계 (4)로 간다. 아니면 단계 (1)로 간다.
- (3) 유효 에지 화소로 표시되지 않은  $g_{NL}(x, y)$ 에 있는 모든 화소를 0으로 만든다.

# Canny Edge

## ● Open CV 함수

입력 영상

출력 영상

`cv::Canny (InputArray src, OutputArray edges,  
double threshold1, double threshold2,  
int apertureSize=3, bool L2gradient=false)`

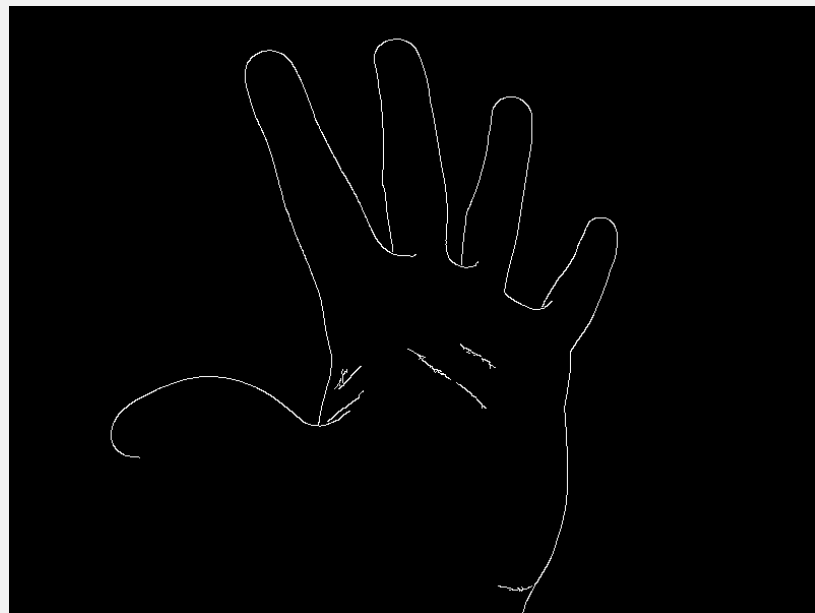
낮은 문턱치

높은 문턱치

# Canny Edge



<원본 이미지>



<Canny Edge 이미지>

Q & A