



# Recuperación de Información Multimedia

## Detección de Bordos

**CC5213 – Recuperación de Información Multimedia**

Departamento de Ciencias de la Computación

Universidad de Chile

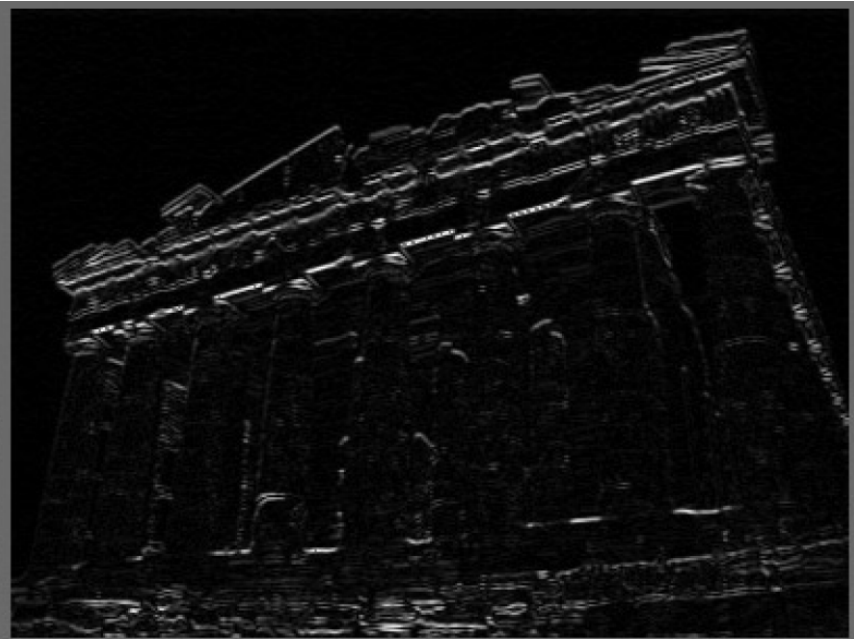
Juan Manuel Barrios – <https://juan.cl/mir/> – 2019

# Gradiente (eje x)



-1	0	1
-1	0	1
-1	0	1

# Gradiente (eje y)



-1	-1	-1
0	0	0
1	1	1



# Gradiente

$$\vec{\nabla} I = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix} = \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

Magnitud del Gradiente:

$$\begin{aligned} ||\vec{\nabla} I(i, j)|| &= \sqrt{I_x(i, j)^2 + I_y(i, j)^2} \\ &\approx |I_x(i, j)| + |I_y(i, j)| \end{aligned}$$

Orientación del Gradiente:

$$\theta(i, j) = \arctan \left( \frac{I_y(i, j)}{I_x(i, j)} \right)$$



# Gradiente

## ■ Prewitt

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

-1	-1	0
-1	0	1
0	1	1

0	1	1
-1	0	1
-1	-1	0



# Gradiente

- Sobel:

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



# Gradiente

- Sobel (descomposición):

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

1
2
1

-1	0	1
----	---	---

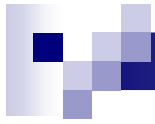


# Gradiente

## ■ Scharr

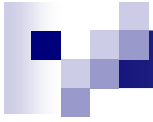
-3	0	3
-10	0	10
-3	0	3





# Ejemplo





# Ejemplo



$I_x$

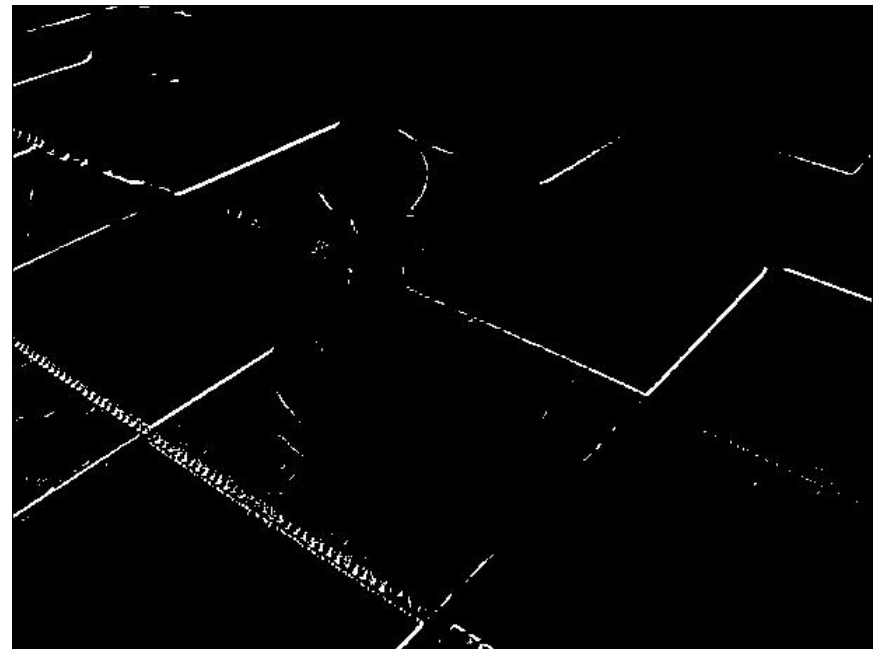


$I_y$

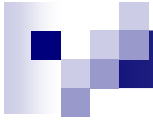
# Ejemplo



Magnitud del gradiente



Aplicando un umbral



# Kernel

0	1	0
1	-4	1
0	1	0



# Segunda Derivada Discreta

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

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



# Laplaciano

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

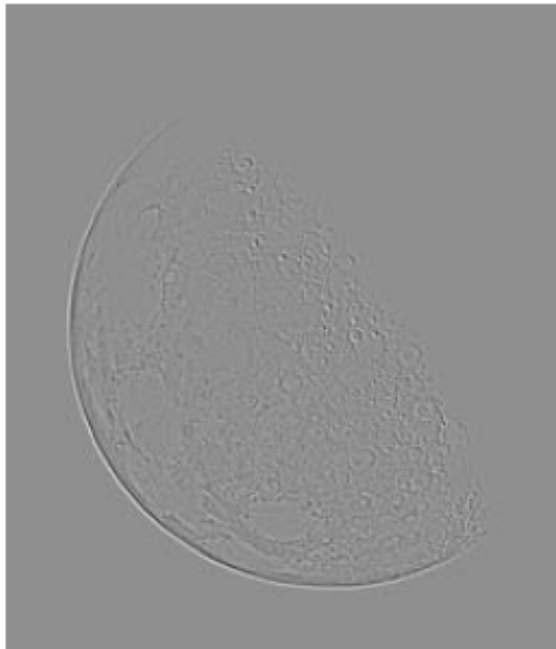
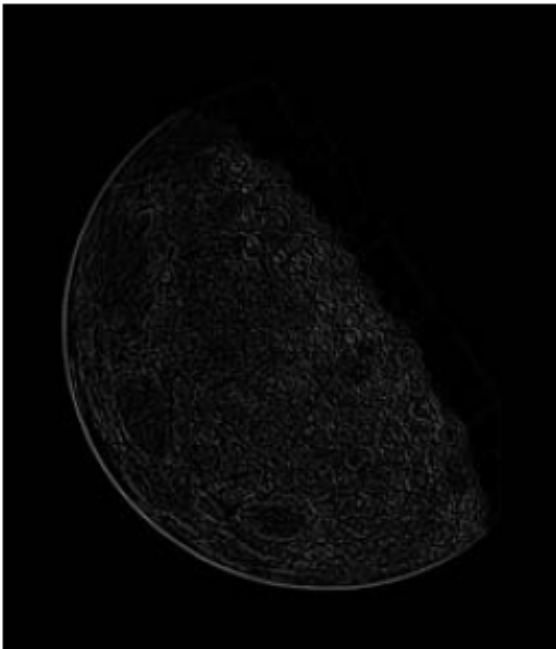
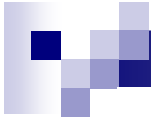
$$\begin{aligned}\nabla^2 f = & [f(x + 1, y) + f(x - 1, y) \\ & + f(x, y + 1) + f(x, y - 1)] \\ & - 4f(x, y)\end{aligned}$$



# Kernels basados en Laplaciano

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1







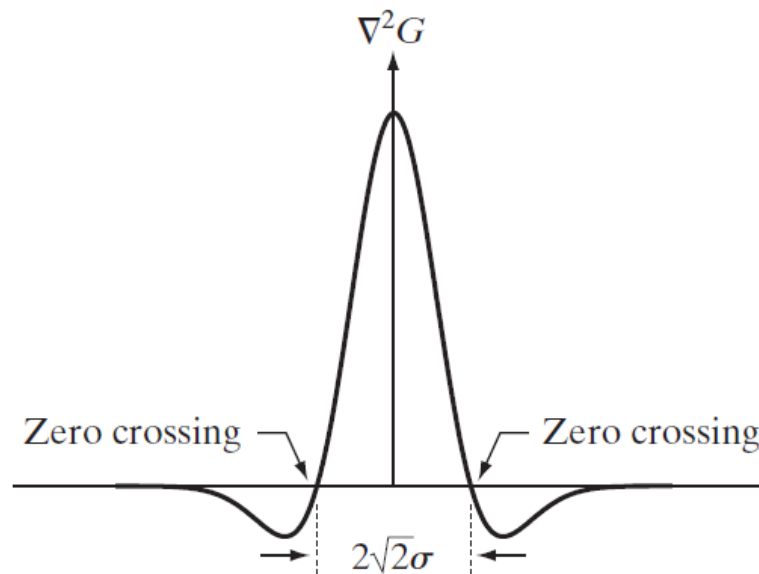
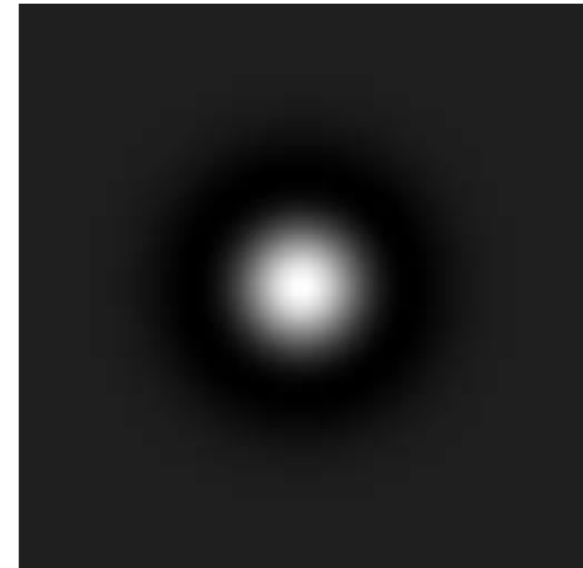
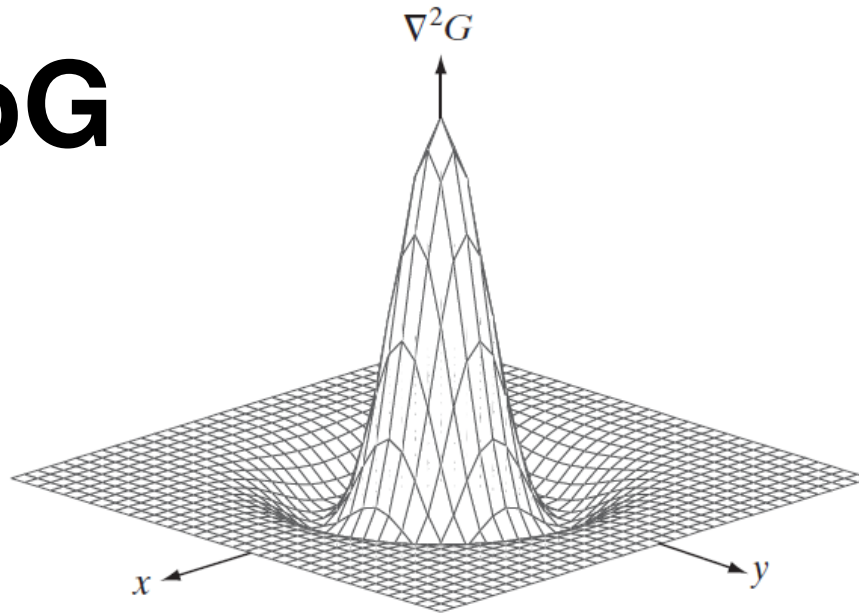
# Laplaciano

- El Laplaciano es muy susceptible a ruido
- Es usual primero aplicar un suavizado gaussiano y luego el Laplaciano
- Ambos procesos unidos se conocen como Laplaciano de Gaussiana (LoG)

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \quad r^2 = x^2 + y^2$$

$$\nabla^2 h(r) = - \left[ \frac{r^2 - \sigma^4}{\sigma^4} \right] e^{-\frac{r^2}{2\sigma^2}}$$

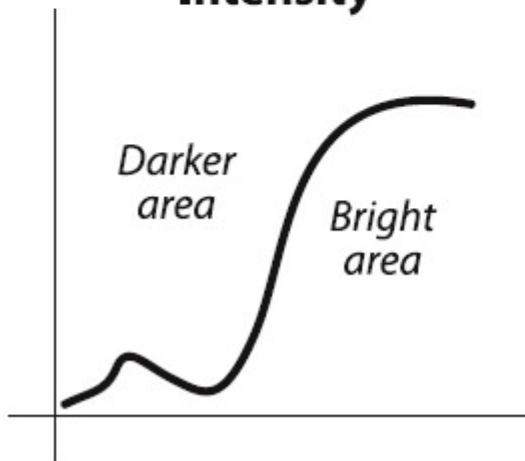
# LoG



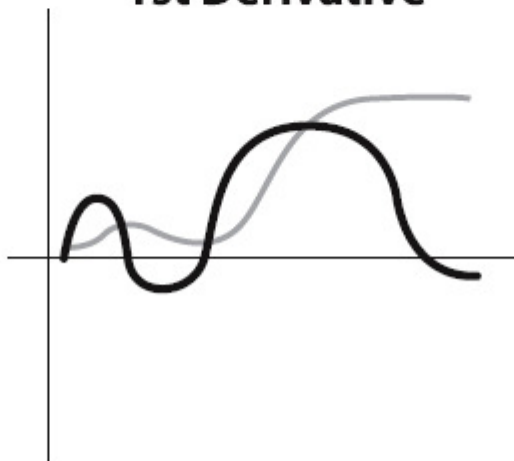
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



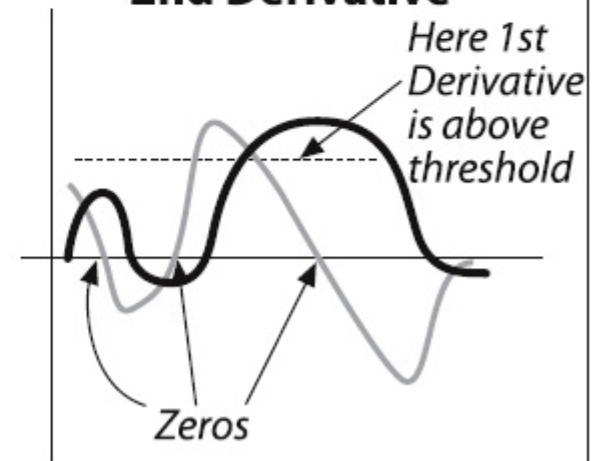
**Intensity**



**1st Derivative**



**2nd Derivative**





# Canny

- Se basa en el criterio de primera y segunda derivada para detectar bordes delgados
- Seleccionar pixeles de borde en forma incremental siguiendo la dirección perpendicular al gradiente
- Aplicar diferentes filtros gaussianos para obtener bordes a distintas escalas y unirlos
- Descartar pixeles cuya magnitud del gradiente no sea un máximo local en una vecindad 3x3



# Canny

- Usar dos umbrales de selección:  $T_{sup}$  y  $T_{inf}$ , el mayor 2 a 3 veces más grande que el menor
  - Pixeles con gradiente  $\geq T_{sup}$  es seleccionado, se recorren vecinos perpendicular al gradiente
  - Pixeles con gradiente  $< T_{inf}$  rechazado
  - Pixeles con gradiente  $\geq T_{inf}$  y con al menos un vecino ya seleccionado también es seleccionado

# Detección según Canny



$T_{sup}=500$   $T_{inf}=10$



$T_{sup}=500$   $T_{inf}=100$



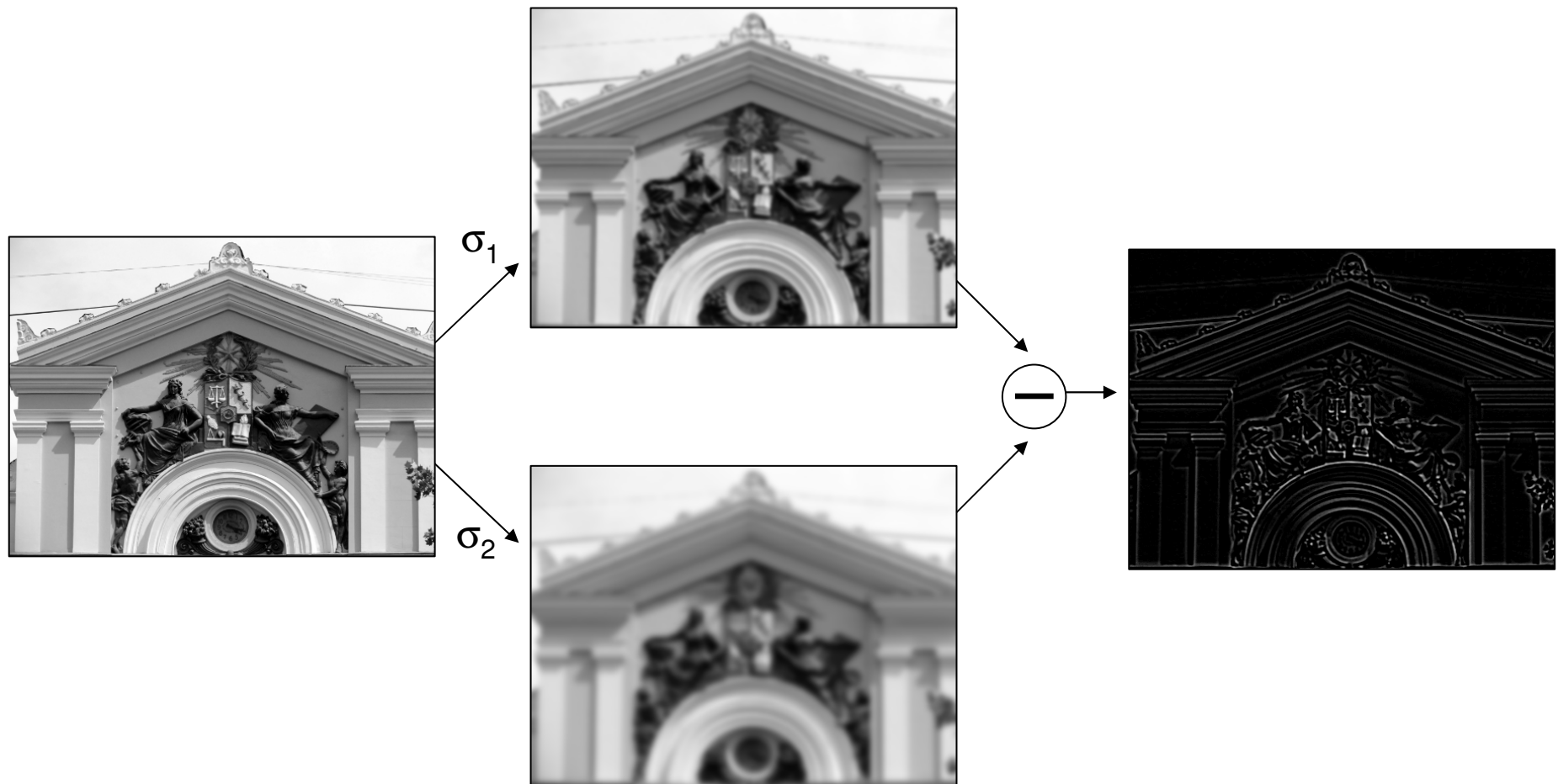
$T_{sup}=500$   $T_{inf}=400$



# Diferencia de Gaussianas (DoG)

- El blur afecta una imagen mayormente en zonas con gran variación en la intensidad
- Al comparar la imagen original con la imagen borrosa se pueden localizar las zonas más afectadas
- Filtro gaussiano se usa para eliminar ruido
- Usar dos filtros gaussianos: uno pequeño para eliminar ruido y uno más grande para restar

# Ejemplo



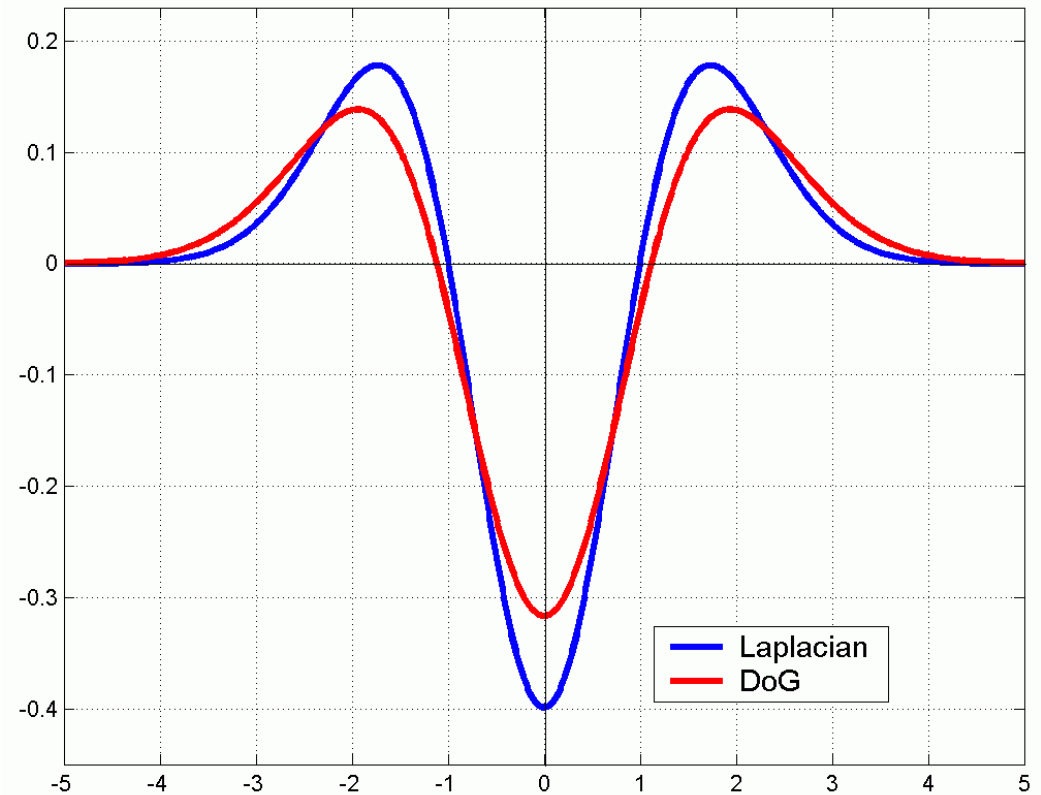


# Comparación DoG vs LoG

$$L = \sigma^2 \left( G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



# Bibliografía

- **Digital Image Processing.** González et al. 2008.
  - Cap. 10
- **The Essential Guide To Image Processing.** Bovik. 2009.
  - Cap. 19
- **Learning OpenCV.** Bradski et al. 2008.
  - Cap. 6
- **Multimedia Retrieval.** Blanken et al. 2007.
  - Cap. 5 (canny)

