

# Introducción

Procesamiento Avanzado de Imágenes  
Javier Ruiz del Solar  
2022

## Computer Vision

Image Processing

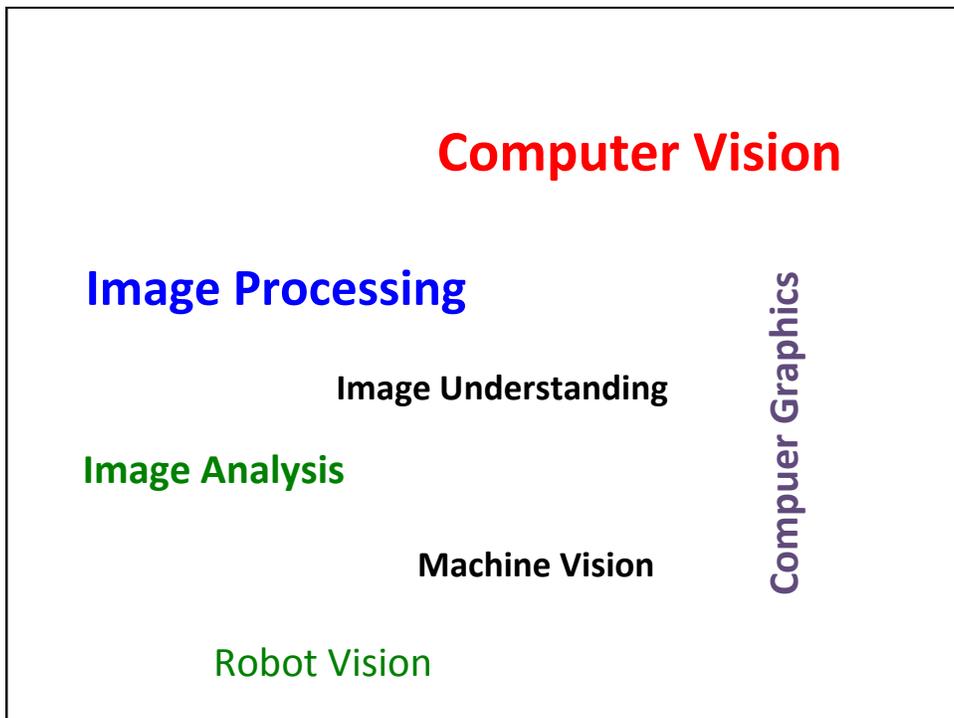
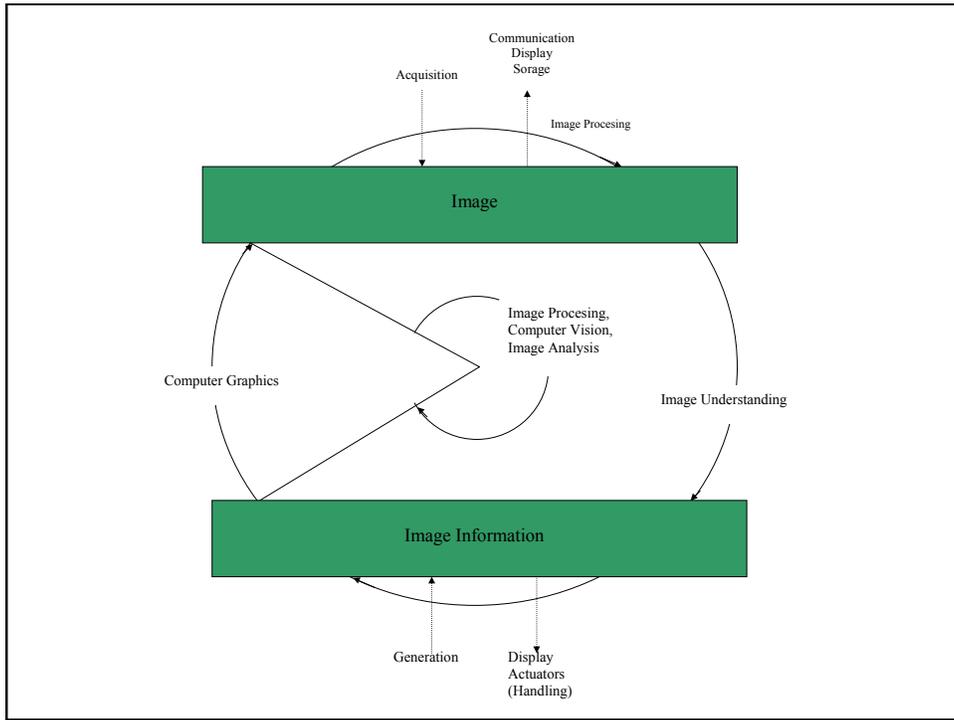
Image Understanding

Image Analysis

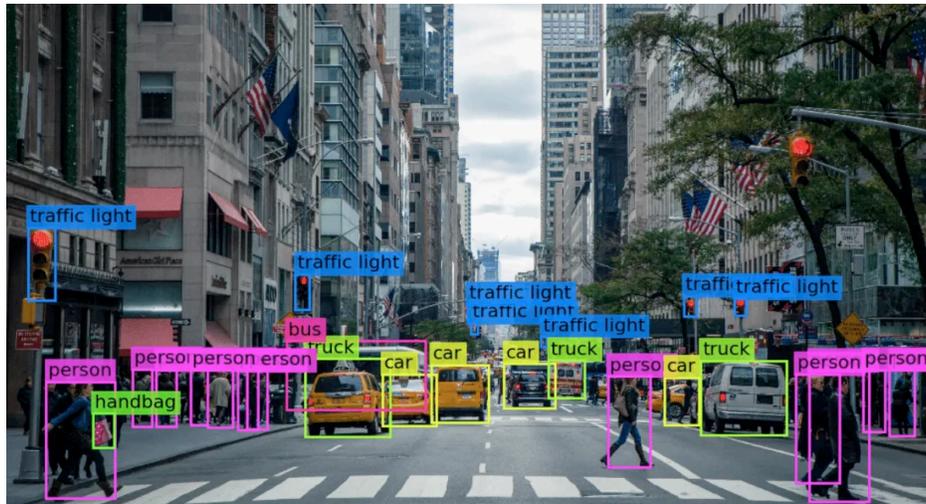
Machine Vision

Robot Vision

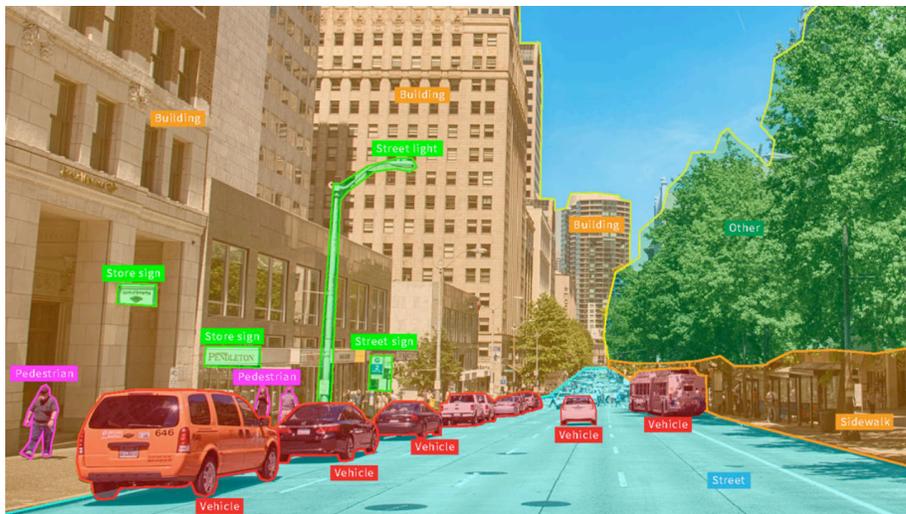
Computer Graphics



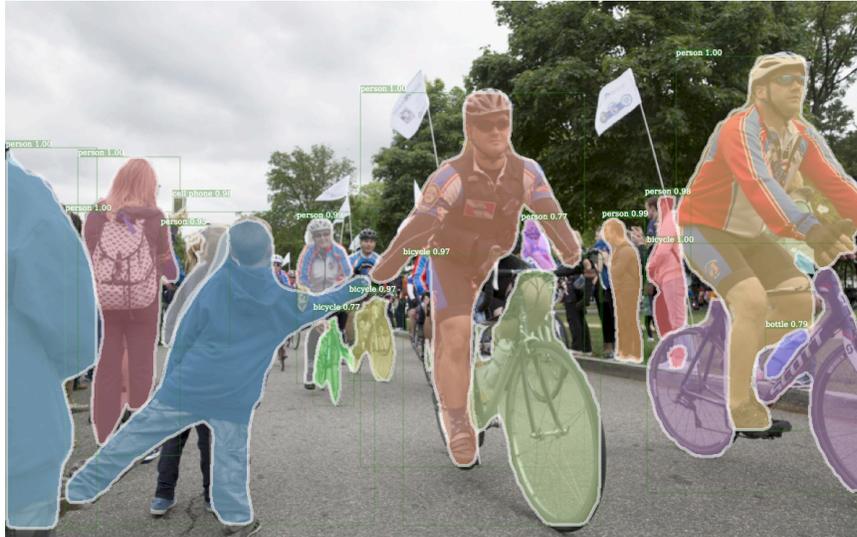
## Estado del Arte



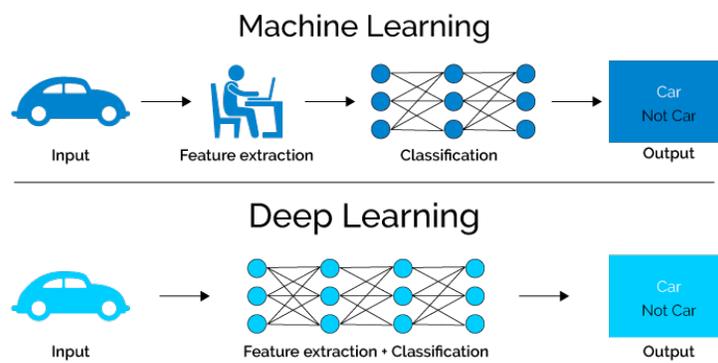
## Estado del Arte



## Estado del Arte

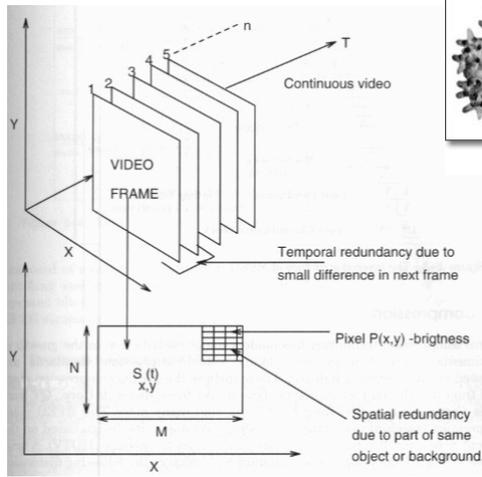


## Paradigmas



# Imágenes Digitales

pixel: picture element



Source: <http://people.cs.pitt.edu/~chang/265/CS/v2.htm>

Una imagen es una señal 2D discreta, pero el mundo es 3D y continuo

Creation of a Digital Image

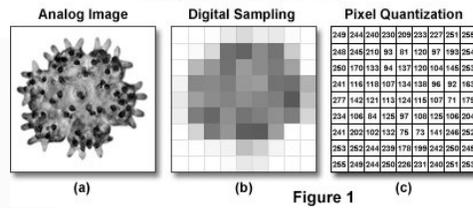
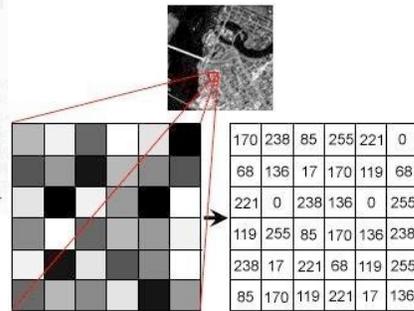


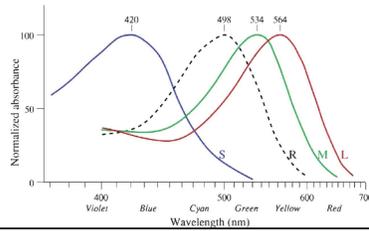
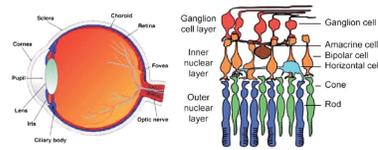
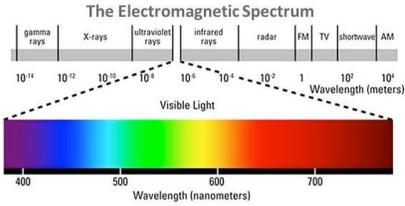
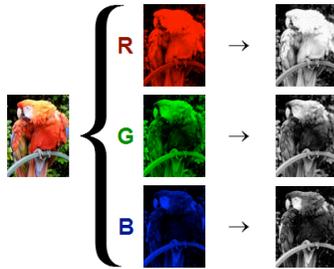
Figure 1

Source: <http://hamamatsu.magnet.fsu.edu/articles/digitalimagebasics.html>



Source: [https://www.researchgate.net/publication/311806469\\_Design\\_and\\_Fabrication\\_of\\_an\\_Image\\_Processing\\_Based\\_Autonomous\\_Weapon/figures?](https://www.researchgate.net/publication/311806469_Design_and_Fabrication_of_an_Image_Processing_Based_Autonomous_Weapon/figures?)

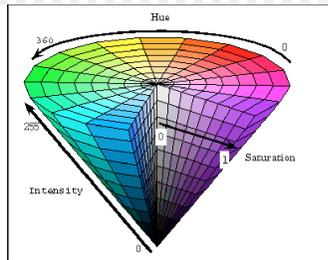
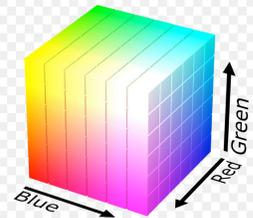
Color -> percibido por humanos como espacio tri-cromático  
 → cada pixel requiere 3 componentes



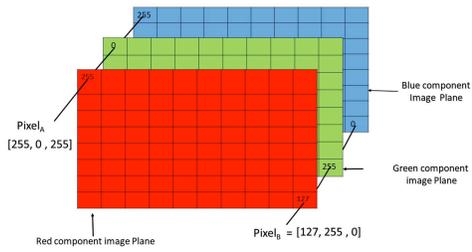
Source: [reddit.com/r/space/comments/5aw6nh/probably the prettiest picture of saturn ever/](https://www.reddit.com/r/space/comments/5aw6nh/probably_the_prettiest_picture_of_saturn_ever/)

Source: [https://www.researchgate.net/publication/255774371\\_Organic\\_semiconductors\\_for\\_artificial\\_vision/figures?o=1](https://www.researchgate.net/publication/255774371_Organic_semiconductors_for_artificial_vision/figures?o=1)

Cada pixel requiere 3 componentes  
 Muchos espacios de color: RGB, HSI, etc.



Representación computacional, cada pixel tiene 3 componentes

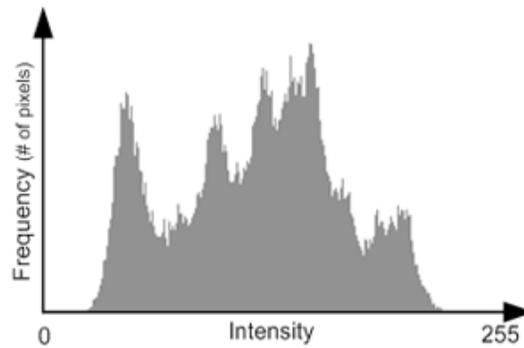


Source: <https://www.geeksforgeeks.org/matlab-rgb-image-representation/>

Source: [https://www.researchgate.net/publication/335504961\\_Construct\\_a\\_Strong\\_and\\_High\\_Performance\\_Algorithm\\_to\\_Generate\\_Pseudorandom\\_Number\\_Generator\\_PRNG\\_for\\_Stream\\_Cipher/figures?o=1](https://www.researchgate.net/publication/335504961_Construct_a_Strong_and_High_Performance_Algorithm_to_Generate_Pseudorandom_Number_Generator_PRNG_for_Stream_Cipher/figures?o=1)

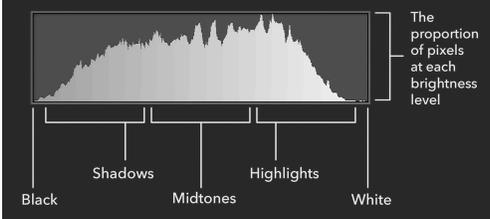
# Histogramas

Histograma: estadística de primer orden de los píxeles de una imagen

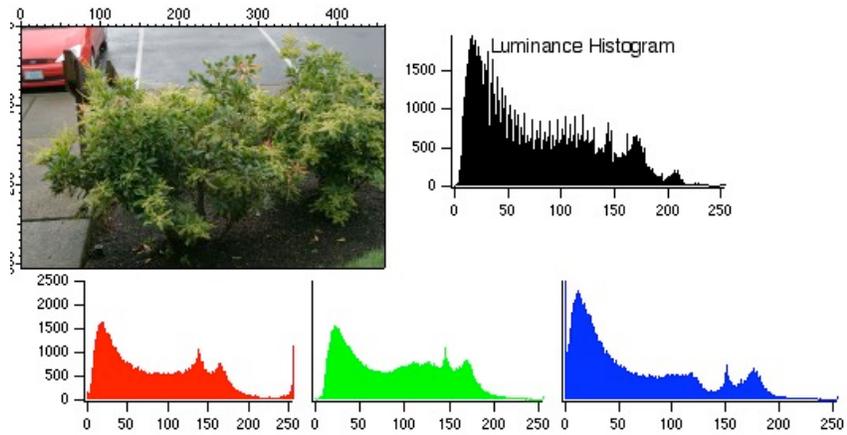


Source: <http://www.songho.ca/dsp/histogram/histogram.html>

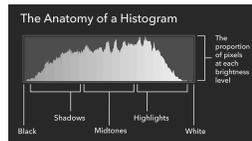
## The Anatomy of a Histogram



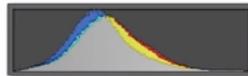
En las imágenes de color se calcula un histograma por cada canal



El histograma es una herramienta que permite mejorar el contenido de una imagen, al visualizar la distribución de sus píxeles



Correct Exposure



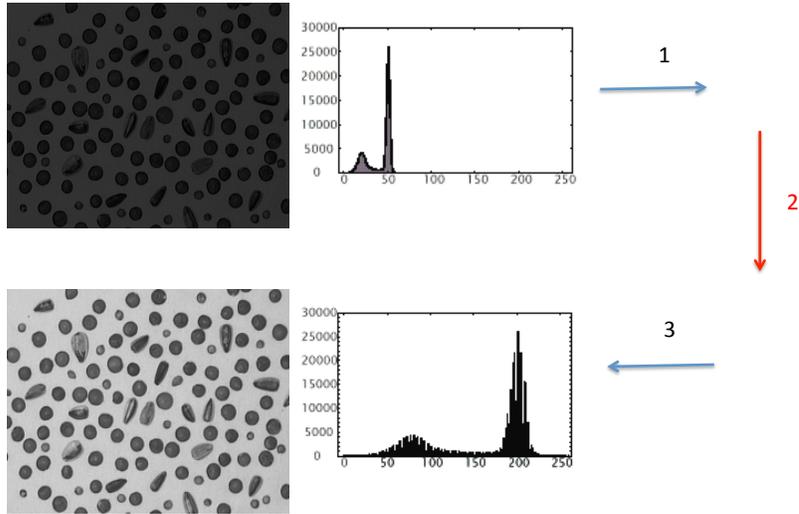
Over Exposure



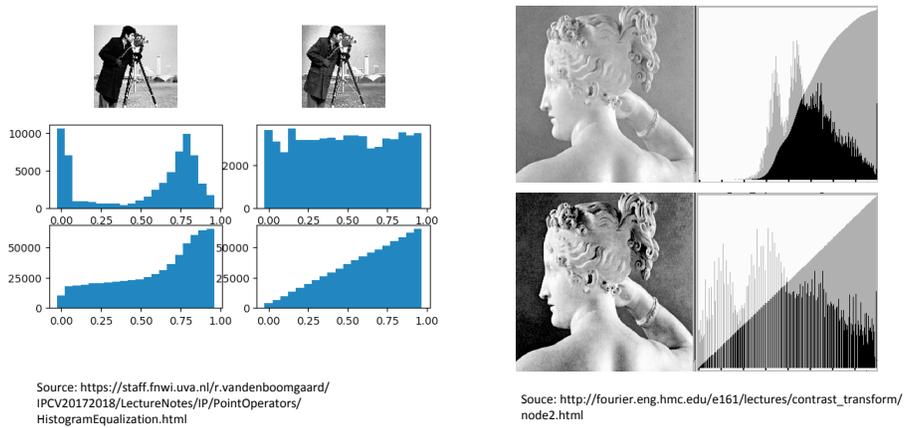
Under Exposure



### Manipulación de histograma



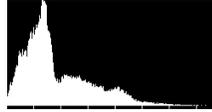
### Ecuación de histograma



### Ecuación de histograma

#### Cálculo de histograma

```
d=1.0/M/N;
for (i=0; i < 256; i++) hist[i] = 0;
for (i=0; i < M; i++)
for (j=0; j < N; j++)
  hist[x[i][j]] += d;
```



$x(i)$



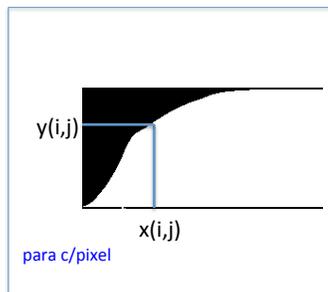
#### Cálculo de histograma acumulado

```
sum=0.0;
for (i=0; i < 256; i++)
  sum += hist[i];
lookup[i] = sum * 255 + 0.5;
```



#### Equalización

```
for (i=0; i < M; i++)
for (j=0; j < N; j++)
  y[i][j] = lookup[x[i][j]];
```

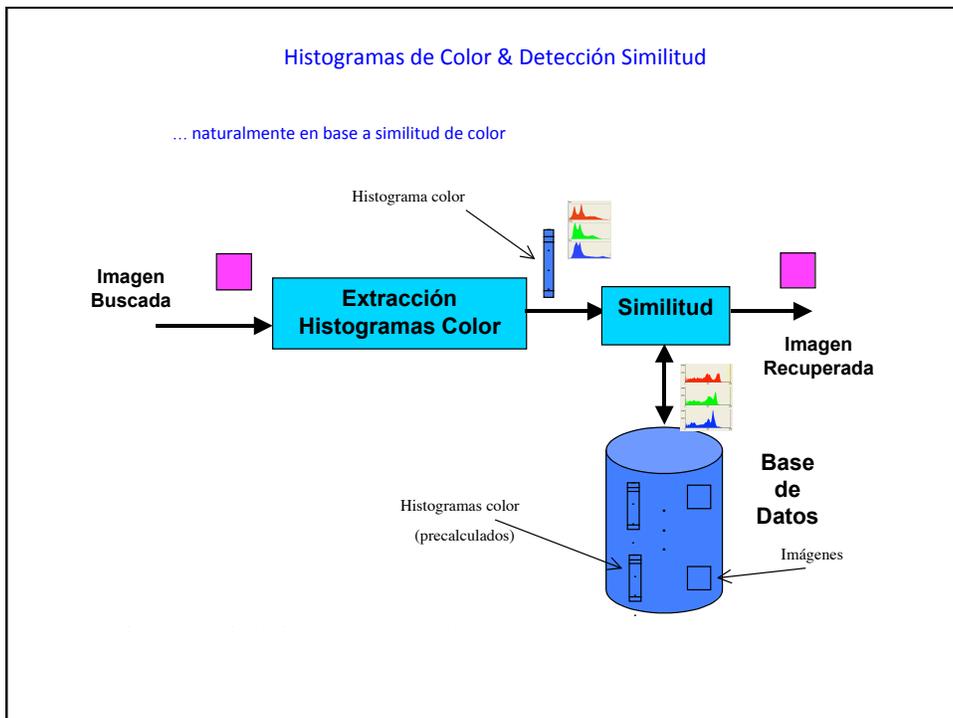
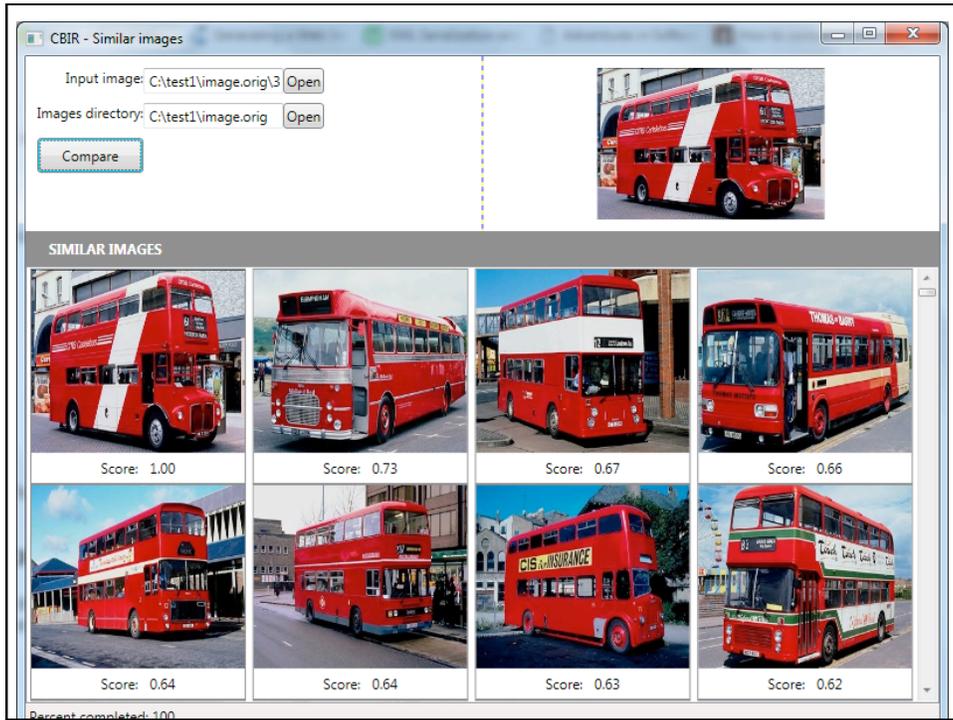


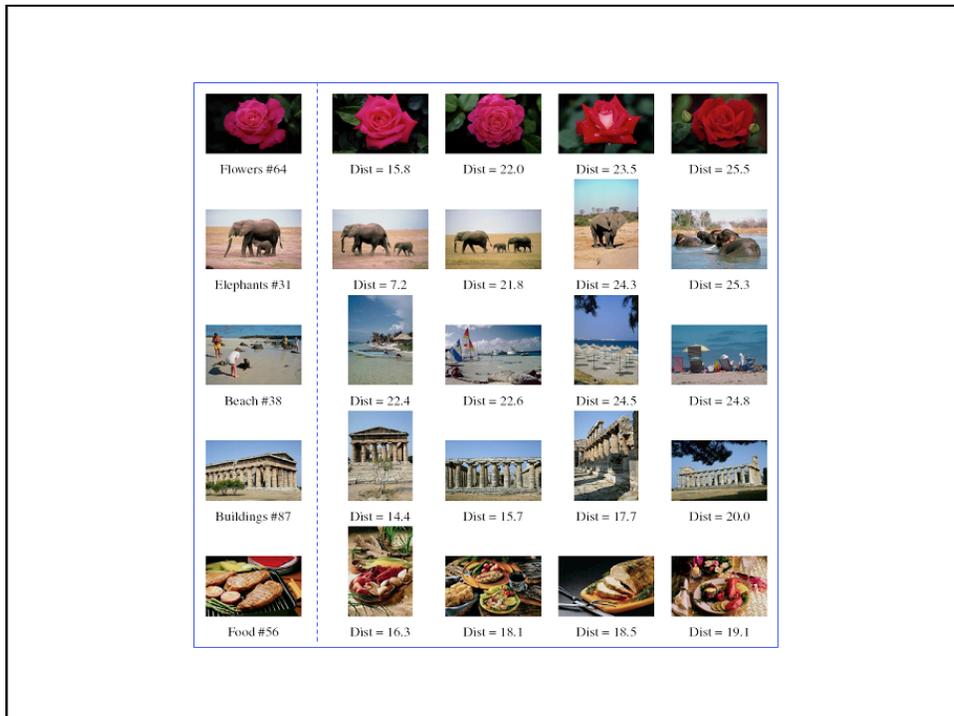
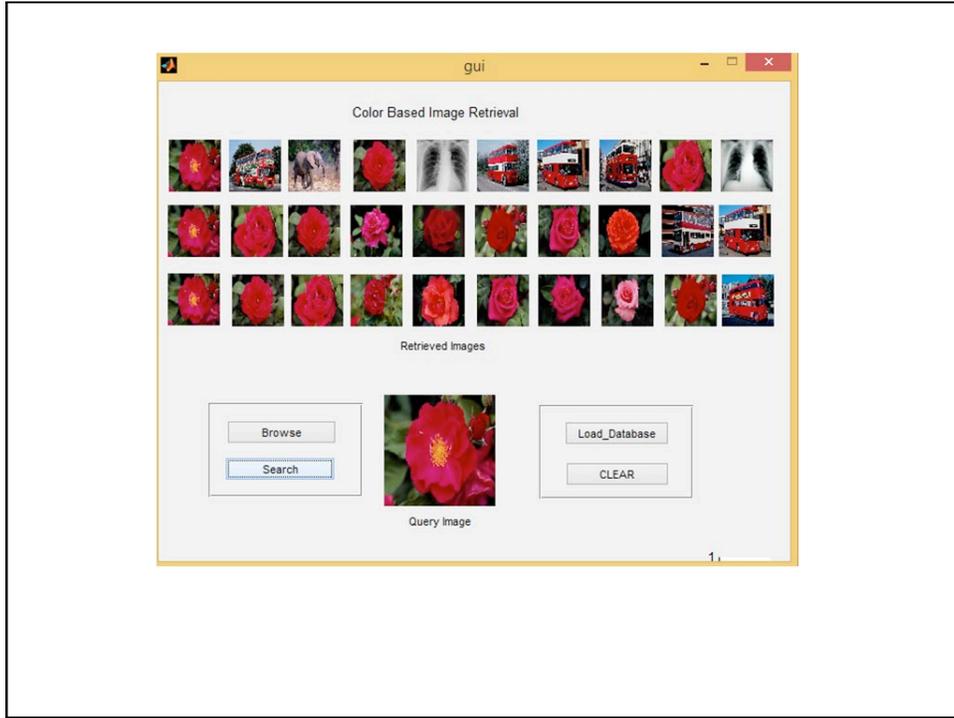
$y(i)$



(ver documentos subidos a U-Cursos explicando algoritmo)

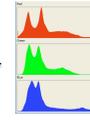
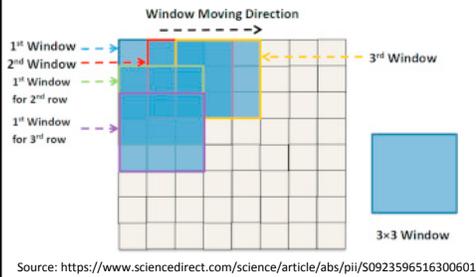
## Histogramas de Color & Detección de Similitud





### Histogramas de Color & Detección de Objetos

- Naturalmente si los objetos son caracterizables en base a su color
- Si este fuera caso y si se usa paradigma de ventana deslizante, la detección es invariante a traslaciones, rotaciones en el plano, cambios de escala (si se normaliza histograma) y parcialmente invariante a rotaciones fuera del plano.
- Parcialmente invariante a la iluminación.



similitud



## Filtrado y Convolución

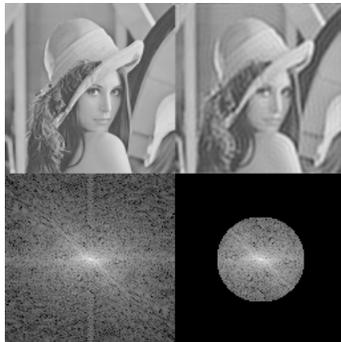


filtrado pasa bajos  
para eliminar ruido

(PD: también usado como etapa previa en detección de bordes y en cambio de resolución de imagen)



filtrado pasa altos  
para detección de bordes



filtrado pasa bajos  
para eliminar ruido

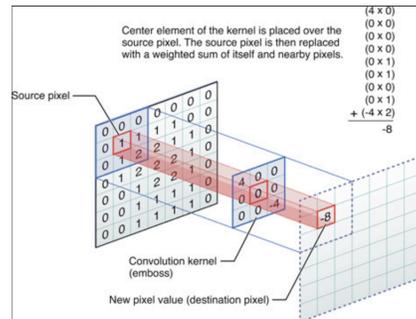
(PD: también usado como etapa previa en detección de bordes y en cambio de resolución de imagen)



filtrado pasa altos  
para detección de bordes

Source: <https://www.math.utah.edu/~gustafso/s2014/3150/pdeNotes/fourierTransform-in-image-processing-john-brayer-univ-new-mexico.html>

- Una operación de filtrado lineal es equivalente a una operación de convolución -> todo filtro que se pueda aplicar mediante convolución es lineal
- En el caso de las imágenes la función a ser filtrada es 2D y el filtro/kernel (operador de convolución) puede ser 1D o 2D.



Si filtro no lineal no hay kernel. Se aplica ventana deslizante y luego op. no lineal (mediana, máximo, etc.) en pixeles bajo ventana.

```

for each image row in input image:
  for each pixel in image row:
    set accumulator to zero
    for each kernel row in kernel:
      for each element in kernel row:
        multiply kernel value to corresponding* pixel value
        add result to accumulator
    set output image pixel to accumulator

```

\*corresponding input image pixels are found relative to the kernel's origin.

```

// find center position of kernel (half of kernel size)
kCenterX = kCols / 2;
kCenterY = kRows / 2;

for(i=0; i < rows; ++i) // rows
{
  for(j=0; j < cols; ++j) // columns
  {
    for(m=0; m < kRows; ++m) // kernel rows
    {
      mm = kRows - 1 - m; // row index of flipped kernel

      for(n=0; n < kCols; ++n) // kernel columns
      {
        nn = kCols - 1 - n; // column index of flipped kernel

        // index of input signal, used for checking boundary
        ii = i + (m - kCenterY);
        jj = j + (n - kCenterX);

        // ignore input samples which are out of bound
        if( ii >= 0 && ii < rows && jj >= 0 && jj < cols )
          out[i][j] += in[ii][jj] * kernel[mm][nn];
      }
    }
  }
}

```

Filtrado pasa-bajo, distintos posibles filtros.  
 -> distinto tipo de filtro, distinto tamaño del kernel

**X** No se puede mostrar la imagen. Puede que su equipo no tenga suficiente memoria para abrir la imagen o que ésta esté dañada. Reinicie el equipo y, a continuación, abra el archivo de nuevo. Si sigue apareciendo la x roja, puede que tenga que borrar la imagen e insertarla de nuevo.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

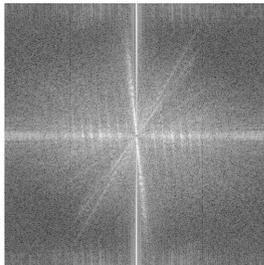
**X** No se puede mostrar la imagen. Puede que su equipo no tenga suficiente memoria para abrir la imagen o que ésta esté dañada. Reinicie el equipo y, a continuación, abra el archivo de nuevo. Si sigue apareciendo la x roja, puede que tenga que borrar la imagen e insertarla de nuevo.

**X** No se puede mostrar la imagen. Puede que su equipo no tenga suficiente memoria para abrir la imagen o que ésta esté dañada. Reinicie el equipo y, a continuación, abra el archivo de nuevo. Si sigue apareciendo la x roja, puede que tenga que borrar la imagen e insertarla de nuevo.

Imagen en dominio espacial



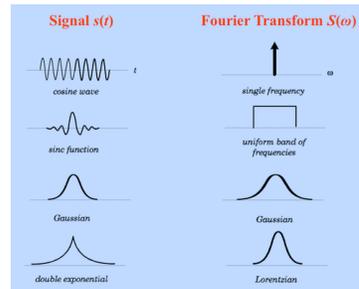
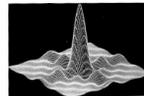
Imagen en dominio espectral



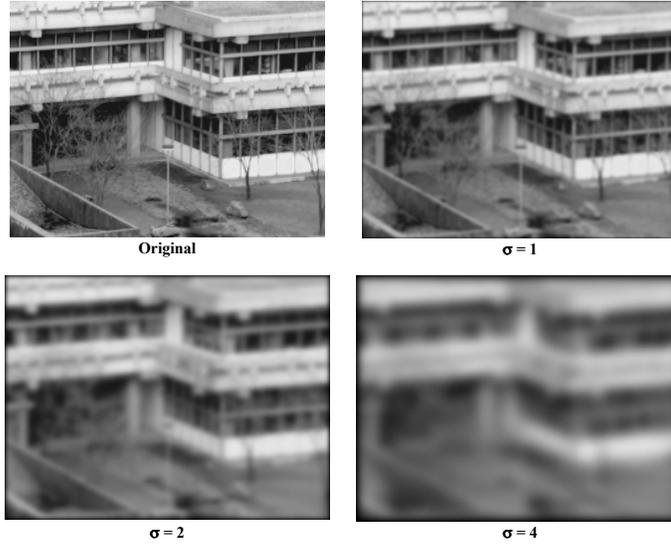
filtro/operador



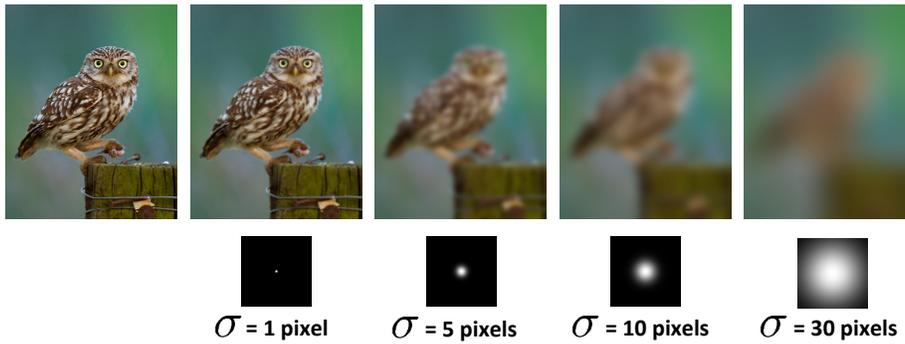
$$\mathcal{F}^{-1}\{f(x,y)\}$$



Ancho de banda del filtro depende del tamaño del kernel  
 -> a mayor tamaño menor ancho de banda

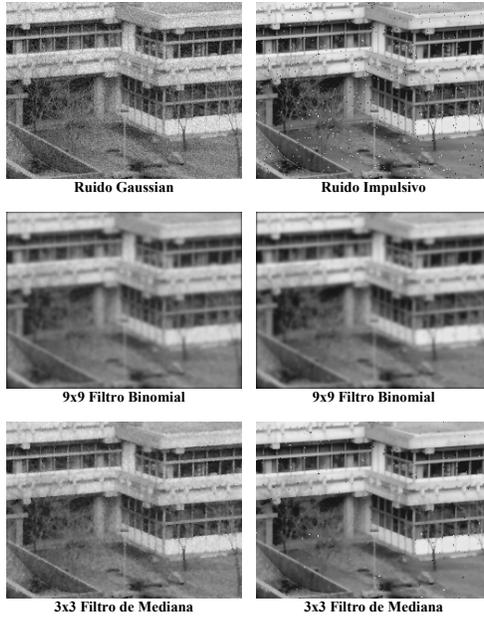


## Gaussian filters

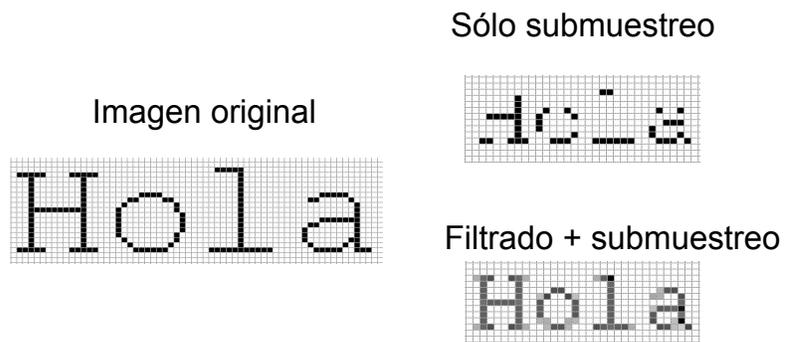


Tomado de CS6670: Computer Vision, Noah Snavely.

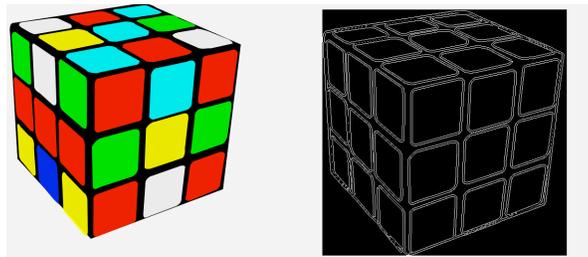
Filtro pasa-bajos (lineal) no necesariamente sirve para todo tipo de ruido.



Filtrado pasa-bajo útil cuando se disminuye resolución de imágenes  
 -> se debe filtrar pasa-bajos antes de disminuir resolución de las imágenes

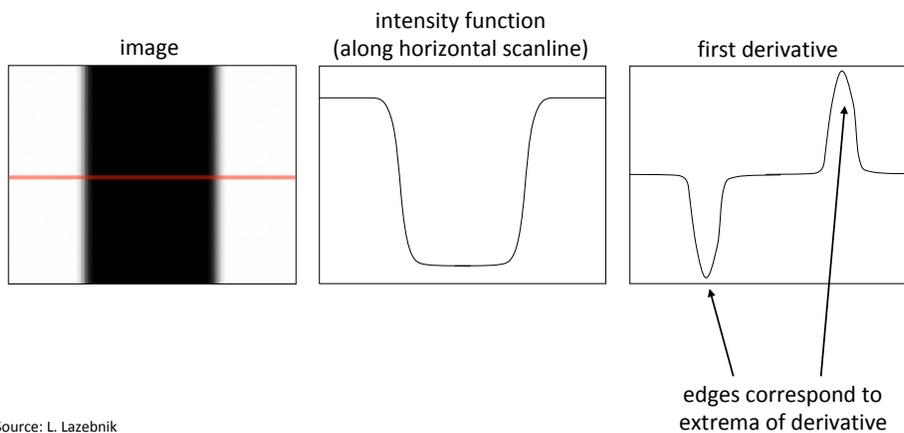


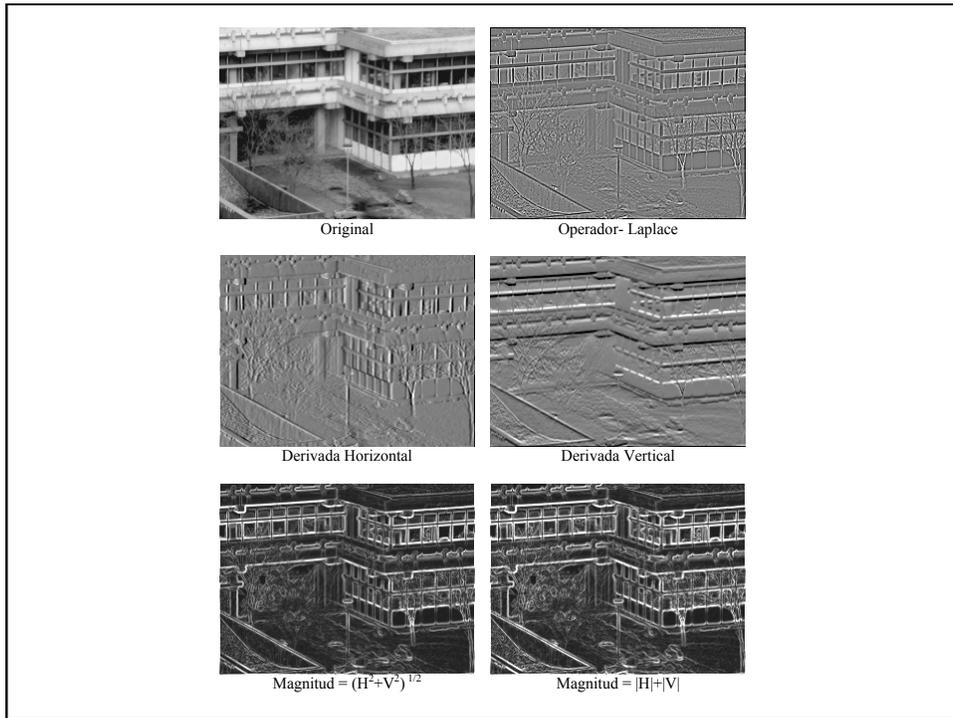
## Detección Robusta de Bordes



### Characterizing edges

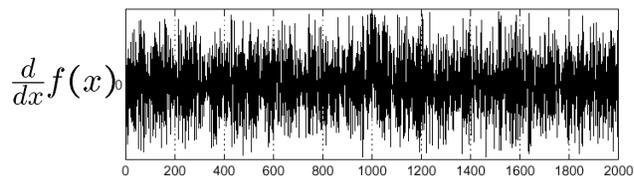
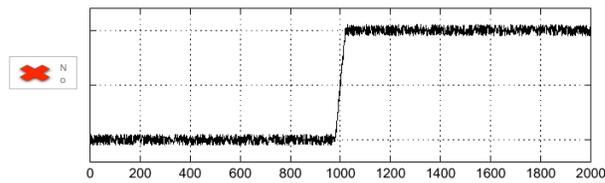
- An edge is a place of rapid change in the image intensity function





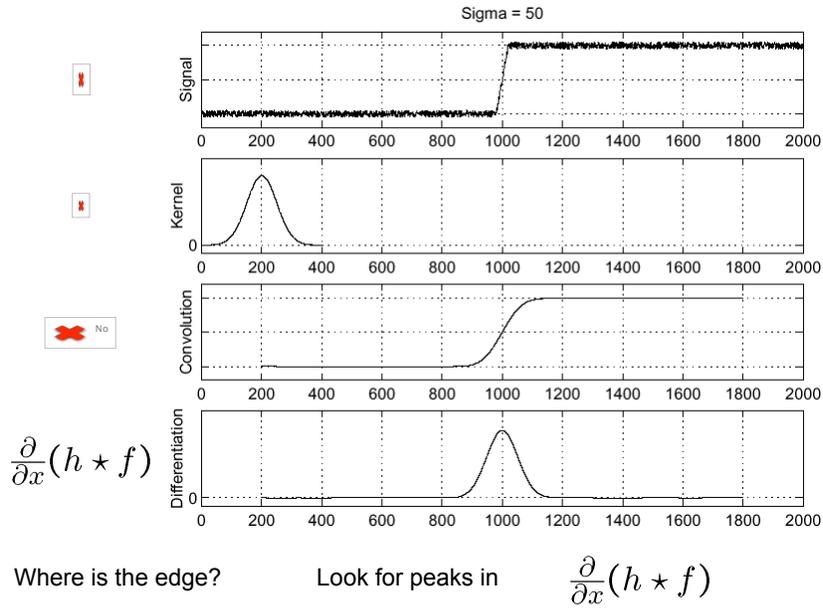
### Effects of noise

Consider a single row or column of the image



Where is the edge?

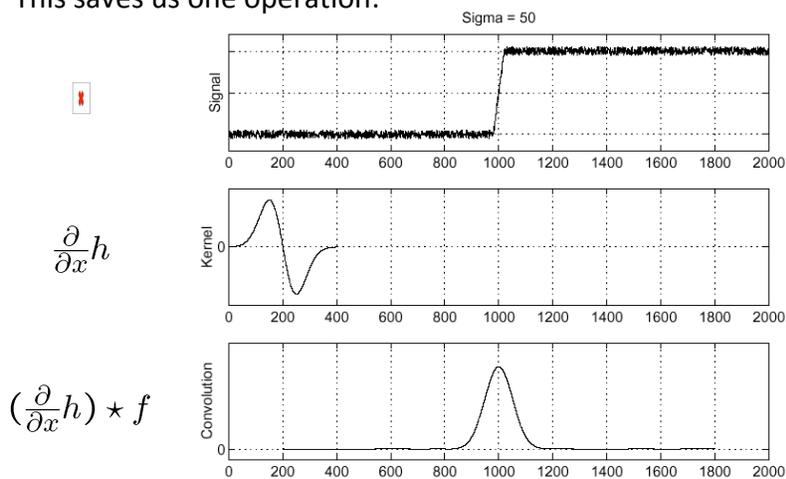
### Solution: smooth first



### Derivative theorem of convolution

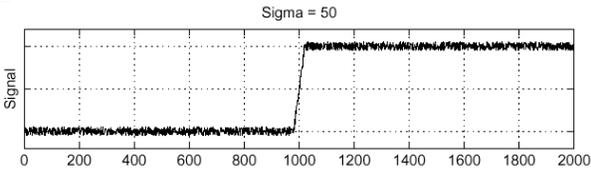
$$\frac{\partial}{\partial x}(h \star f) = \left(\frac{\partial}{\partial x}h\right) \star f$$

This saves us one operation:

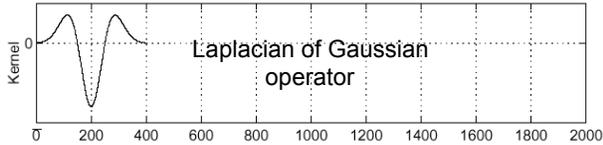


## Laplacian of Gaussian

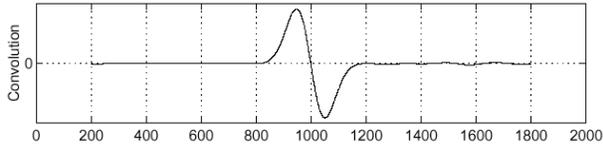
- Consider  $\frac{\partial^2}{\partial x^2}(h \star f)$



$$\frac{\partial^2}{\partial x^2}h$$



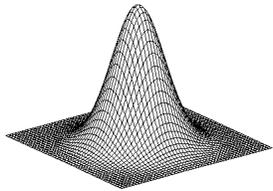
$$\left(\frac{\partial^2}{\partial x^2}h\right) \star f$$



Where is the edge?

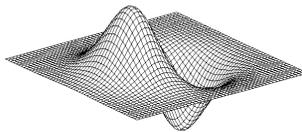
Zero-crossings of bottom graph

## 2D edge detection filters



Gaussian

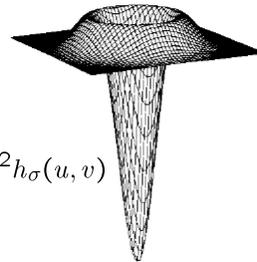
$$h_\sigma(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x}h_\sigma(u, v)$$

Laplacian of Gaussian



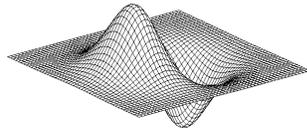
$$\nabla^2 h_\sigma(u, v)$$



The **Laplacian** operator:

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

## Aproximación de operadores mediante máscaras/kernels



derivative of Gaussian

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

Sobel Operator

$\frac{1}{8}$	-1	0	1
	-2	0	2
	-1	0	1

$s_x$

$\frac{1}{8}$	1	2	1
	0	0	0
	-1	-2	-1

$s_y$

3x3 mask, horizontal gradient

$$\frac{df}{dx} = f(x) - f(x-1) = f'(x) \quad \text{Backward difference}$$

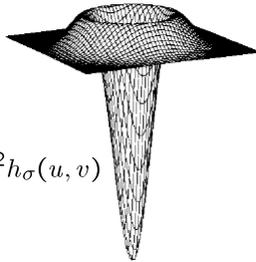
$$\frac{df}{dx} = f(x) - f(x+1) = f'(x) \quad \text{Forward difference}$$

$$\frac{df}{dx} = f(x+1) - f(x-1) = f'(x) \quad \text{Central difference}$$

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

## Aproximación de operadores mediante máscaras/kernels

Laplacian of Gaussian



$$\nabla^2 h_{\sigma}(u, v)$$

Aproximaciones de tamaño 3x3

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Aproximación de tamaño 9x9

0	0	3	2	2	2	3	0	0
0	2	3	5	5	5	3	2	0
3	3	5	3	0	3	5	3	3
2	5	3	-12	-23	-12	3	5	2
2	5	0	-23	-40	-23	0	5	2
2	5	3	-12	-23	-12	3	5	2
3	3	5	3	0	3	5	3	3
0	2	3	5	5	5	3	2	0
0	0	3	2	2	2	3	0	0