



22.90 – AUTOMACIÓN INDUSTRIAL

Visión: Histogramas y tipos de funciones

OBJETIVO

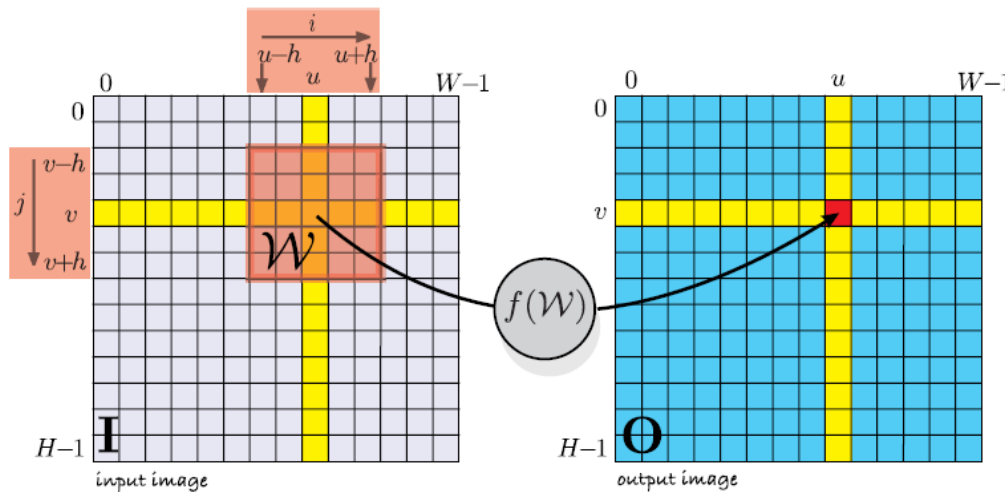
En esta clase, revisaremos las principales funciones espaciales dentro del campo de la visión por computadora

Uno de los conceptos mas importantes de esta clase será la convolucion, que es un tipo particular de operación que permite realizar diversas tareas, como filtrado de ruido y detección de bordes.

Además, revisaremos algunas herramientas de comparación.

FUNCIÓN ESPACIAL

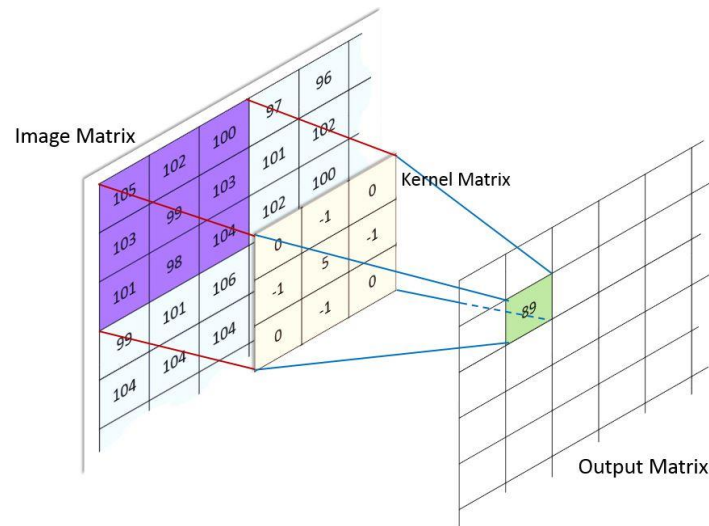
Este tipo de funciones, calculan el valor de cada pixel en la imagen de salida como una función de los valores de los pixeles en un rango de la imagen de entrada.



$$O[u, v] = f(I[u + i, v + j]), \quad \forall (i, j) \in W, \quad \forall (u, v) \in I$$

KERNELS

- Matriz de peso de dimensión (n,m), que utilizaremos como parte de las funciones espaciales.
- La dimensión y los valores de la matriz varían dependiendo del objetivo de la operación que se aplique.



CONVOLUCION

- Operador espacial de tipo LINEAL
- Descripción Matemática

$$O[u, v] = \sum_{(i,j) \in \mathcal{W}} I[u + i, v + j] K[i, j], \quad \forall (u, v) \in \mathbf{I}$$

$$\mathbf{O} = \mathbf{K} \otimes \mathbf{I}$$

- Aplicación en Vision Toolbox de Peter Corke:

Output = iconv (K , Input)

CONVOLUCION

○ Propiedades

Properties of convolution. Convolution obeys the familiar rules of algebra, it is commutative

$$A \otimes B = B \otimes A$$

associative

$$A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)$$

distributive (superposition applies)

$$A \otimes (B + C) = A \otimes B + A \otimes C$$

linear

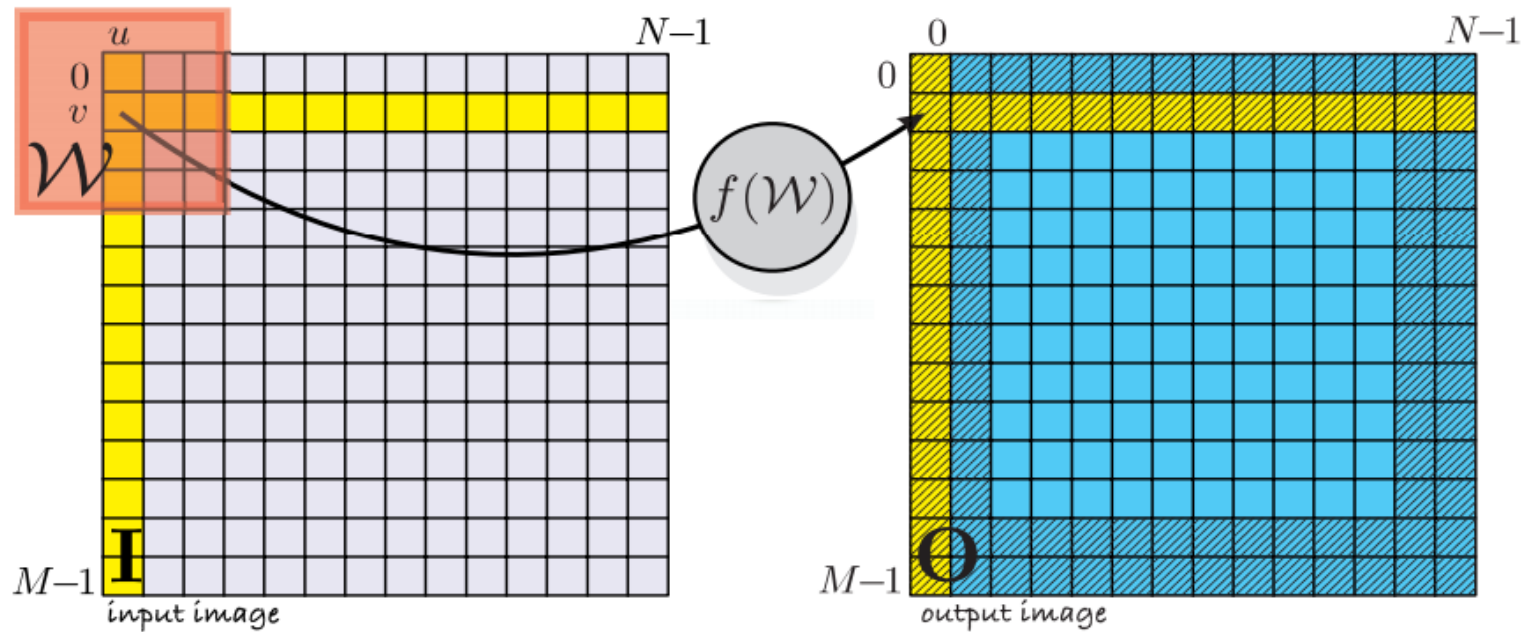
$$A \otimes (\alpha B) = \alpha(A \otimes B)$$

and shift invariant – if $S(\cdot)$ is a spatial shift then

$$A \otimes S(B) = S(A \otimes B)$$

that is, convolution with a shifted image is the same as shifting the result of the convolution with the unshifted image.

CONVOLUCION: BORDES DE LA IMAGEN



FILTRADO

Una importante aplicación de la convolucion es el filtrado de ruido en las imágenes.

Muchas imágenes poseen una componente de ruido, la cual puede aproximarse como una función armónica de una cierta frecuencia y amplitud.

Este ruido dificulta la tarea de extracción de información, al generar saltos en la imagen que no existen en la realidad

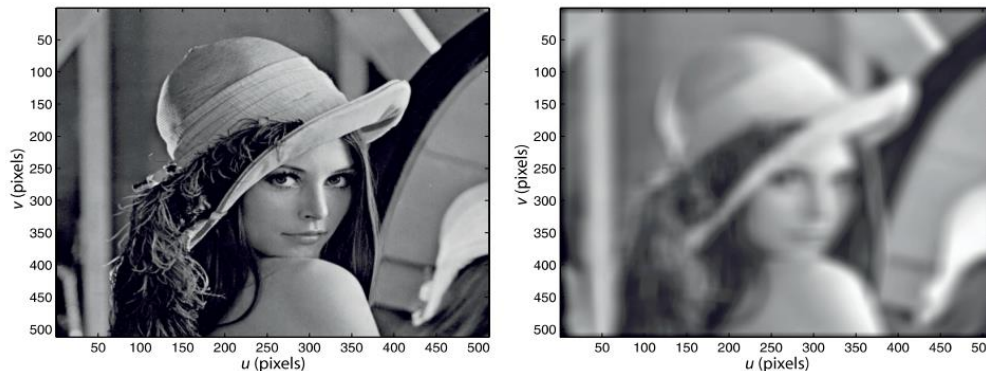


FILTRADO: PROMEDIO

Si aplicamos a una imagen un kernel K de $n \times n$, donde cada valor vale $1/n^2$, por ejemplo:

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Obtendríamos una imagen de salida, en la que cada pixel se calcula como el promedio de los valores de intensidad en un área de $n \times n$.

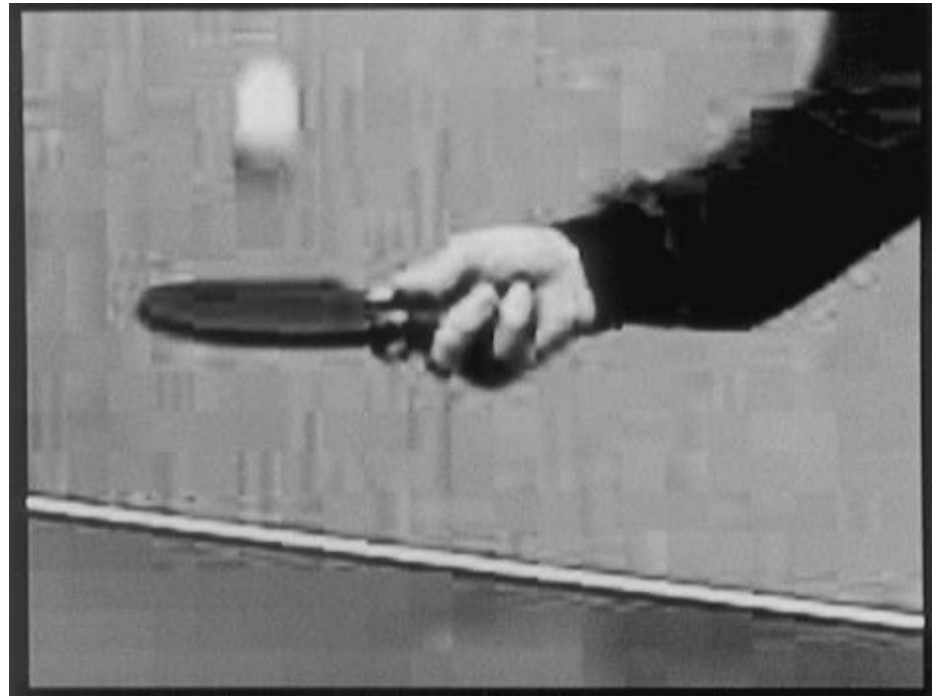


FILTRADO: PROMEDIO

Este tipo de filtrado, si bien es capaz de eliminar gran parte del ruido, genera un defecto en las imágenes conocido como 'ringing' (líneas horizontales-verticales).

Este efecto se genera por la distribución de pesos en el kernel propuesto.

Se le da la misma ponderación a todos los pixels del área, sin importar si están cerca o lejos del pixel de interés.



FILTRADO: KERNEL GAUSSIANO

Una matriz de pesos que no sufre esta desventaja es la matriz gaussiana.

- Descripción matemática

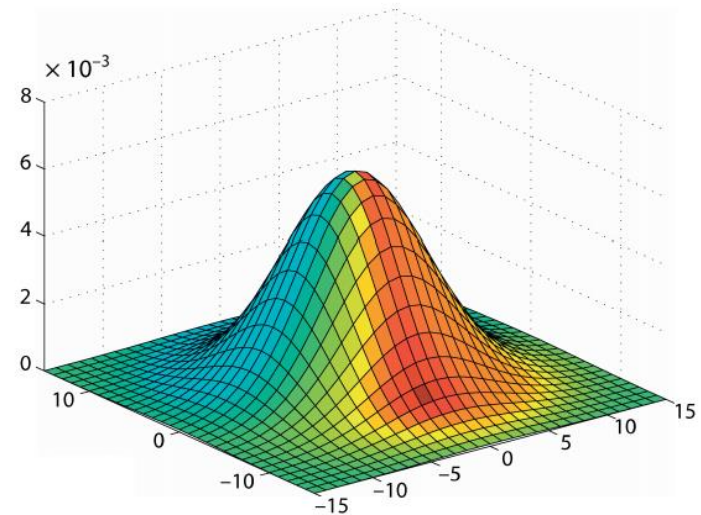
$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

- Función en MATLAB

`K=kgauss(σ)`

`Size(K)=(w,w)`

Regla general (1%): $3*\sigma = h \ // \ 2*h+1=w$



FILTRADO: KERNEL GAUSSIANO

○ Propiedades

Properties of the Gaussian. The Gaussian function has some special properties. The convolution of two Gaussians is another Gaussian

$$G(\sigma_1) \otimes G(\sigma_2) = G\left(\sqrt{\sigma_1^2 + \sigma_2^2}\right)$$

For the case where $\sigma_1 = \sigma_2 = \sigma$ then

$$G(\sigma) \otimes G(\sigma) = G(\sqrt{2}\sigma)$$

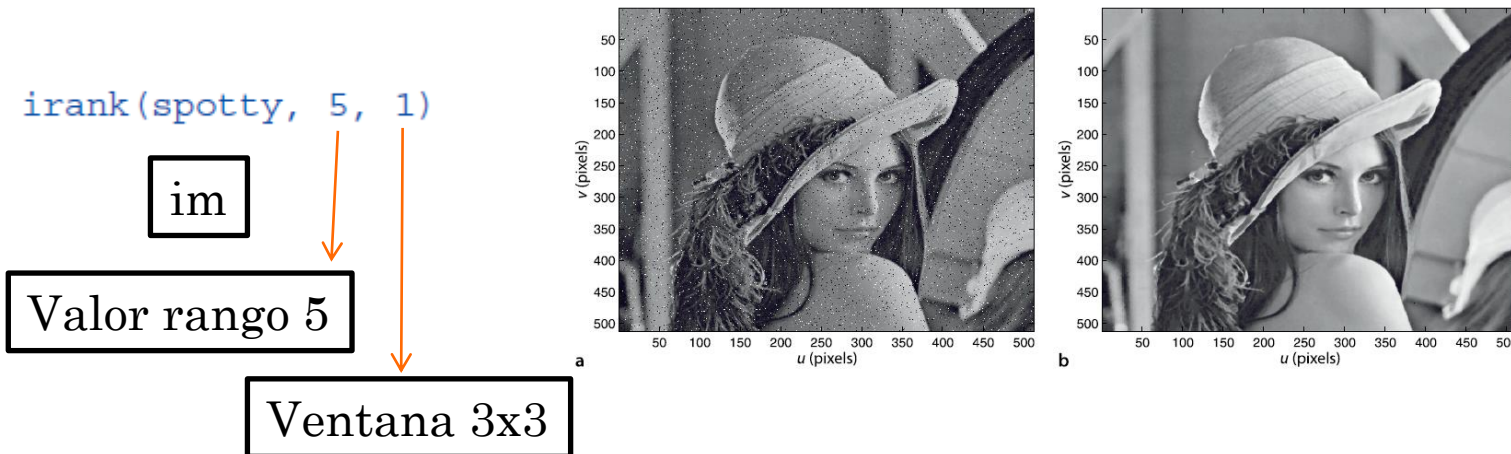
The 2-dimensional Gaussian is separable – it can be written as the product of two 1-dimensional Gaussians

$$G(u, v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u^2}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{v^2}{2\sigma^2}}$$

This implies that convolution with a 2-dimensional Gaussian can be computed by convolving each row with a 1-dimensional Gaussian, and then each column. The total number of operations is reduced to $2wN^2$, better by a factor of w . A Gaussian also has the same shape in the spatial and frequency domains.

FILTROS NO LINEALES

- Existe un tipo de ruido llamado «salt and pepper», el cual se caracteriza por una elevada amplitud (unos y ceros en la imagen en posiciones aleatorias). Existen técnicas no lineales que permiten filtrar este efecto, sin la necesidad de perder definición en la imagen, ni ‘contaminar’ el promedio con el valor del ruido.



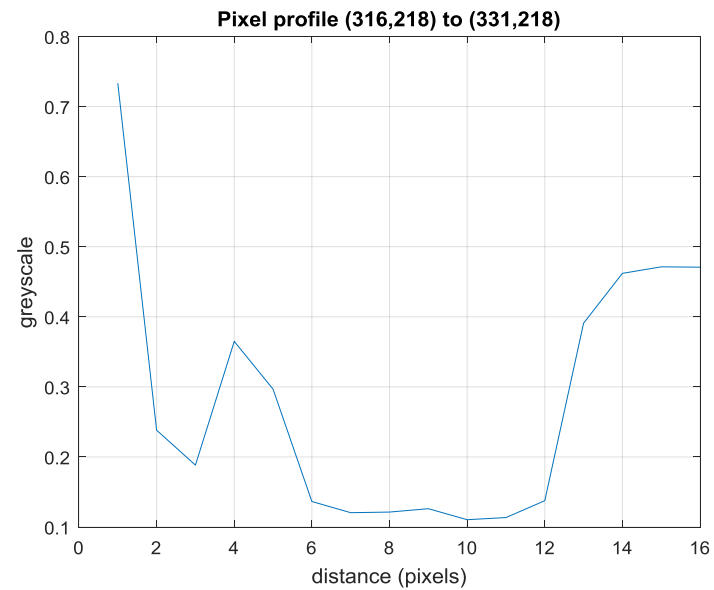
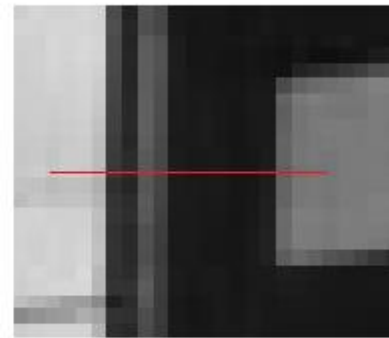
DETECCIÓN DE BORDES

Un área de interés en la visión por computadora, es la detección de bordes en las imágenes.

Existen muchos métodos para encontrar la interfase entre las diferentes figuras que componen una imagen. En términos generales, los métodos buscan resaltar las zonas en las se ven diferencias marcadas entre las intensidades de los pixeles cercanos.

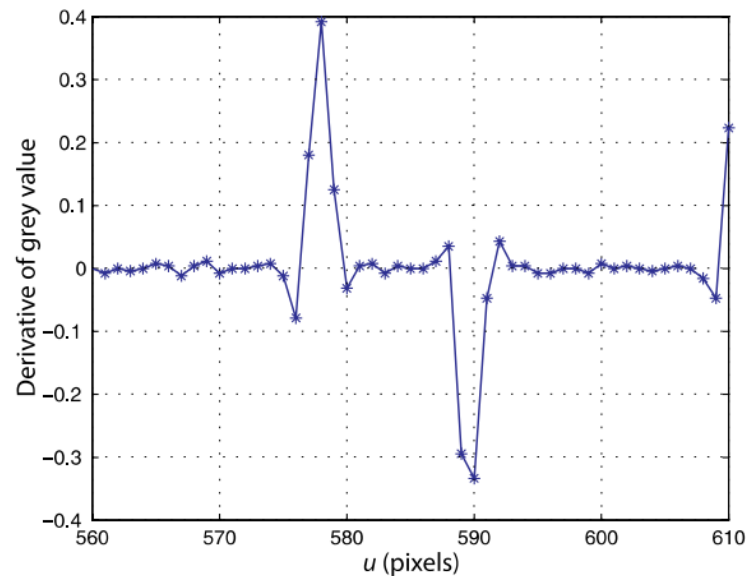
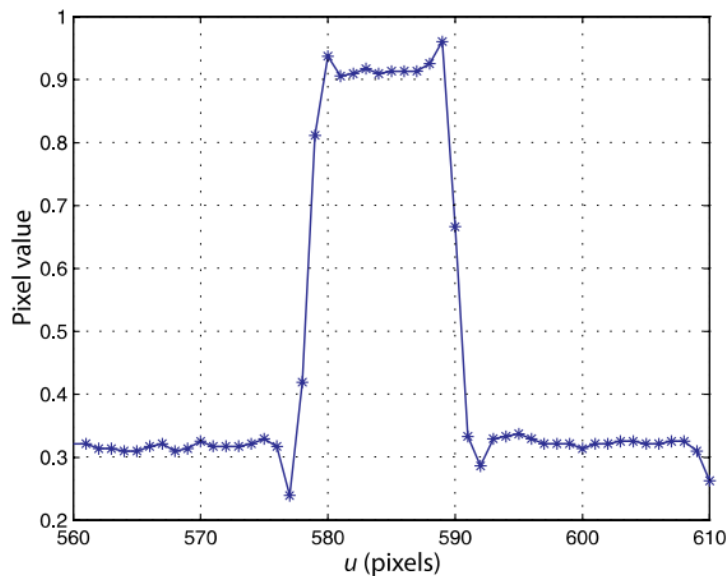
Veamos en el siguiente ejemplo, la variación de intensidades a lo largo de una línea recta:

DETECCIÓN DE BORDES



DETECCIÓN DE BORDES

- Una forma sencilla de computar las variaciones de la intensidad sería calcular la derivada puntual de la intensidad, como función de la posición.



DETECCIÓN DE BORDES

Una forma de calcular esta derivada en un espacio donde las posiciones son numeros naturales podria ser la resta entre valores sucesivos. O tambien para mayor comodidad, la diferencia simetrica dividida por dos:

$$p'[v] = \frac{1}{2}(p[v+1] - p[v-1])$$

En forma de Kernel (para usar en convolucion):

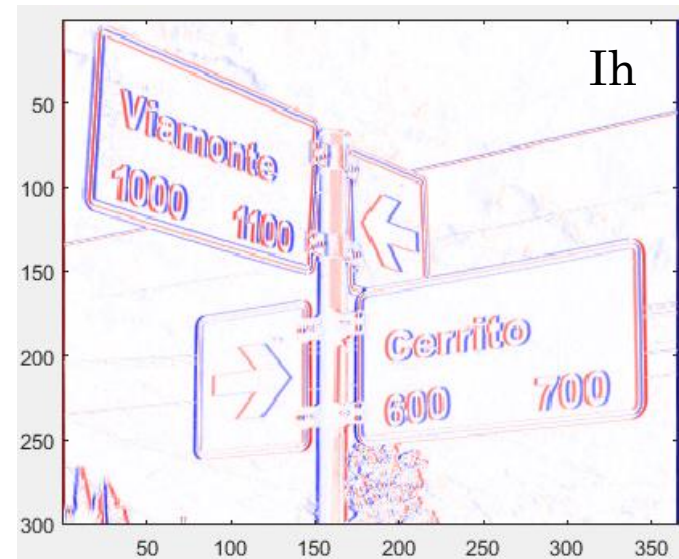
$$K = \begin{pmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

Ver también:
ksobel()

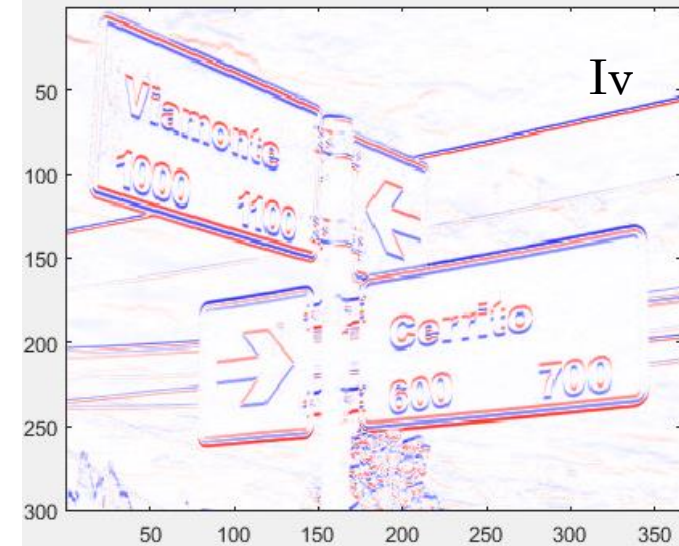
DETECCIÓN DE BORDES



K_x



K'_x

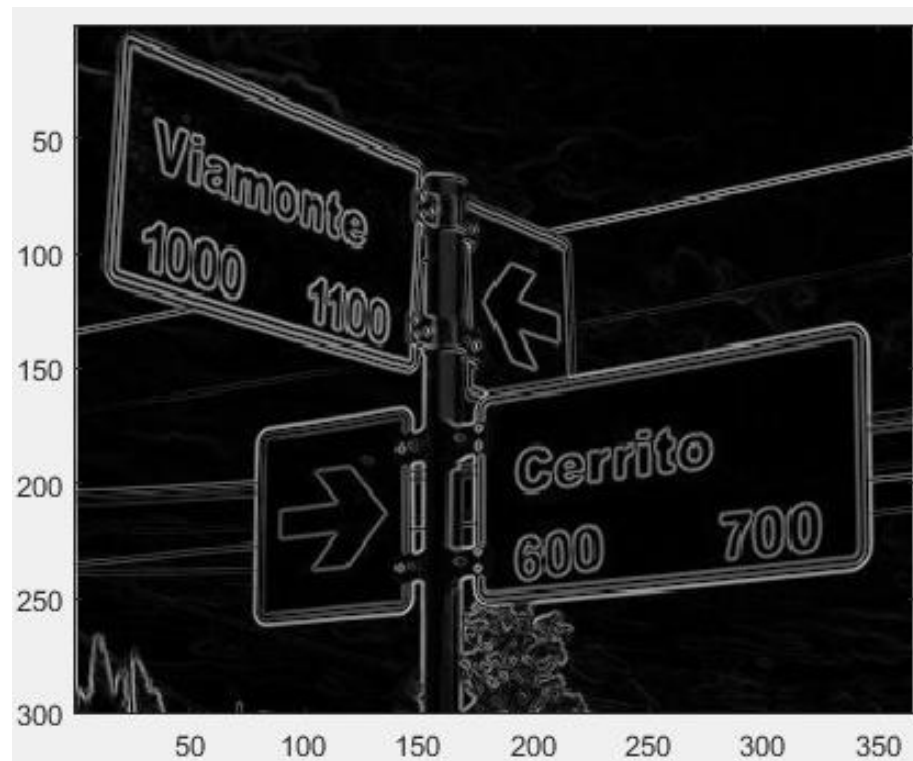


DETECCIÓN DE BORDES

Eliminamos valores negativos, y resaltamos todos los bordes calculando la norma punto a punto como:

$\text{Sqrt}(I_v^2 + I_h^2)$ // Elemento a elemento

Este método es
muy sensible al
ruido



KERNELS DE DOBLE EFECTO

En la mayoría de los casos, será de interés filtrar una imagen antes de empezar a trabajar con ella. Para realizar una menor cantidad de operaciones, utilizaremos la propiedad asociativa de la convolucion:

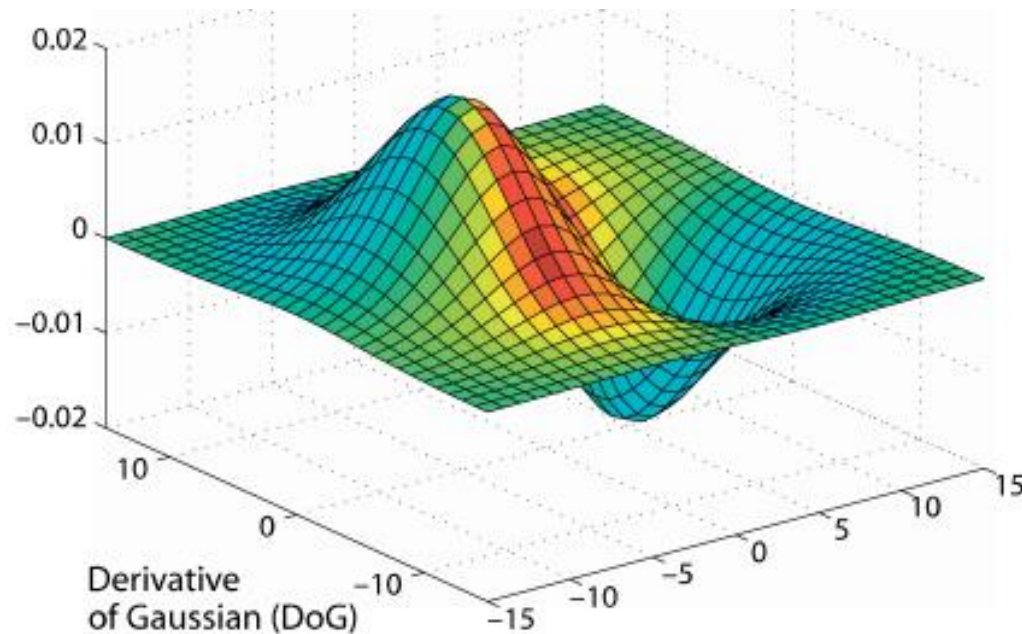
$$\nabla I = \mathbf{D} \otimes (\mathbf{G}(\sigma) \otimes I) = \underbrace{(\mathbf{D} \otimes \mathbf{G}(\sigma))}_{\text{DoG}} \otimes I$$

De esta forma, en vez de aplicar un Kernel derivativo sobre una imagen filtrada, utilizaremos una matriz derivativa sobre un kernel de filtro, para luego aplicar el conjunto a la imagen.

DERIVADA DEL GAUSSIANO

- Descripción Matemática

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



DERIVADAS DE MAYOR ORDEN

Una alternativa para encontrar las zonas donde el gradiente de intensidad es alto, es calcular la segunda derivada y buscar los valores cero.

Esto puede calcularse con el operador de Laplace:

$$\nabla^2 I = \frac{\partial^2 I}{\partial u^2} + \frac{\partial^2 I}{\partial v^2} = I_{uu} + I_{vv}$$

Este operador puede computarse como Kernel con la función `klaplace()` del Toolbox de Corke:

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

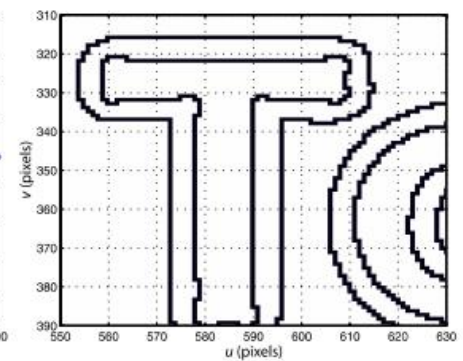
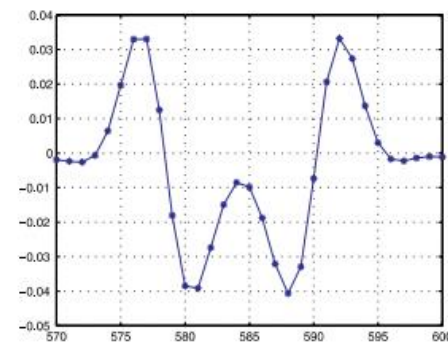
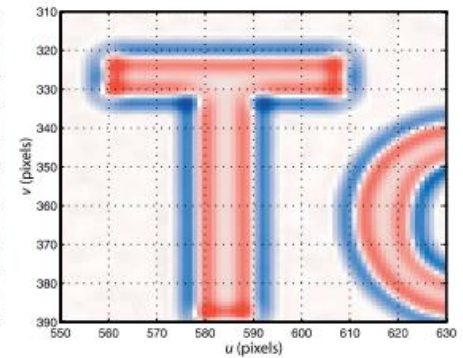
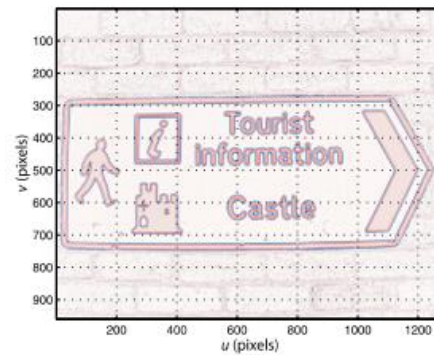
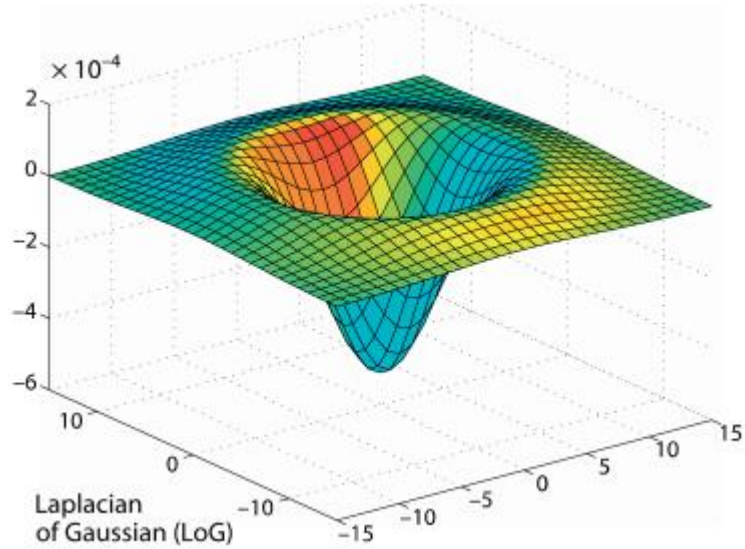
LAPLACIANO DEL OP. GAUSSIANO

De la misma forma que combinamos operaciones para obtener el kernel de doble efecto DoG, podemos combinar el efecto del kernel de Laplace, con el efecto del kernel filtro de Gauss, obteniendo así el operador de Marr-Hildreth (computable como $\text{klog}(\sigma)$):

$$\nabla^2 \mathbf{I} = \mathbf{L} \otimes (\mathbf{G}(\sigma) \otimes \mathbf{I}) = \underbrace{(\mathbf{L} \otimes \mathbf{G}(\sigma))}_{\text{LoG}} \otimes \mathbf{I}$$

$$\begin{aligned} \text{LoG}(u, v) &= \frac{\partial^2 \mathbf{G}}{\partial u^2} + \frac{\partial^2 \mathbf{G}}{\partial v^2} \\ &= \frac{1}{\pi \sigma^4} \left(\frac{u^2 + v^2}{2\sigma^2} - 1 \right) e^{-\frac{u^2 + v^2}{2\sigma^2}} \end{aligned}$$

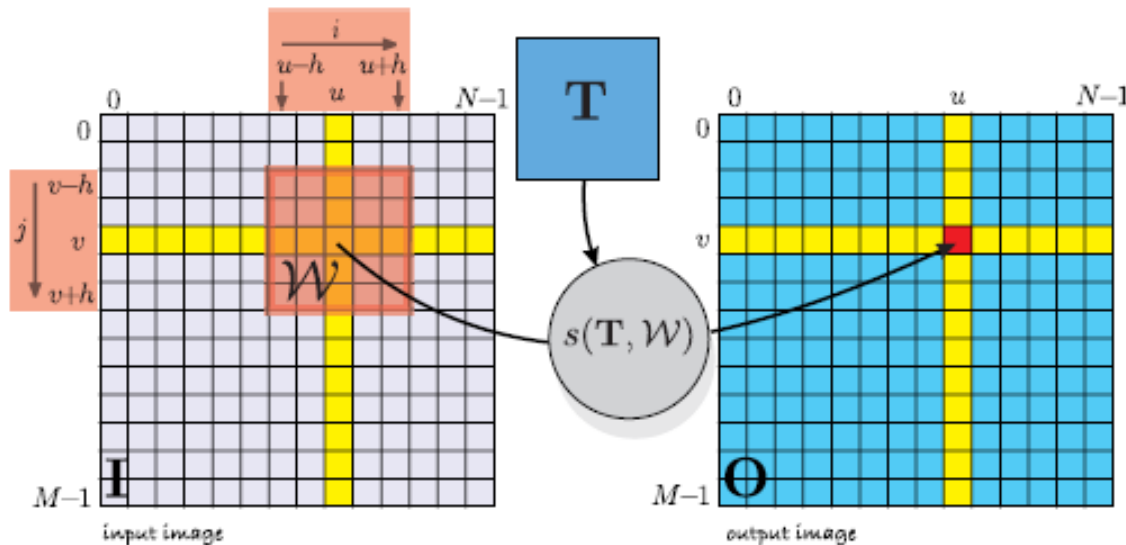
LoG



COMPARACION

‘Template Matching’ es una técnica utilizada para comparar dos imágenes, y calcular el grado de similitud entre ellas.

Una aplicación para este tipo de técnicas, es generar una matriz/imagen T (template), y comparar el grado de similitud en cada posible sección (W) de una imagen I , y generar una matriz de salida O que contenga el resultado de cada comparación.



TÉCNICAS DE COMPARACIÓN

Sum of absolute differences

SAD	$s = \sum_{(u,v) \in I} I_1[u, v] - I_2[u, v] $	sad
-----	--	-----

ZSAD	$s = \sum_{(u,v) \in I} (I_1[u, v] - \bar{I}_1) - (I_2[u, v] - \bar{I}_2) $	zsad
------	--	------

Sum of squared differences

SSD	$s = \sum_{(u,v) \in I} (I_1[u, v] - I_2[u, v])^2$	ssd
-----	--	-----

ZSSD	$s = \sum_{(u,v) \in I} ((I_1[u, v] - \bar{I}_1) - (I_2[u, v] - \bar{I}_2))^2$	zssd
------	--	------

Cross correlation

NCC	$s = \frac{\sum_{(u,v) \in I} I_1[u, v] \cdot I_2[u, v]}{\sqrt{\sum_{(u,v) \in I} I_1^2[u, v] \cdot \sum_{(u,v) \in I} I_2^2[u, v]}}$	ncc
-----	---	-----

ZNCC	$s = \frac{\sum_{(u,v) \in I} (I_1[u, v] - \bar{I}_1) \cdot (I_2[u, v] - \bar{I}_2)}{\sqrt{\sum_{(u,v) \in I} (I_1[u, v] - \bar{I}_1)^2 \cdot \sum_{(u,v) \in I} (I_2[u, v] - \bar{I}_2)^2}}$	zncc
------	---	------

TÉCNICAS DE COMPARACIÓN

SAD – Valor 0 para identidad, y crece a medida que difiere. Difícil de medir grado de similitud sin rankear con el resto de los valores.

Este método es sensible a cambios de intensidad generales (tanto de offset como proporcionales).

Cross Correlation – Valores entre $[-1,1]$, con 1 para sectores idénticos. Resiste modificaciones proporcionales en la intensidad.

Índice Z – Resta de cada imagen la intensidad media, de este modo elimina el offset de intensidad que estas pudieran presentar.

TÉCNICAS DE COMPARACIÓN

- Todos estos métodos fallan con modificaciones en tamaño o rotación de la imagen.
- Es normal que los templates que generemos sean matrices cuadradas, no así la imagen de interés, con lo cual no solo comparo la imagen que busco, sino también el fondo.



- Estos métodos tampoco manejan las diferencias que puedan surgir en la imagen por perspectiva.
- Ver también técnicas no paramétricas.

TÉCNICAS DE COMPARACIÓN

Se presenta a continuación un ejemplo de aplicación en el Toolbox de Corke

```
>> S = isimilarity(T, crowd, @zncc);  
>> [mx,p] = peak2(S, 1, 'npeaks', 5);  
>> mx  
mx =  
    0.5258  
    0.5230  
    0.5222  
    0.5032  
    0.5023
```

Técnicas de selección:
Threshhold para grado de similitud, diferencia entre candidatos, información adicional particular del problema (cámara, movimiento, etc)

