

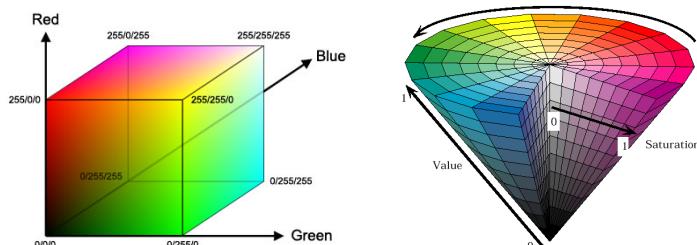
## Tema 8: Introducción a los Sistemas para Percepción, orientando hacia Visión Artificial. Operaciones básicas: histograma, ruido, suavizado, Detección de aristas y puntos característicos.

## Índice

- Definiciones
- Convolución
- Filtrado
- Histograma de la imagen
- Aristas: algoritmo de Canny
- Detector de Nitzberg-Harris

## Modelos de color

- Dependiendo de los datos de la matriz de la imagen, tendremos distintos tipos de imagen
  - Binaria: valores 0 y 1
  - Gris: valores 0 a 255 (usualmente)
  - Color RGB: tres valores entre 0 y 255
- En cuanto al color, existen otros modelos (HSV)



Operaciones básicas

3

## Resolución de una imagen

- Una imagen tendrá un tamaño MxN
- Podemos ampliar o reducir la imagen.
  - Ampliar: se repiten píxeles
  - Reducir: al perder píxeles en la reducción, podemos: escoger un píxel o promediar entre los píxeles eliminados



Operaciones básicas

4

## Definiciones

- Cambio de brillo: aumenta o reduce el valor de cada píxel



- Ruido: errores que pueden degradar la imagen
  - Se produce: adquisición, transmisión o el procesado de la imagen
  - Diremos que un método de procesamiento es robusto ante el ruido, cuando genera los mismos resultados con y sin ruido

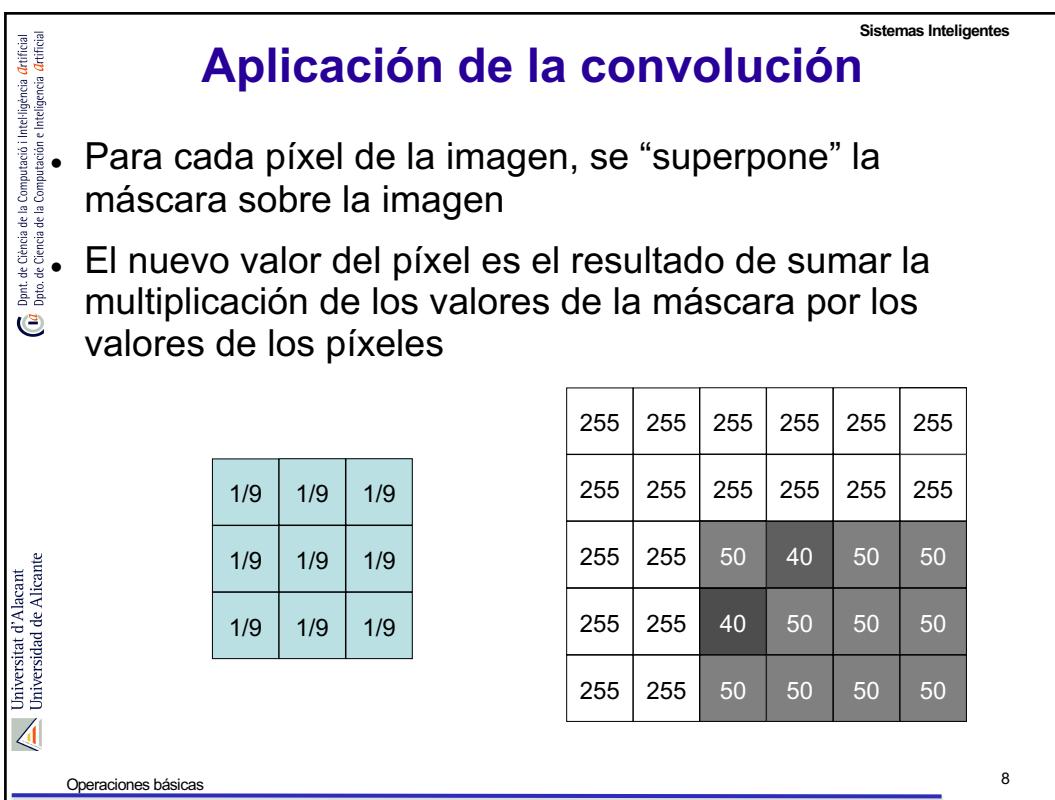
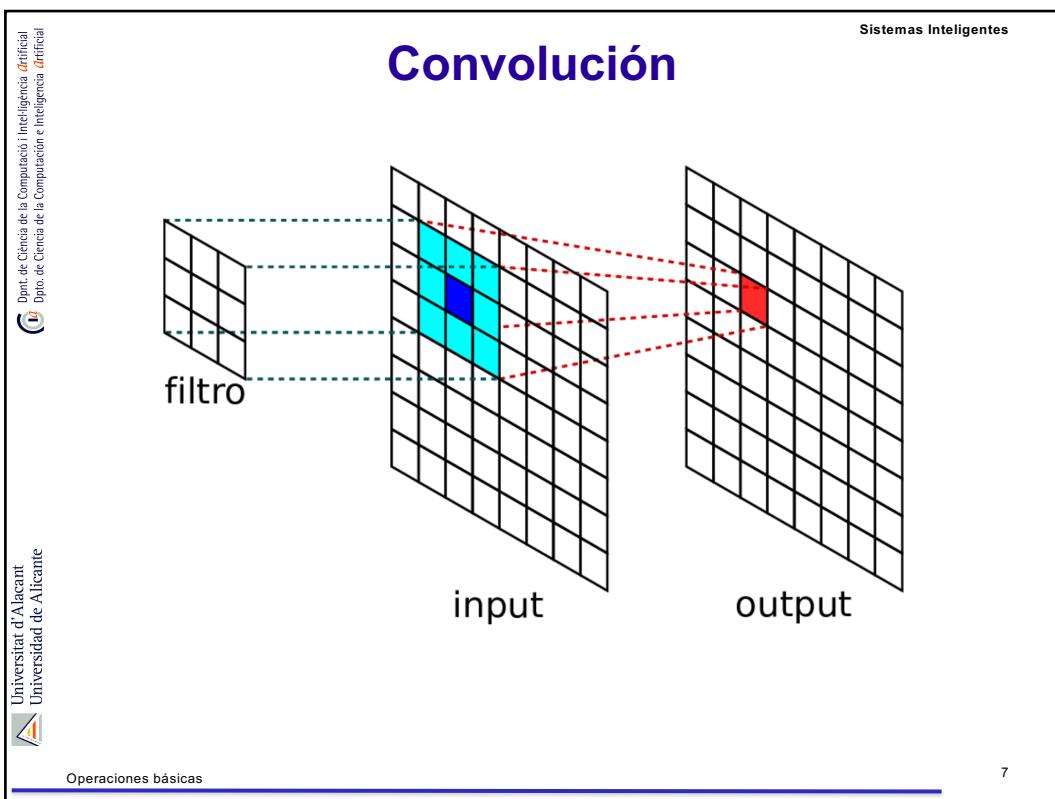
## Convolución

- Operación matemática aplicada a dos funciones,  $f(x)$  y  $g(x)$  que tiene como resultado otra función  $h(x)$
- Está definida, en el caso continuo, como:

$$f(x) * g(x) = h(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$$

- Y en el caso discreto y en dos dimensiones:

$$f(x, y) * g(x, y) = h(x, y) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} f(m, n)g(x-m, y-n)$$



## Filtros lineales

### Filtro de media

1/9

1	1	1
1	1	1
1	1	1

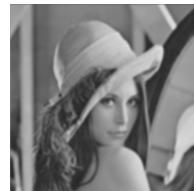
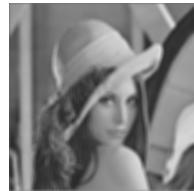
### Filtrado Gaussiano

- Crea un filtro de convolución con una función gaussiana de media 0 y varianza  $\sigma$

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

1/16

1	2	1
2	4	2
1	2	1



Original

Media

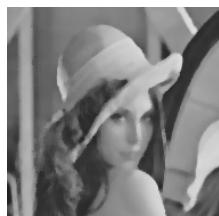
Gaussiano

## Filtro de mediana y resultados

- Filtro no lineal de mediana: calcula la mediana en una vecindad



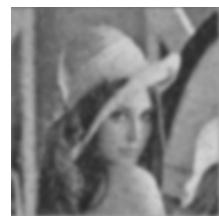
Original



Mediana



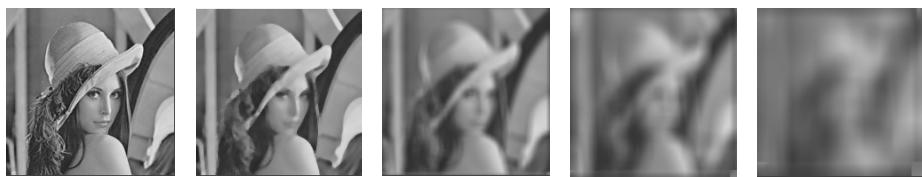
Media



Gaussiana

## Efecto del tamaño de la máscara

- Hay que tener en cuenta que el tamaño de la máscara usada o de la vecindad escogida ( $3 \times 3$ ,  $5 \times 5$ ,  $10 \times 10$ ) puede tener un efecto no deseado en el resultado, aunque hay veces que es lo que se pretende



## Histograma de la imagen

- Un histograma es una función definida en  $K$  niveles (los valores de intensidad de la imagen) e indica el número de píxeles en la imagen que tienen ese valor
- Si está normalizado (la suma de todos los valores es 1.0) nos indica la probabilidad de que un píxel tenga un determinado valor
- ¿Qué ocurre en el histograma si aumentamos el brillo?



## Operaciones con el histograma

- Binarización: indicar uno o dos umbrales y poner a 1 los píxeles que estén por debajo o por encima del umbral o entre dos umbrales, y a cero el resto
- También se puede hacer con imágenes de color, aplicando umbrales a cada banda de color



Binarización con  
umbrales 128 y  
255



## Ecualización del histograma

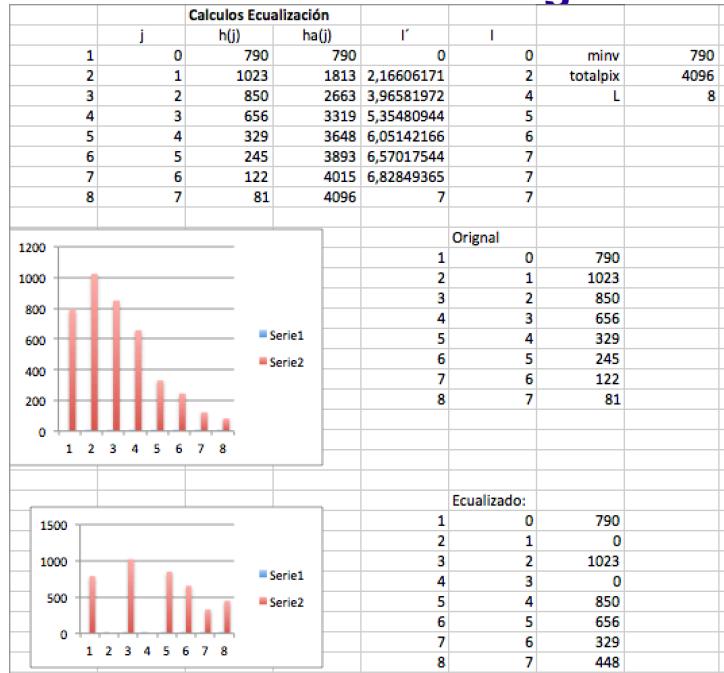
- Cuando tenemos una imagen con poco contraste podemos “expandir” el histograma
- El algoritmo es el siguiente:
  - Calculamos el histograma de la imagen  $h(k)$
  - Calculamos el histograma acumulado  $h_a(k) = \sum_{j=0}^k h(j)$
  - Se produce un mapeo con la siguiente fórmula:

$$I'(x, y) = \frac{h_a(I(x, y)) - \text{minv}}{\text{totalpix} - \text{minv}}(L - 1)$$

donde L es el número de niveles de intensidad y  $\text{minv}$  es el valor de  $h_a$  correspondiente al menor valor de gris.

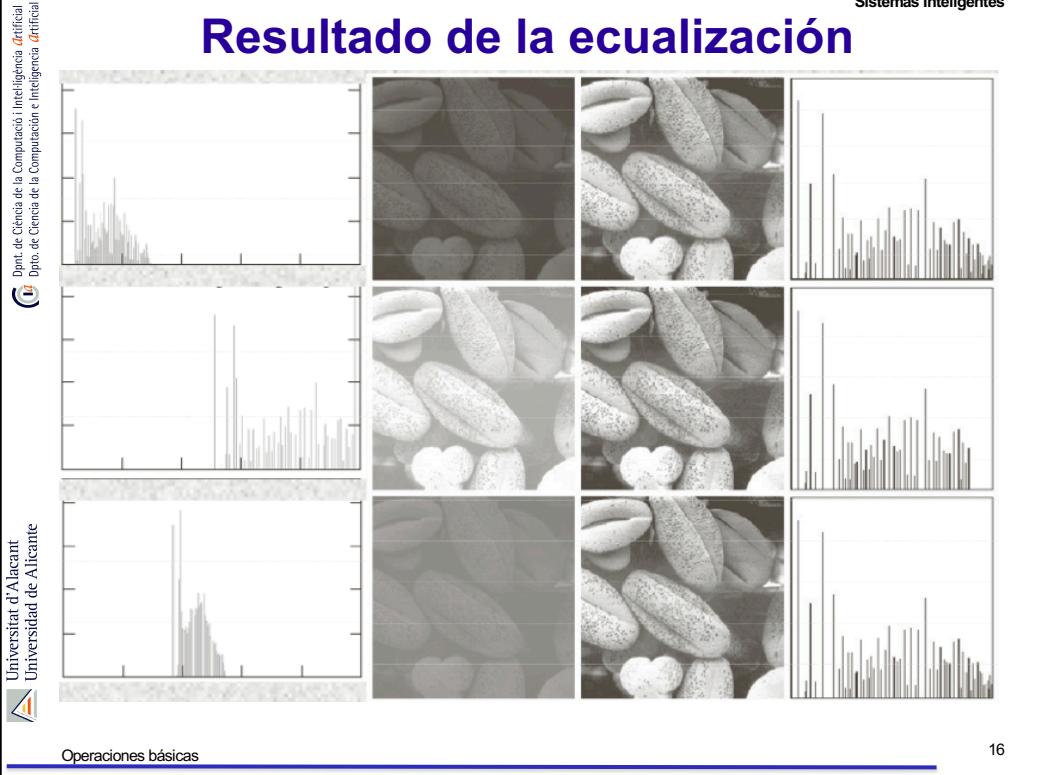


## Ecualización del histograma



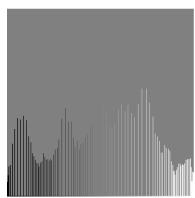
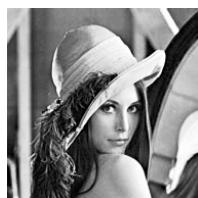
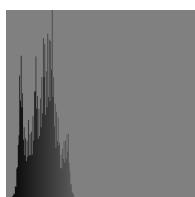
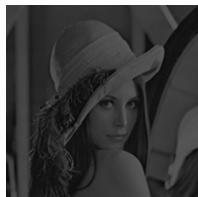
15

## Resultado de la ecualización



16

## Resultado de la ecualización



## Ejemplo

250	252	253	254	255	255
251	253	254	255	255	255
252	253	45	40	50	10
253	254	40	50	50	20
254	255	50	50	50	30

106	132	167	202	255	255
114	167	202	255	255	255
132	167	44	35	97	0
167	202	35	97	97	9
202	255	97	97	97	18

## Aristas (“Edges”)

- Edges:

- Puntos de alto contraste en la imagen, es decir donde la intensidad de la imagen aumenta o disminuye bruscamente.
- Edges=puntos de alta derivada en valor absoluto. Derivamos numéricamente la imagen (entendida como función bidimensional de variable real) para identificar las aristas.

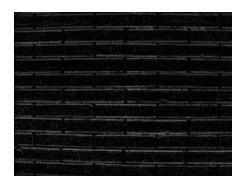
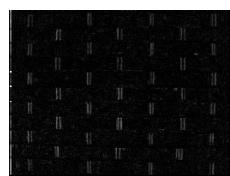
- Operador básico:

- Diferencias de primer orden.
- En dos fases: horizontal y vertical.

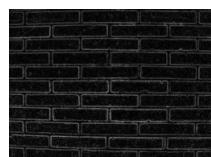
$$\nabla I = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} f(x+1, y) - f(x, y) \\ f(x, y+1) - f(x, y) \end{bmatrix}$$

## Aristas

- Operadores de gradiente:
- Diferencias de primer orden:



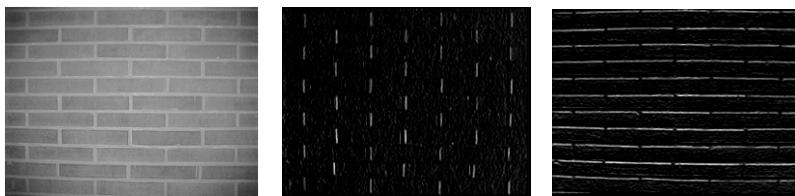
- Podemos calcular la magnitud y la orientación del gradiente



## Aristas

- Operadores de gradiente:
- Operadores de Sobel: (promediado)

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$



## Detector de Canny

### Fundamentos:

- Criterios (formalización):
  - Buena detección: minimizar el número de falsos positivos y falsos negativos.
  - Buena localización: las aristas deben marcarse en el lugar real.
  - Respuesta única: como resultado deben generarse aristas de un pixel de anchura.
- Supuestos:
  - Ruido gaussiano y aristas tipo “escalón”.
- Resultado:
  - Filtro óptimo: es la primera derivada de la gaussiana.

$$\nabla(G_\sigma * I) = \nabla(G_\sigma) * I$$

## Aristas (“Edges”)

### • Identificación de contornos:

- Detección:
  - Identificación de píxeles candidatos a formar parte de un contorno. Normalmente aquellos con “alto contraste local”.
  - Filtros lineales. Como veremos, existen máscaras de convolución que aplicadas sobre la imagen destacan los píxeles candidatos e inhiben el resto.
  - Selección o eliminación de no-máximos. Si resulta que hay más de un candidato en una misma vecindad hay que elegir uno de ellos y eliminar el resto.
- Agrupamiento:
  - Proceso de “local” a “global” que permite enlazar píxeles vecinos siempre y cuando se cumplan ciertas reglas de “continuidad” y “curvatura”.

## Detector de Canny. Resumen

El detector Canny es considerado, el más efectivo a la hora de detectar bordes, debido a su eficacia. Este detector, realiza su trabajo de la siguiente manera:

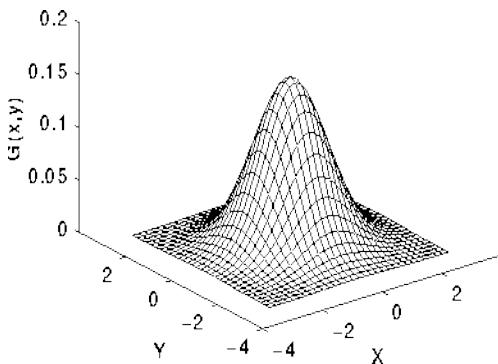
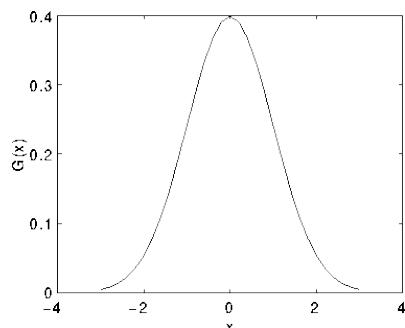
1. La imagen se suavizada usando un filtro Gausiano con una desviación estándar, esto para reducir el ruido. El gradiente local, y la dirección del borde, son computadas en cada punto. El detector Sobel, puede ser usado para computar  $G_x$  y  $G_y$ .
2. Se define un punto de borde, como un punto cuyo peso es localmente máximo en la dirección del gradiente.
3. Los puntos de borde determinados en (2) originan crestas en la dirección de crecimiento del gradiente de la imagen. El algoritmo luego rastrea a lo largo de la cima de estas crestas, y lleva a cero los píxeles que no están en realidad sobre la cima de la cresta, originando una línea delgada en la salida, un proceso conocido como supresión no máxima. Los píxeles de crestas son luego comparados usando dos umbrales,  $T_1$  y  $T_2$  con  $T_1 < T_2$ . Los píxeles de crestas con valores mayores que  $T_2$  se dice que son “probables candidatos” para ser píxeles de borde. Los píxeles en crestas con valores entre  $T_1$  y  $T_2$  se dice que son “candidatos poco probables” para ser píxeles de borde.
4. Por último, el algoritmo realiza la unión de los píxeles, incorporando “candidatos débiles” que están 8-conectados a los píxeles “probables”.

## Detector de Canny

- Filtrado gaussiano:
  - Funciones 1D y 2D:

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

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



## Detector de Canny

- Filtrado gaussiano:
  - Máscara de convolución:

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

$\frac{1}{273}$

## Detector de Canny

- Algoritmo:

- Convolución:
  - Filtro gaussiano
  - Gradiente
- Vectorización
  - Módulo
  - Dirección
- Post-proceso
  - Supresión no-máximos.
  - Histéresis.

```

Algoritmo CANNY {
   $S \leftarrow G_\sigma * I$ 
   $[S_x \ S_y] \leftarrow \nabla(S)$ 
  Para cada pixel  $(i, j)$ 
     $|\nabla S(i, j)| \leftarrow \sqrt{S_x^2(i, j) + S_y^2(i, j)}$ 
     $S_\phi(i, j) \leftarrow \arctan \frac{S_x(i, j)}{S_y(i, j)}$ 
  SUPRESIÓN
  HISTERESIS
  Devolver  $\{|\nabla S|, S_\phi\}$ 
}

```

## Detector de Canny

- Supresión no-máximos:

- Tomar un conjunto discreto de direcciones.
- Para cada píxel:
  - Considerar la dirección más similar
  - Comprobar si es mejor que sus vecinos en dicha dirección

```

Algoritmo SUPRESIÓN DE NO-MÁXIMOS {
   $d_i \leftarrow [0^\circ, 45^\circ, 90^\circ, 135^\circ]$ 
  Para cada pixel  $(i, j)$ 
     $\hat{d}_k \leftarrow \arg \min_{d_k} |d_k - S_\phi(i, j)|$ 
    Si  $|\nabla S(i, j)|$  es menor que uno de sus dos vecinos en la dirección  $\hat{d}_k$ 
       $S_e(i, j) \leftarrow 0$ 
    Si no
       $S_e(i, j) \leftarrow |\nabla S(i, j)|$ 
    Devolver  $S_e$ 
}

```

## Detector de Canny

- Histéresis:

```

Algoritmo HYSTERESIS {
     $p \leftarrow$  Umbral a nivel alto
     $q \leftarrow$  Umbral a nivel bajo
     $L_{visitados} \leftarrow \emptyset$ 
     $L_{bordes} \leftarrow \emptyset$ 
    Para cada pixel  $(i, j)$ 
        Si  $(i, j) \notin L_{visitados}$  y  $S_e(i, j) > p$ 
             $(k, l) \leftarrow (i, j)$ 
             $B_{ij} \leftarrow \emptyset$ 
            Hacer
                 $B_{ij} \leftarrow B_{ij} \cup \{(k, l)\}$ 
                 $L_{visitados} \leftarrow L_{visitados} \cup \{(k, l)\}$ 
                 $(k, l) \leftarrow$  Pixel adyacente a  $(k, l)$  en dirección perpendicular a  $S_\phi(k, l)$ 
            Mientras  $S_e(k, l) > q$ 
                 $L_{bordes} \leftarrow L_{bordes} \cup B_{ij}$ 
            Devolver  $L_{bordes}$ 
}

```

- Fijar umbrales:

- Primario ( $p$ )
- Nivel bajo ( $q$ )

- Filtrado por  $p$ :

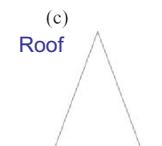
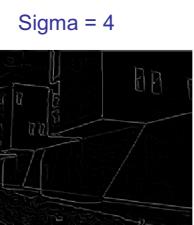
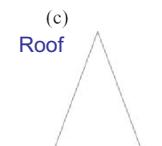
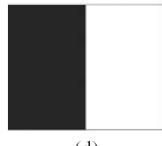
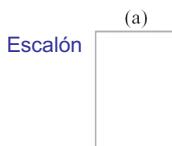
- Eliminar aristas que no alcancen el umbral alto.

- Añadir por  $q$ :

- Si  $> p$ , completar la arista con adyacentes mientras  $> q$

## Detector de Canny

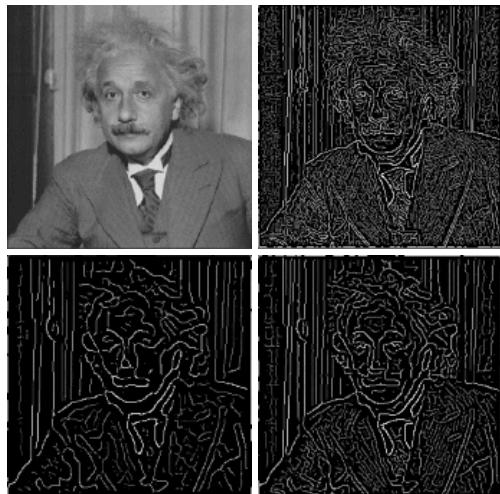
- Resultados:



## Detector de Canny

- Trade-off:

- Debido al filtro lineal (convolución)
- No es posible conjugar “detección” y “localización”
- Si aumentamos “sigma” reducimos ruido pero difuminamos los bordes y perdemos calidad en la localización.
- La anchura de la gaussiana controla el nivel de ruido pero también el de detalle.



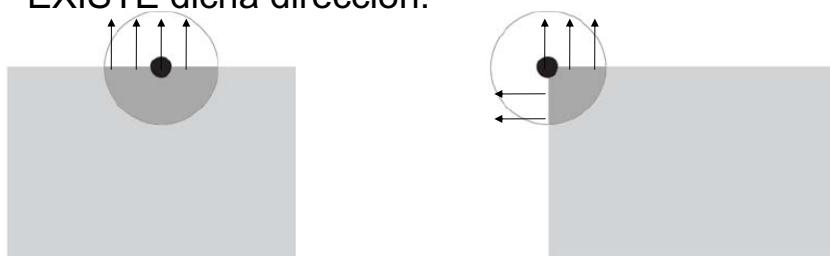
## Puntos esquina o “corners”

- Fundamentos:

- Definición y utilidad:
  - En un contorno: puntos de elevada curvatura (discontinuidad en la orientación del vector gradiente)
  - En uniones (“junctions”): punto de intersección de varias aristas, normalmente con un significado geométrico (ocultación, etc).
- Tipos de detectores:
  - Basados en “edges”: a partir de la información de curvatura.
  - Basados en “niveles de gris”: utilizan directamente el nivel de gris en una vecindad del píxel candidato para encontrar evidencia de que hay una esquina.
- Ejemplos:
  - Nitzberg/Harris

## Detector Nitzberg-Harris

- Principios de N-H:
  - Gradiente + filtrado lineal (gaussiano):
    - Construir una matriz 2x2 que codifica las estadísticas locales de gradiente y extraer sus autovalores y autovectores.
  - Se trata de determinar si existe o no alguna dirección de gradiente dominante. En un corner NO EXISTE dicha dirección.



Operaciones básicas

33

33

## Detector Nitzberg-Harris

- Matriz de momentos:
  - Construcción:
    - Calcular gradientes horizontal y vertical para cada uno de los píxeles de la vecindad
    - Aplicar promediado y registrar la matriz para cada punto.

$$I_x(x, y) = \frac{I(x+1, y) - I(x-1, y)}{2}$$

$$I_y(x, y) = \frac{I(x, y+1) - I(x, y-1)}{2}$$

$$\nabla I(x, y) = \begin{bmatrix} I_x(x, y) \\ I_y(x, y) \end{bmatrix}$$

$$A(\mathbf{x}) = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2(\mathbf{x}) & I_x I_y(\mathbf{x}) \\ I_x I_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{bmatrix} \quad w(x, y) = g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

## Detector Nitzberg-Harris

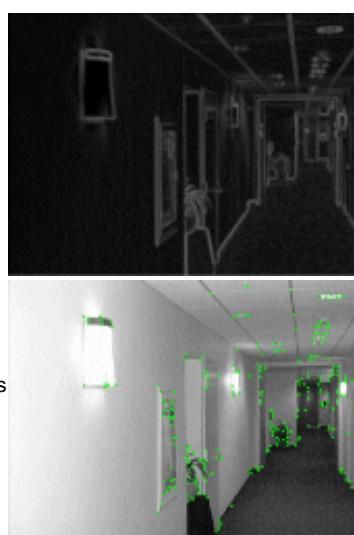
Matriz de momentos:

- Donde la matriz  $A(x, y)$  captura la estructura de la intensidad de la vecindad local.
- Sean  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz  $A(x, y)$ . Los valores propios forman una descripción rotacionalmente invariante.
- Autovalores y autovectores:  $(\lambda_1, e_1), (\lambda_2, e_2)$ 
  - $(\lambda_1, e_1)$ : El autovalor indica la fuerza de la dirección dominante y el autovector indica dicha dirección.
  - $(\lambda_2, e_2)$ : El autovalor indica la fuerza de la dirección secundaria y el autovector indica una dirección perpendicular a la principal.
- Interpretación:
  - Zona uniforme: no hay gradiente y entonces los autovalores son ambos cero.
  - Edges: solo hay una dirección primaria y no hay secundaria por lo que el segundo autovalor será cero.
  - Corners: Ambos autovalores son significativos, el autovector segundo se utiliza como “*cornerness*” y el primero como “*edgeness*”.

## Detector Nitzberg-Harris



original



edgeness

cornerness

corners

## Referencias

- Detector de Harris  
[http://en.wikipedia.org/wiki/Harris\\_affine\\_region\\_detector](http://en.wikipedia.org/wiki/Harris_affine_region_detector)
- Comparativa de distintos detectores de características K.  
Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, A comparison of affine region detectors. In IJCV 65(1/2):43-72, 2005
- Histograma de una imagen <http://es.wikipedia.org/wiki/Histograma>
- Ecualización del histograma  
[http://es.wikipedia.org/wiki/Ecuaci%C3%B3n\\_del\\_histograma](http://es.wikipedia.org/wiki/Ecuaci%C3%B3n_del_histograma)
- Algoritmo de Canny: Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986

