



Detección de bordes

El detector de Canny



Universidad
de Granada



DECSAI
Universidad de Granada

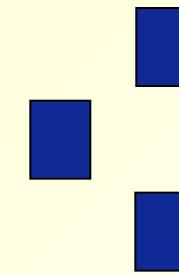




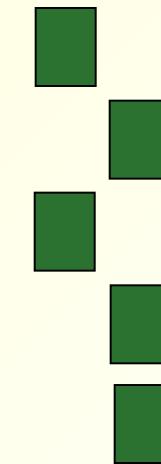
- Robusta al ruido
- Buena localización
- Respuesta única



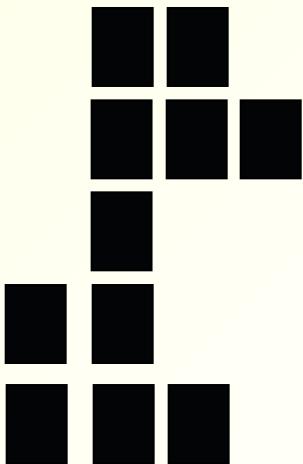
Frontera
verdadera



Pobre robustez
al ruido



Pobre
localización

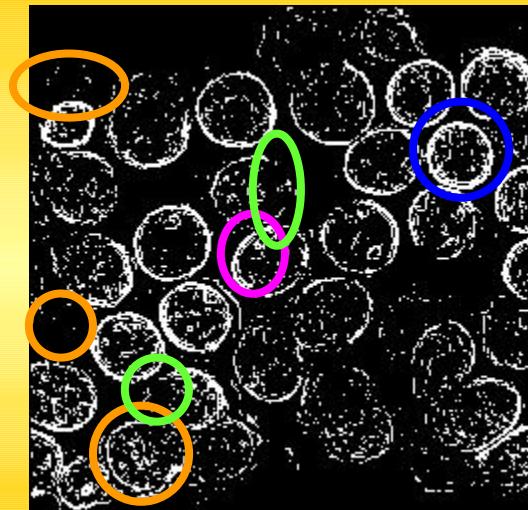
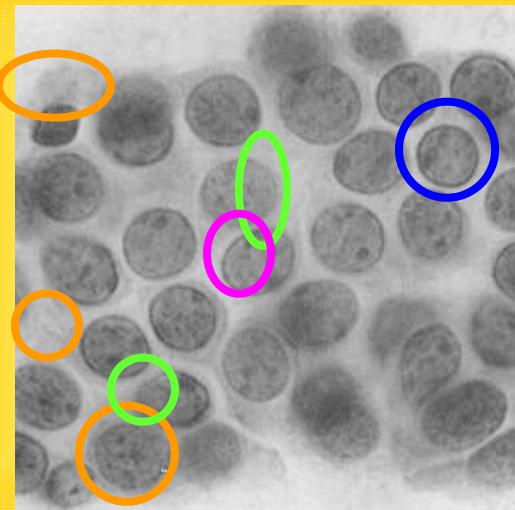


Demasiadas
respuestas



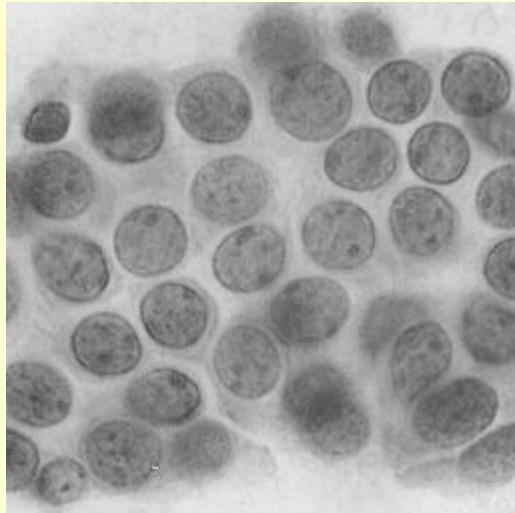
Los métodos basados en gradiente y laplaciana ...

- ... son muy sensibles al ruido.
- ... omiten algunos trozos de frontera quedando estas cortadas.
- ... obtienen fronteras de grosor superior a 1 pixel.
- ... a veces detectan fronteras dobles.

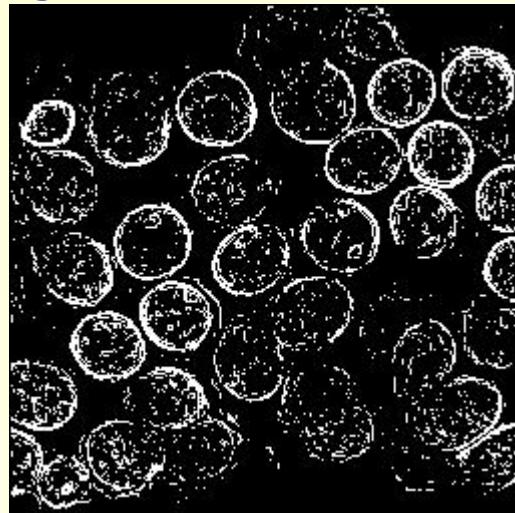




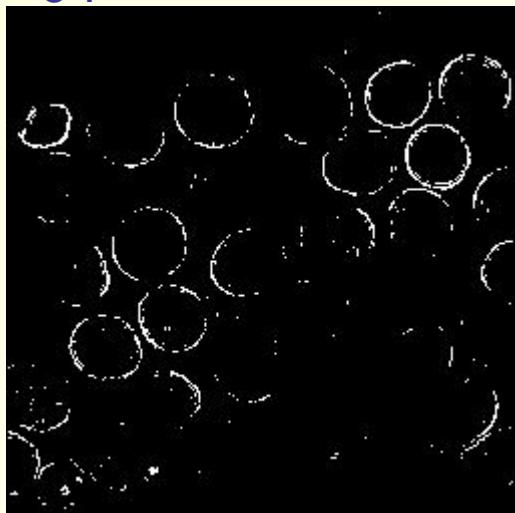
Original



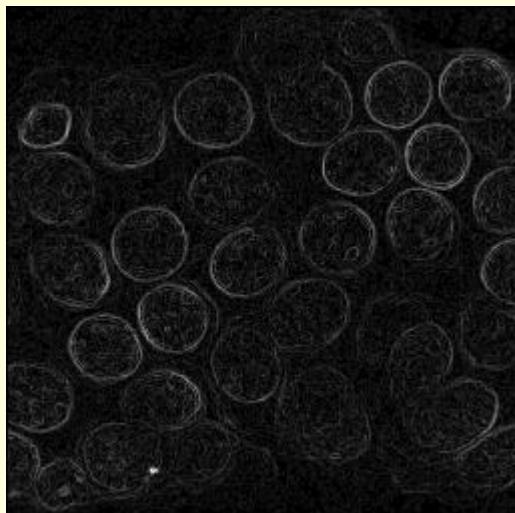
32



64



Umbralizaciones



Roberts

96



128

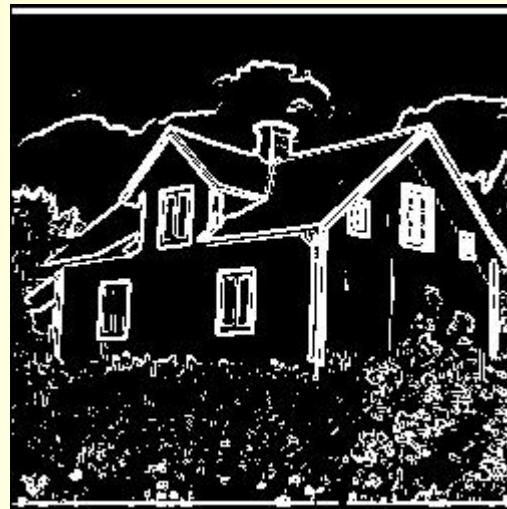




Original



32



64

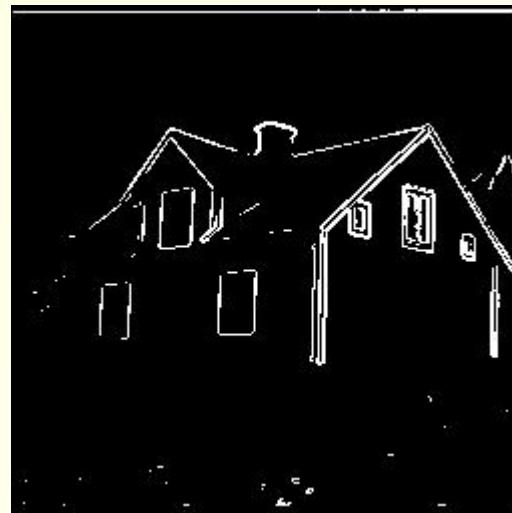


Umbralizaciones



Sobel

96



128





Criterios de optimalidad de Canny



El filtro propuesto por Canny ha de ser óptimo en base a estos 3 criterios:

→ **Buena detección**

Minimizar la probabilidad de falsos positivos (fronteras falsas).

→ **Buena localización**

Las fronteras detectadas han de estar tan cerca como sea posible de las fronteras reales.

→ **Respuesta única**

El detector devolverá un único punto por cada punto de frontera real.



1.- Detección de fronteras

Esta detección inicial seguirá teniendo los mismos problemas que las detecciones basadas en el gradiente aunque minimizados.

2.- Localización de fronteras

En esta segunda etapa se van a solucionar esos problemas.

2.1.- Supresión de no máximos

Obtendremos fronteras de grosor 1 pixel.

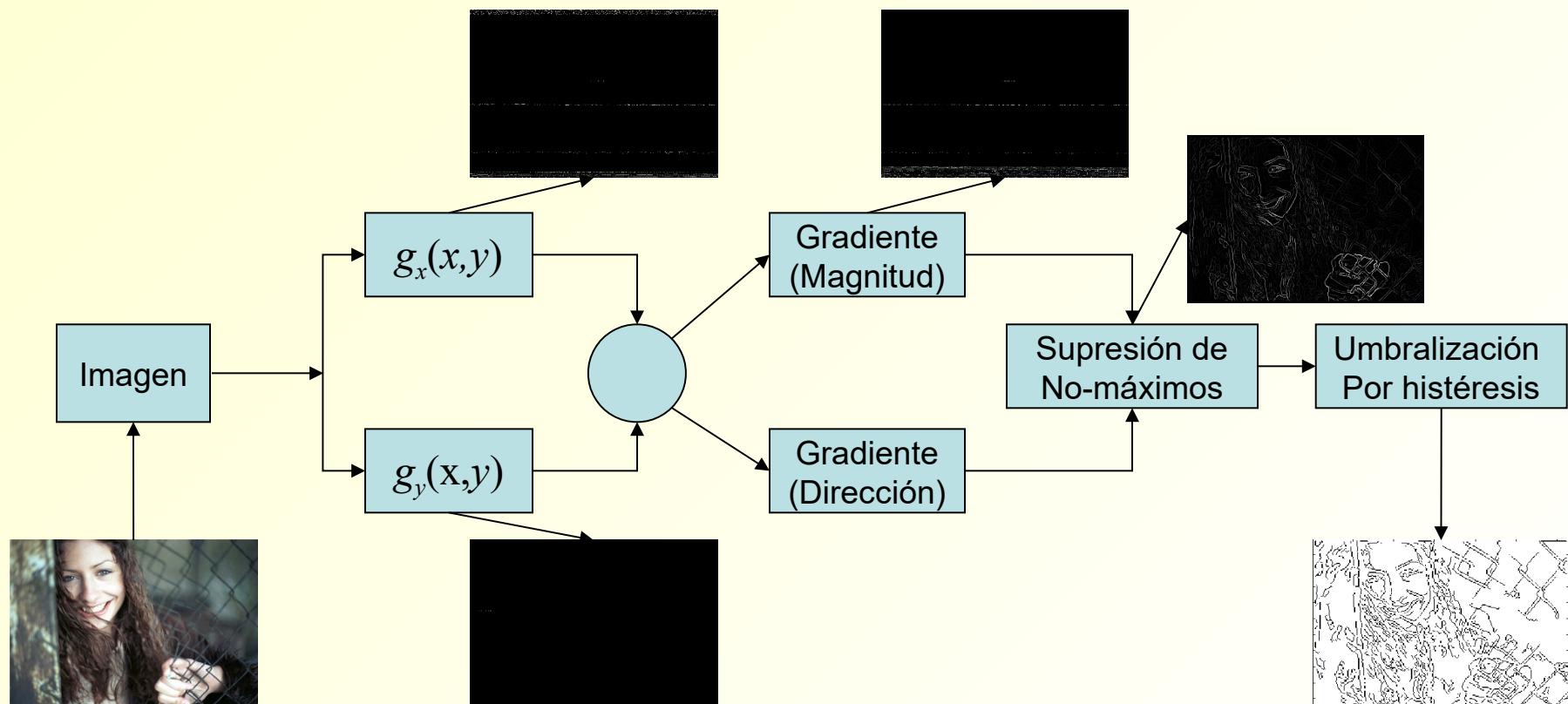
2.2.- Histéresis

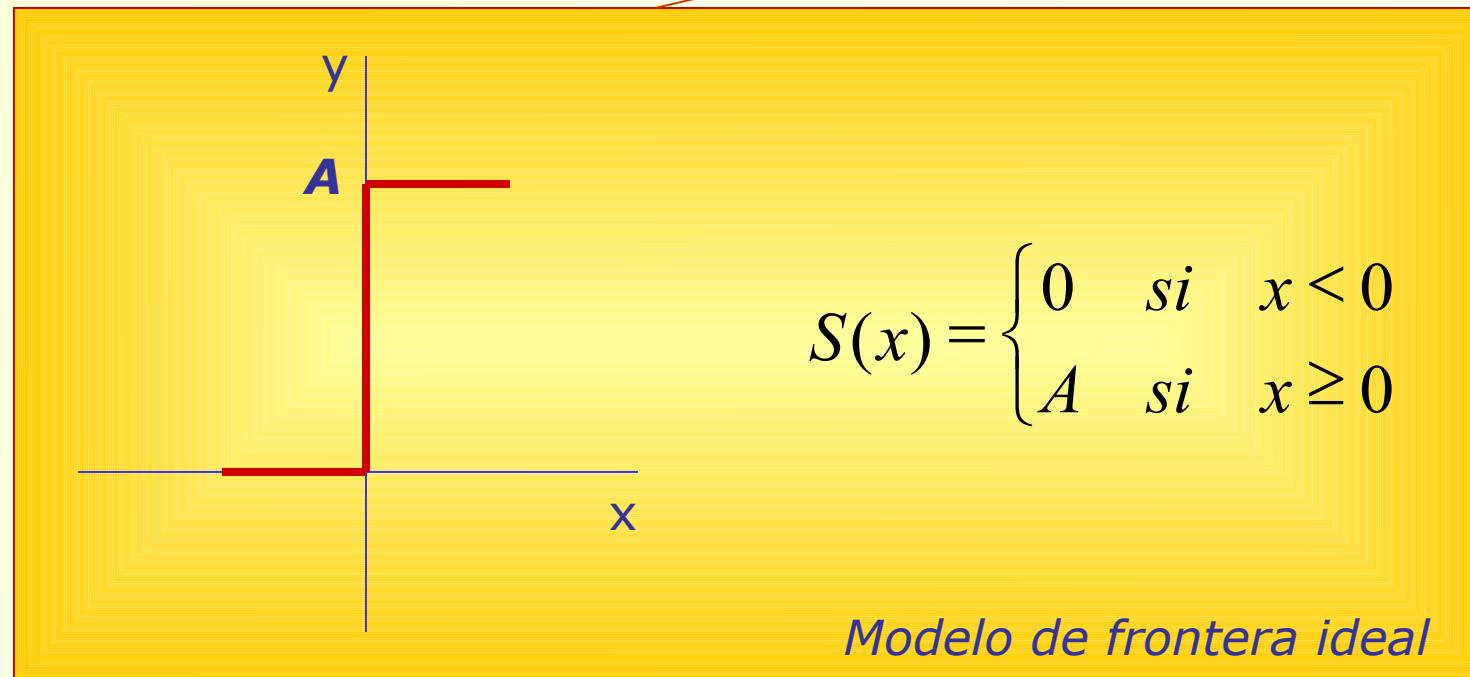
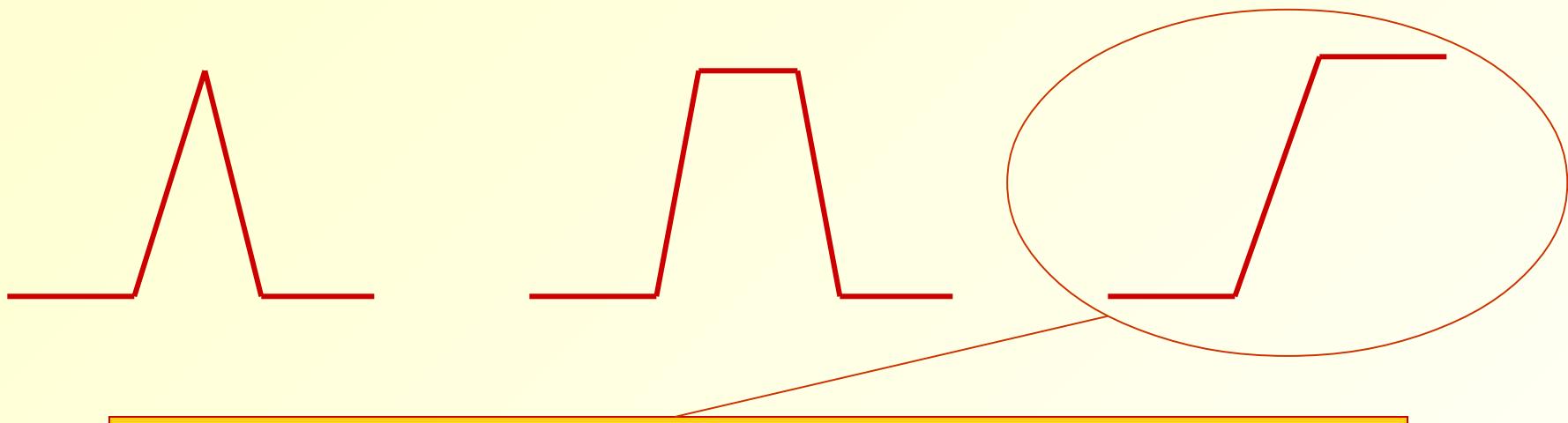
Eliminaremos gran parte de las fronteras debidas al ruido.

Evitaremos, en lo posible, tener fronteras cortadas.



Etapas del algoritmo. Esquema visual







Objetivo: encontrar un filtro óptimo para detectar las fronteras de tipo $S(x)$ y que optimice los criterios de optimalidad

$f(x)$ = filtro que buscamos.

W = ancho, en píxeles, del filtro $f(x)$

Respuesta del filtro $f(x)$ a la frontera ideal $S(x)$ en el punto 0:

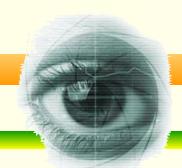
$$\int_{-W}^W S(-t) f(t) dt = A \int_{-W}^0 f(t) dt$$

Trabajaremos en 1D y después pasaremos a 2D.



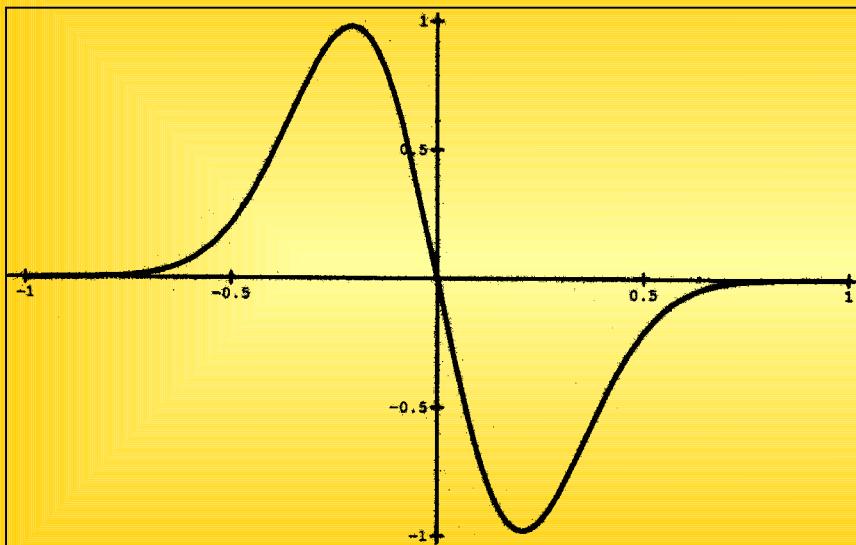
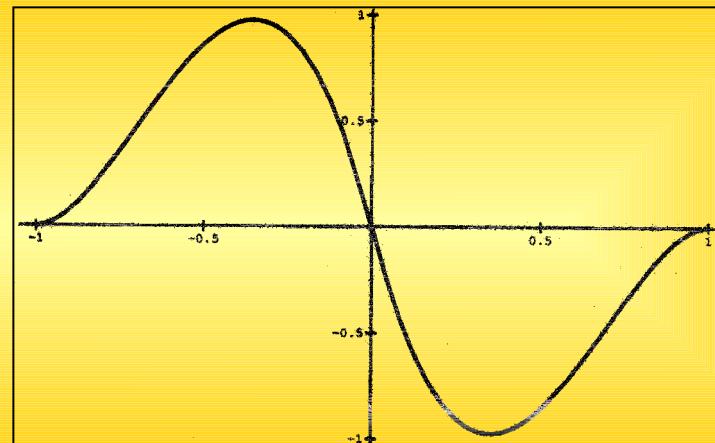
El detector de Canny

Detección. Filtro óptimo



El filtro óptimo propuesto por Canny tiene este perfil:

Es complejo de calcular.



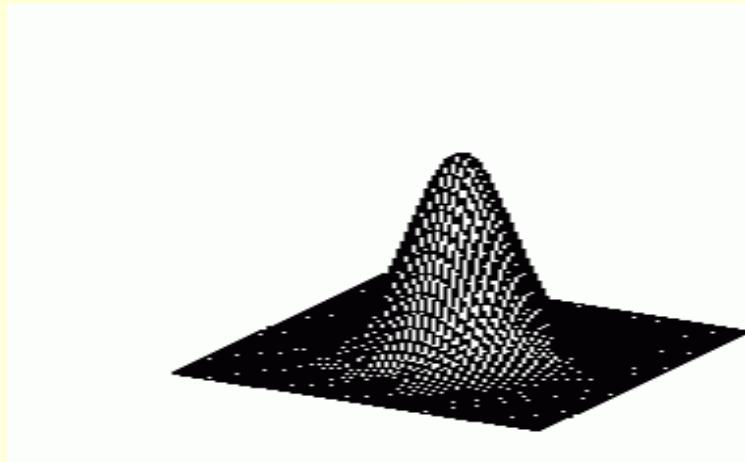
Se aproxima bien con una **derivada de gaussiana** (gradiente). Es fácil de calcular.

$$G_{1D}(x) = e^{-\frac{x^2}{2\sigma^2}}$$

$$G'_{1D}(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} = f(x)$$

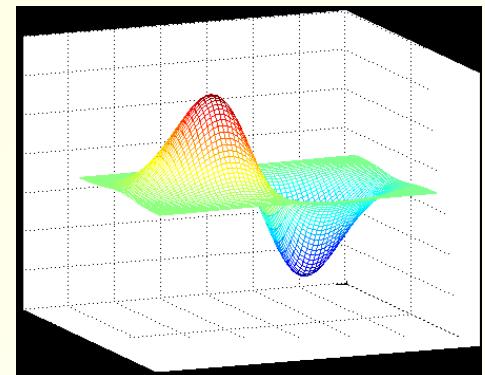


El filtro de Canny

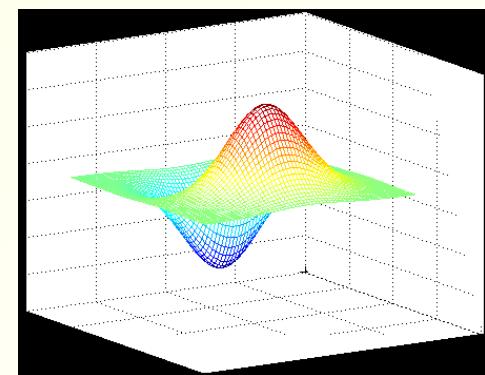


$$g(x, y)$$

$$g_x(x, y)$$



$$g_y(x, y)$$





$$G_{1D}(x) = e^{-\frac{x^2}{2\sigma^2}} \quad G_{2D}(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} e^{-\frac{y^2}{2\sigma^2}} = G_{1D}(x)G_{1D}(y)$$

Cálculo de las componentes horizontal y vertical del gradiente de la imagen $I(x,y)$

$$G_x(x, y) = \frac{\partial(I(x, y) \otimes G_{2D}(x, y))}{\partial x} = G_{1D}'(x) \otimes (G_{1D}(y) \otimes I(x, y))$$

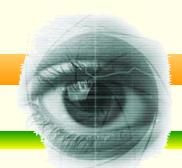
$$G_y(x, y) = \frac{\partial(I(x, y) \otimes G_{2D}(x, y))}{\partial y} = G_{1D}'(y) \otimes (G_{1D}(x) \otimes I(x, y))$$

Magnitud y dirección del gradiente:

$$M(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$$

La magnitud se normaliza al rango [0..1]

$$D(x, y) = \arctan \frac{G_y(x, y)}{G_x(x, y)}$$



Detector de Canny: es la primera derivada de una función Gaussiana junto con técnicas de mejora de las aristas detectadas.

Aproximación

1. Sea $I[i, j]$ la imagen de entrada, aplicamos un filtro gaussiano de desviación estandar σ a esta imagen para crear una imagen alisada $S[i, j]$:

$$S = G_\sigma \otimes I$$

2.-Obtenemos el gradiente de la imagen suavizada S . Para ello usamos filtros rápidos

$$\begin{matrix} -0.5 & 0.5 \\ -0.5 & 0.5 \end{matrix}$$

$$\begin{matrix} 0.5 & 0.5 \\ -0.5 & -0.5 \end{matrix}$$

Obteniendo el gradiente en la dirección x y el gradiente en la dirección y:

$$P[x, y] = G_x[i, j] = (S[i, j+1] - S[i, j] + S[i+1, j+1] - S[i+1, j])/2$$

$$Q[x, y] = G_y[i, j] = (S[i, j] - S[i+1, j] + S[i, j+1] - S[i+1, j+1])/2$$

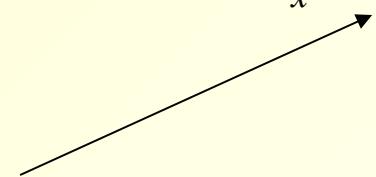


El filtro de Canny

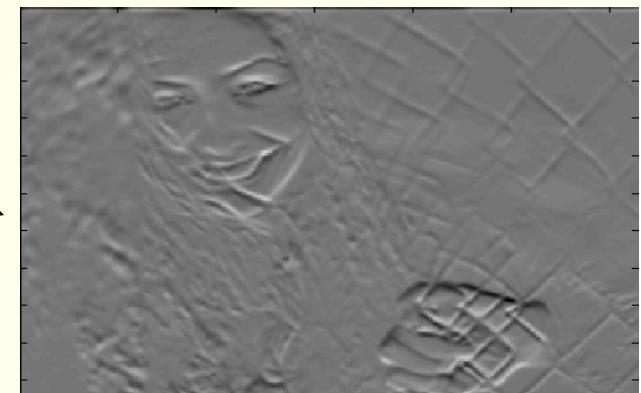
I



G_x



G_y





Aproximación (continuación)

4.-A partir de los gradientes $P(i,j)$ y $Q(i,j)$ obtenemos la imagen magnitud y orientación

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad \theta[i, j] = \arctan(Q[i, j], P[i, j])$$

Con la magnitud M obtenemos la fuerza de la arista, como de significativa es.

Con la orientación tenemos la dirección de la arista.



Imagen



Magnitud del gradiente



Detección. Otro ejemplo de gradiente



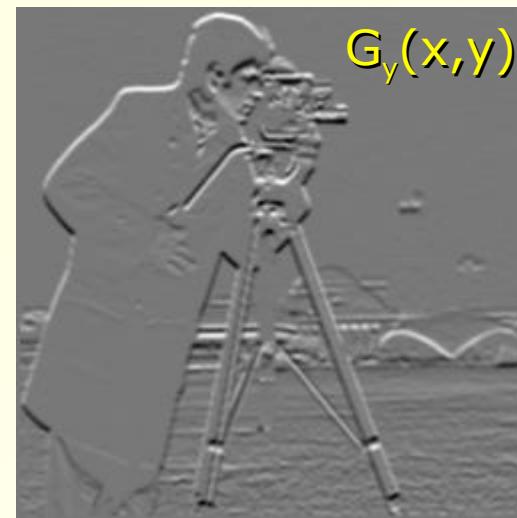
Original



Gradiente horizontal



Magnitud



Gradiente vertical



Dirección



Detección. Efecto del parámetro de escala

Detección vs. Localización



$\sigma=1$



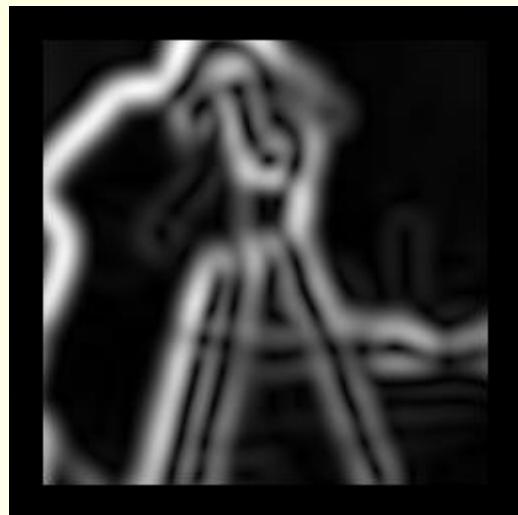
$\sigma=2$



$\sigma=3$



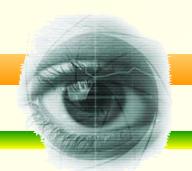
$\sigma=5$



$\sigma=7$



$\sigma=10$



Todos los puntos tienen un valor de gradiente

Sólo nos interesan los de gradiente elevado

Cerca de las fronteras reales habrá varios puntos con gradiente elevado

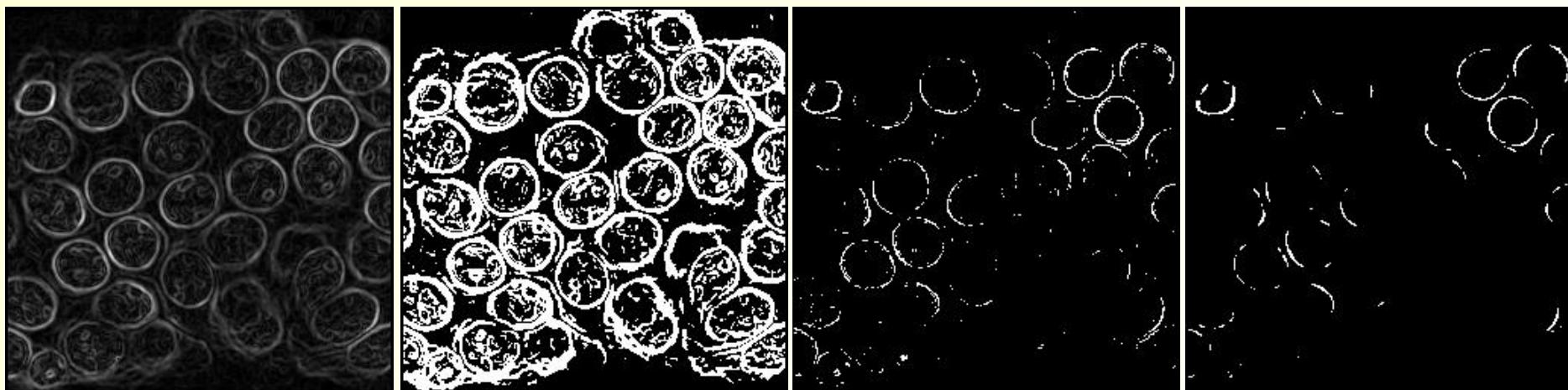
Grosor de frontera superior a 1 pixel

No bastaría un simple proceso de umbralización. Problemas:

Grosor de frontera superior a 1 pixel

Umbral bajo: demasiado ruido

Umbral alto: perdemos y cortamos fronteras





1.- Detección de fronteras

Esta detección inicial seguirá teniendo los mismos problemas que las detecciones basadas en el gradiente aunque minimizados.

2.- Localización de fronteras

En esta segunda etapa se van a solucionar esos problemas.

→ 2.1.- Supresión de no máximos

Obtendremos fronteras de grosor 1 pixel.

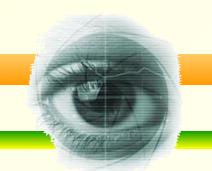
2.2.- Histéresis

Eliminaremos gran parte de las fronteras debidas al ruido.

Evitaremos, en lo posible, tener fronteras cortadas.



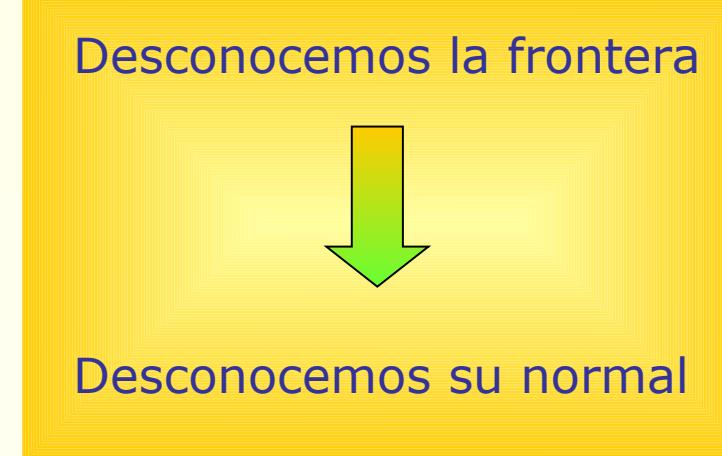
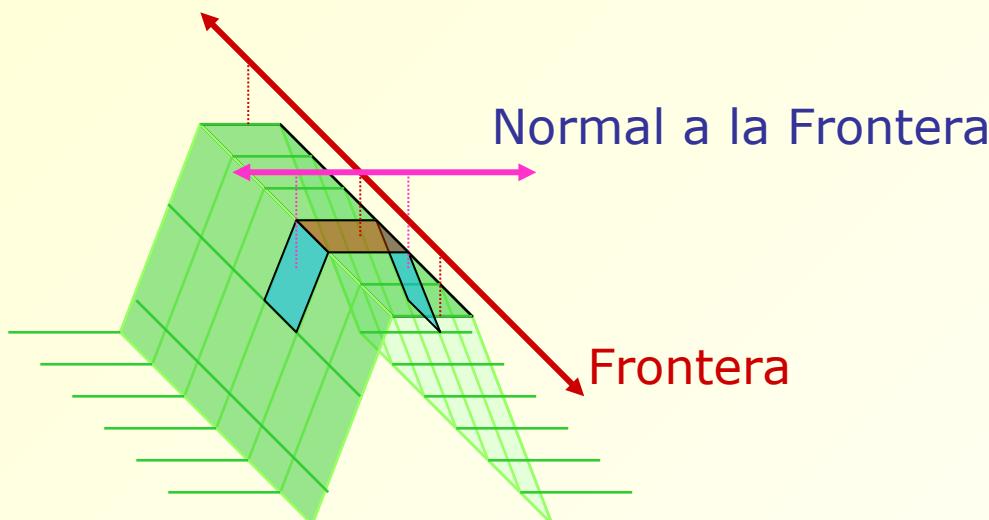
Localización. Supresión de no máximos

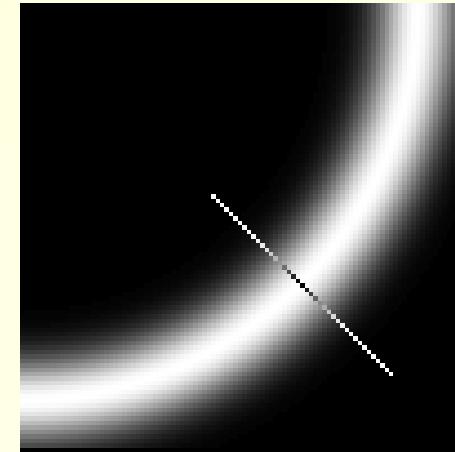
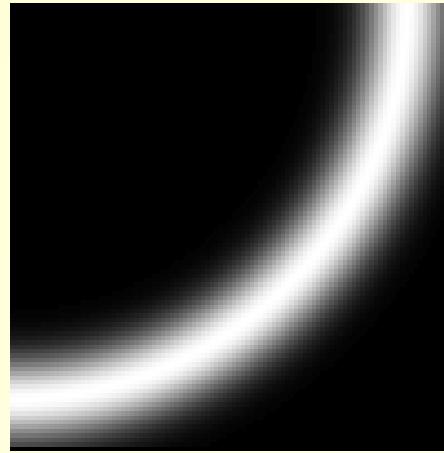


Prescindir de aquellos puntos que no sean máximos locales de la magnitud del gradiente.

$$E(x,y) = \text{supnomax}(x,y) = \begin{cases} 0 & \text{si } M(x,y) \text{ no es máximo local} \\ M(x,y) & \text{si } M(x,y) \text{ es máximo local} \end{cases}$$

¿Con qué criterio decidimos si un punto es máximo local?





Objetivo.- Detectar los puntos que son lo máximos a lo largo de la normal.

Procedimiento: Buscar los máximos a lo largo de la normal a la dirección de la frontera.

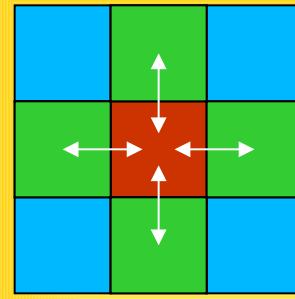


El detector de Canny

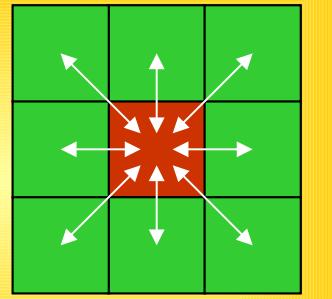
Localización. Supresión de no máximos



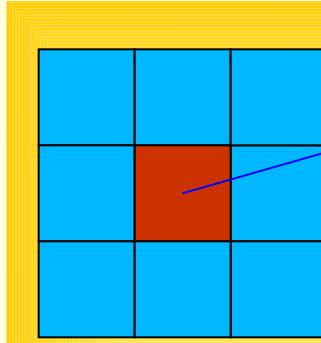
Aproximaciones para calcular máximos locales:



4-entorno

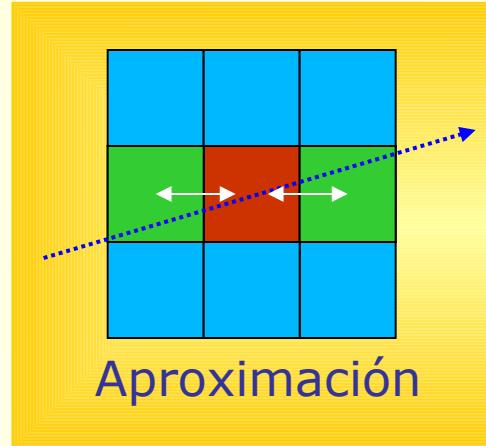
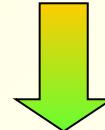


8-entorno

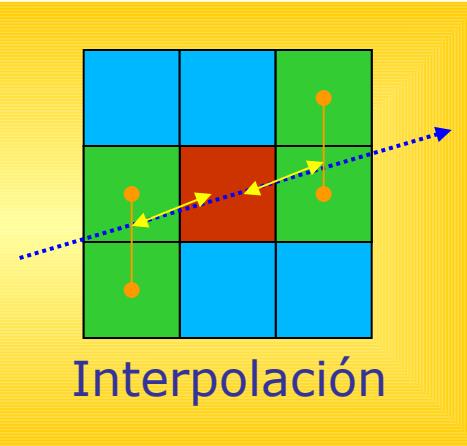


$D(x,y)$ = dirección
del gradiente.

Problema:
imágenes discretas



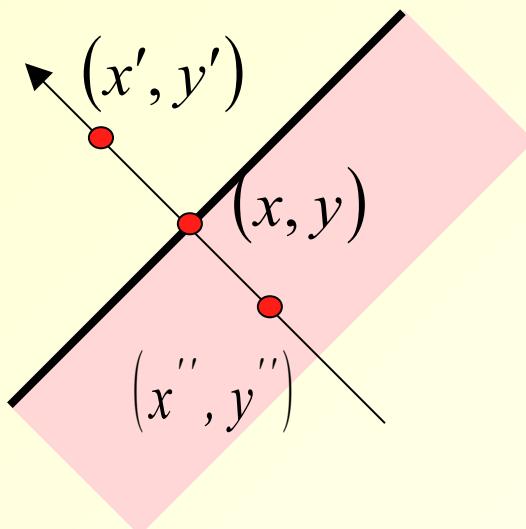
Aproximación



Interpolación



Aquellos puntos cuya $|\nabla S|$ que no son máximos locales no son considerados puntos de frontera.



$$M(x, y) = \begin{cases} |\nabla S|(x, y) & \text{if } |\Delta S|(x, y) > |\Delta S|(x', y') \\ & \& |\Delta S|(x, y) > |\Delta S|(x'', y'') \\ 0 & \text{en otro caso} \end{cases}$$

x' y x'' son los vecinos de x a lo largo de la dirección normal a la frontera

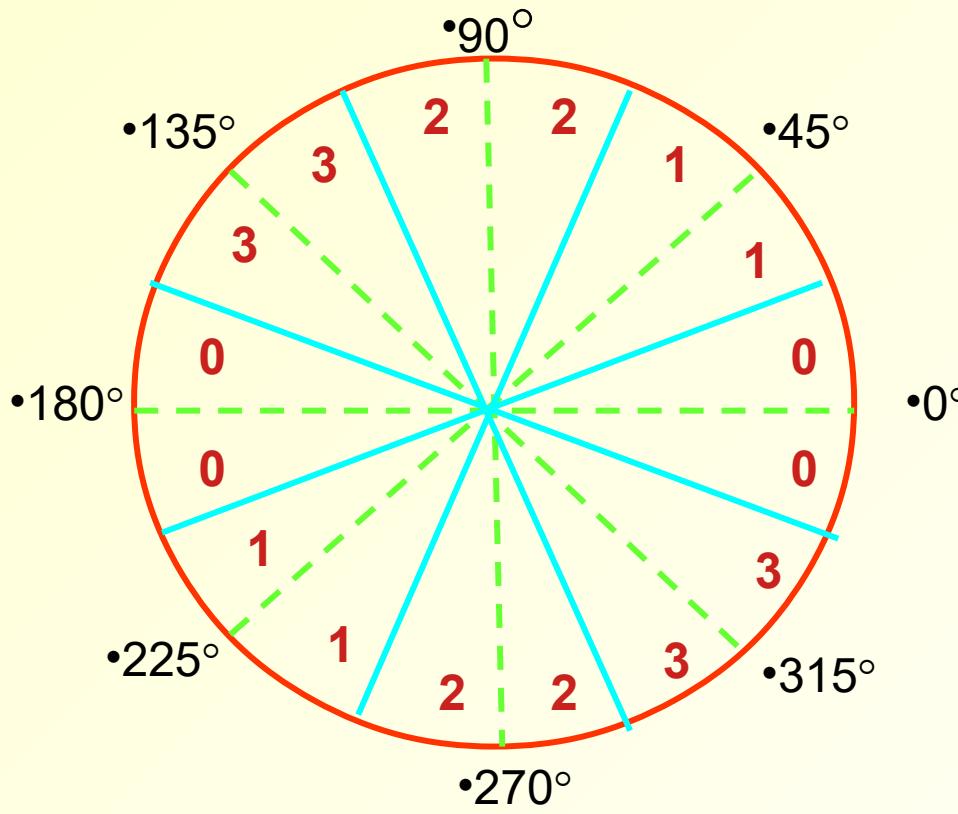


Supresión de no máximos

Algoritmo Supresión de no máximos

1.- Asignar un sector $\xi(i,j)$ a cada pixel en función de la orientación obtenida en $\theta(i,j)$

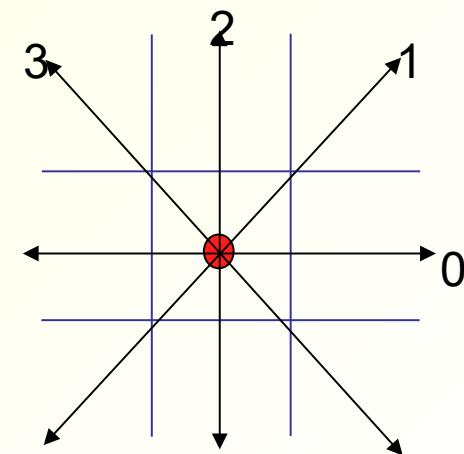
$$\theta[i, j] = \arctan(Q[i, j], P[i, j])$$



$$\tan \theta = \frac{G_y}{G_x}$$

Cuantizaciones:

- 0: $-0.4142 < \tan \theta \leq 0.4142$
- 1: $0.4142 < \tan \theta < 2.4142$
- 2: $|\tan \theta| \geq 2.4142$
- 3: $-2.4142 < \tan \theta \leq -0.4142$





Supresión de no máximos

Algoritmo Supresión de no máximos

2.- Para cada elemento (i,j) de la magnitud M comparamos $M(i,j)$ con dos de sus vecinos de su 8 entorno

Los dos vecinos con los que se compará depende de $\zeta(i, j)$

3	2	1
0	(i,j)	0
1	2	3

P.e si (i,j) tiene $\zeta(i, j) = 3$ se compara con $(i-1,j-1)$ y $(i+1,j+1)$

Si el valor de $M(i, j)$ es menor que cualquiera del valor M en estas dos posiciones vecinas entonces $E(i, j) = 0$.
En otro caso hacer $E(i,j)=M(i,j)$

Resultado de esta fase



La matriz $E(i,j)$ que contiene valores diferentes de cero en aquellos puntos cuya magnitud era un máximo local en $M(i,j)$

Siguiente fase: Umbralización de E



Histerisis



Supresión de no máximos



$$|\Delta G| = \sqrt{G_x^2 + G_y^2}$$

M



E

Para visualización
 $E \geq Umbral = 25$



Localización. Supresión de no máximos

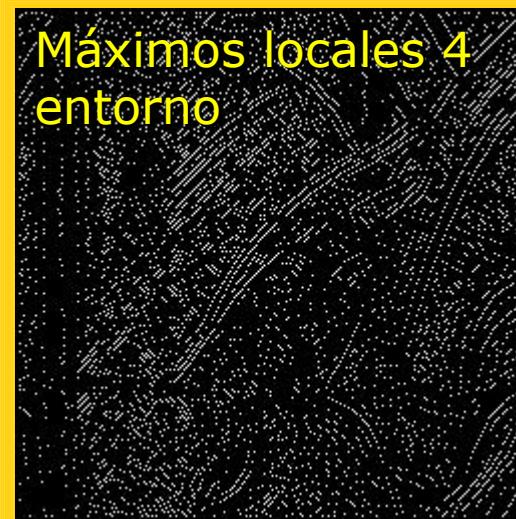
Original



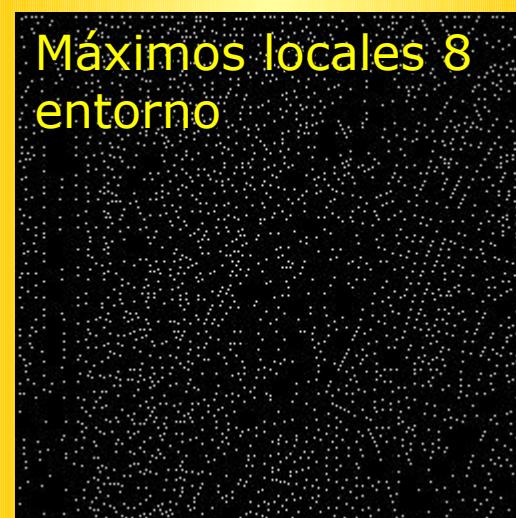
Magnitud Gradiente
 $M(x,y)$



Máximos locales 4
entorno



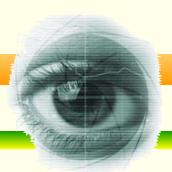
Máximos locales 8
entorno



Máximos locales en la
dirección del gradiente
(sin interpolación)

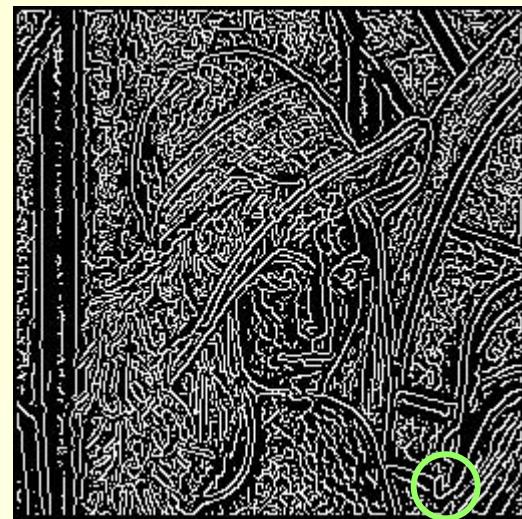
$\text{supnmax}(x,y)$





Localización. Supresión de no máximos

El ruido sigue estando
muy presente



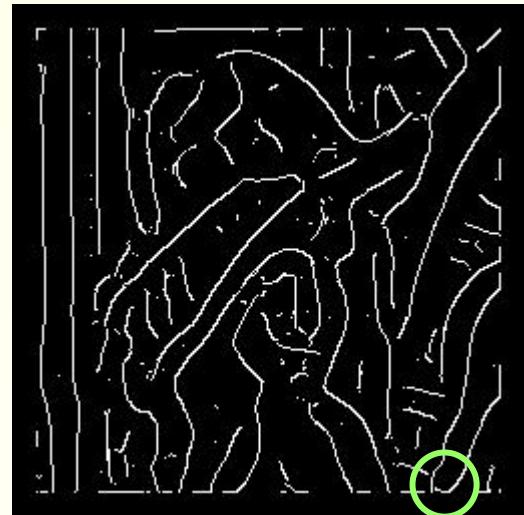
$\sigma=1$



$\sigma=2$



$\sigma=3$



$\sigma=5$

Detección
vs.
Localización



1.- Detección de fronteras

Esta detección inicial seguirá teniendo los mismos problemas que las detecciones basadas en el gradiente aunque minimizados.

2.- Localización de fronteras

En esta segunda etapa se van a solucionar esos problemas.

2.1.- Supresión de no máximos

Obtendremos fronteras de grosor 1 pixel.



2.2.- Histéresis

Eliminaremos gran parte de las fronteras debidas al ruido.

Evitaremos, en lo posible, tener fronteras cortadas.



Histéresis: Solución al problema de ruido y fronteras cortadas de la umbralización clásica. Consiste en:

Establecemos dos umbrales

Umbral superior $usup$

Umbral inferior $uinf$

Calculamos la imagen $his(x,y)$ como:

$his(x,y) = \text{FRONTERA}$

si $\text{supnomax}(x,y) > usup$

$his(x,y) = \text{NO-FRONTERA}$

si $\text{supnomax}(x,y) < uinf$

$his(x,y) = \text{CANDIDATO}$

en otro caso

En los puntos **CANDIDATOS**:

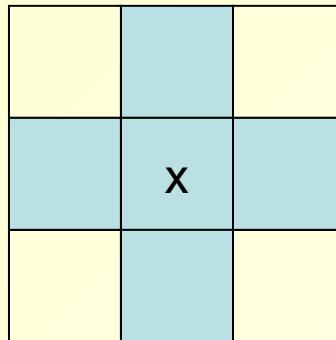
Un punto **CANDIDATO** será marcado como **FRONTERA** si está conectado con otro punto **FRONTERA**.

El proceso se aplica de forma iterativa o recursiva hasta que no se puedan marcar más puntos **FRONTERA**.

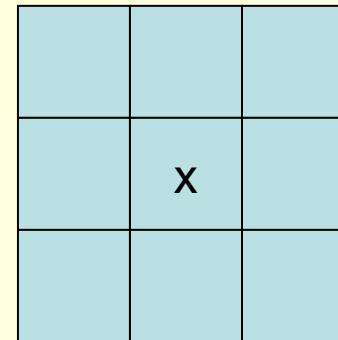
El resto de puntos **CANDIDATOS** serán marcados como **NO-FRONTERA**.



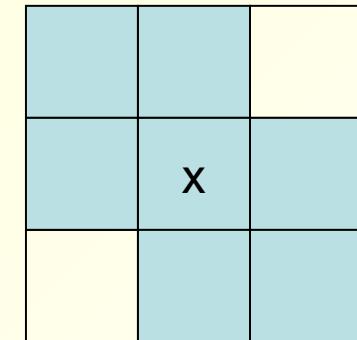
Criterios de Conectividad



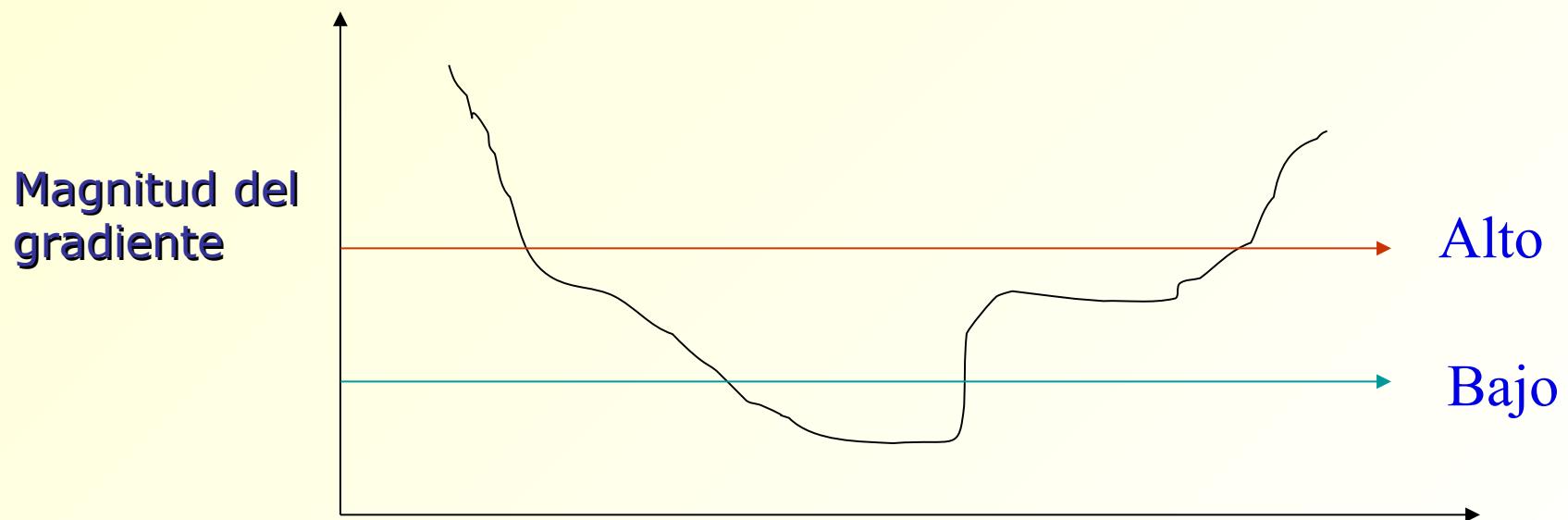
4-conexión



8-conexión



6-conexión





Ejemplo de umbralización clásica

El detector de Canny

Localización. Histéresis



Perfil 1D real

Perfil 1D con ruido

Puntos frontera reales

Puntos frontera detectados

Umbral

Umbral

Umbral

Perdemos puntos frontera
Fronteras cortadas
Marcamos el ruido como frontera



Ejemplo de histéresis

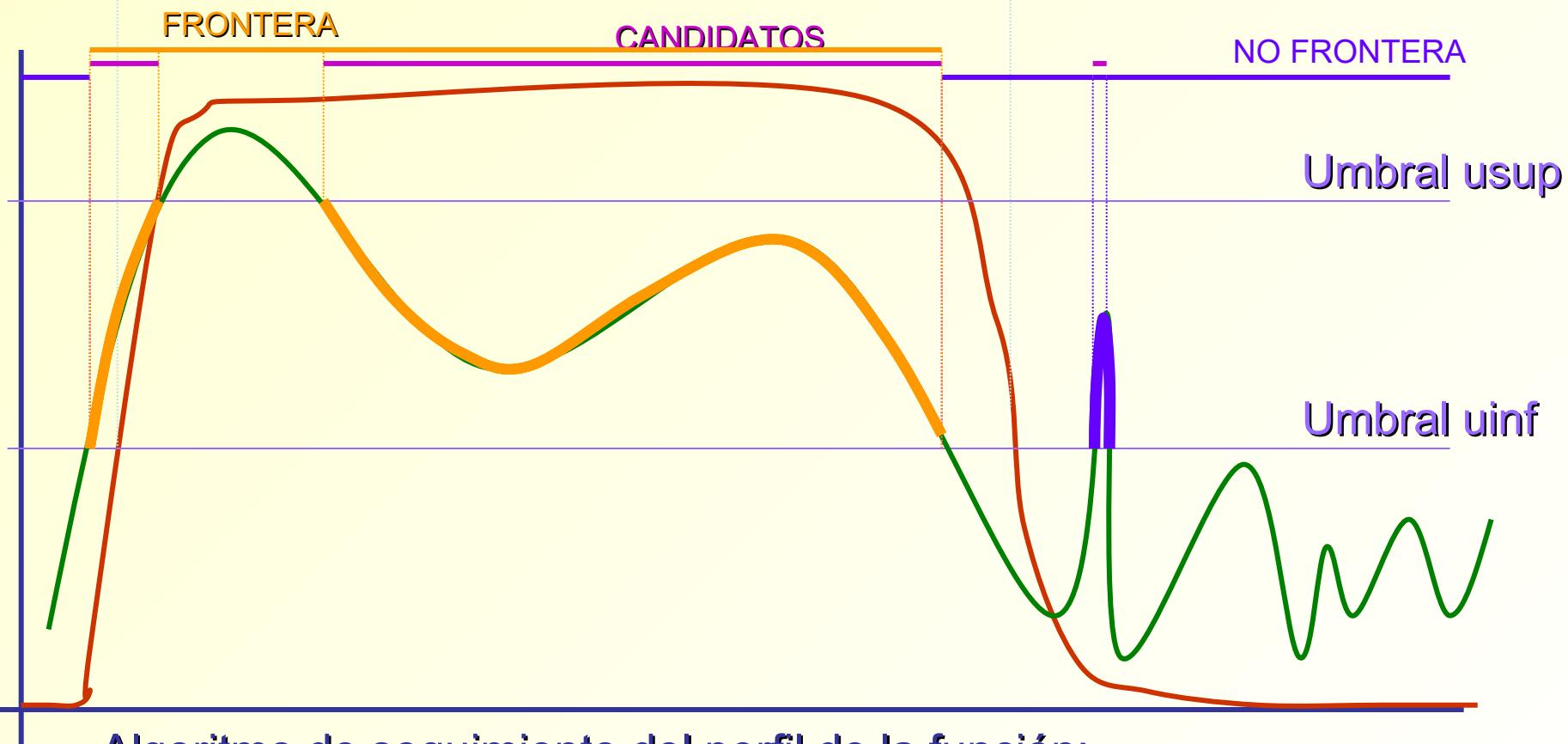
El detector de Canny

Localización. Histéresis



Puntos frontera reales

Perfil 1D real
Perfil 1D con ruido



Algoritmo de seguimiento del perfil de la función:
Desde cada punto FRONTERA
Hasta llegar a un punto NO-FRONTERA



Algoritmo:

- 1.- Recorrer la imagen por filas. Para cada pixel (i,j) :
 - 1.1 Si $E(i,j) < U_{inf}$ declararlo como punto no frontera (no considerarlo más)
 - 1.2 Si $E(i,j) > U_{sup}$ considerarlo como punto frontera.
 - 1.2.1 Para cada pixel (l,m) vecino a (i,j)
*Si $E(l,m) > U_{inf}$
Considerarlo como punto frontera
Aplicar el paso 1.2.1 para cada vecino de (l,m)
sino (l,m) declararlo como punto no frontera (no considerarlo más)

E



normal $E \geq 25$



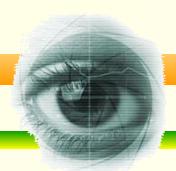
Histéresis Alto=35 Bajo =15



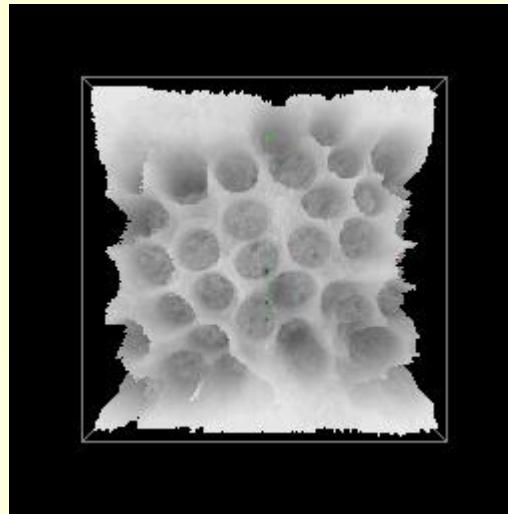
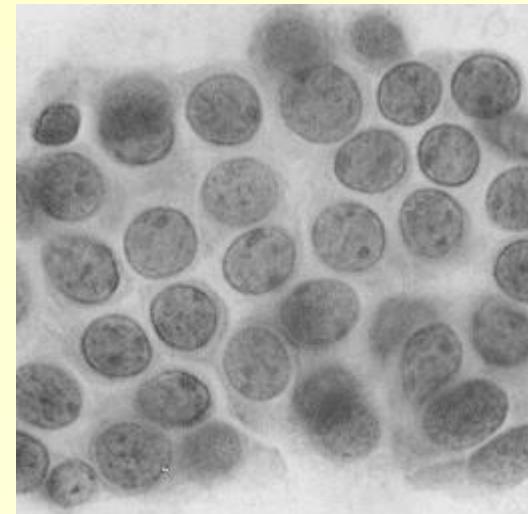


El detector de Canny

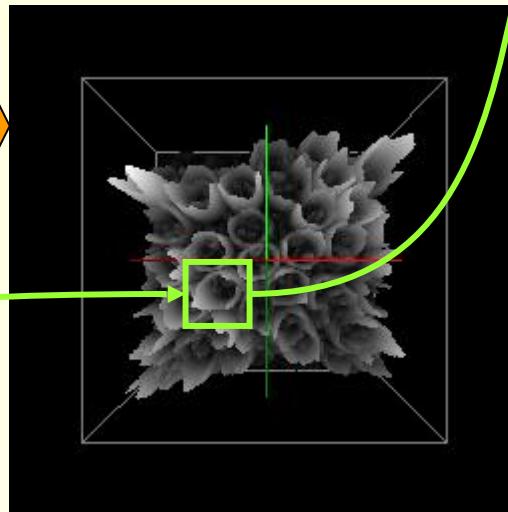
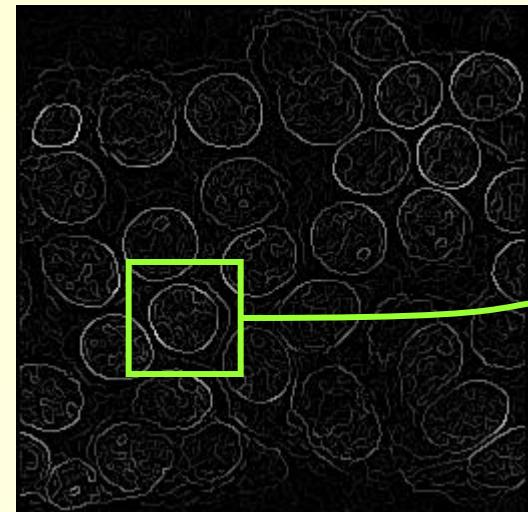
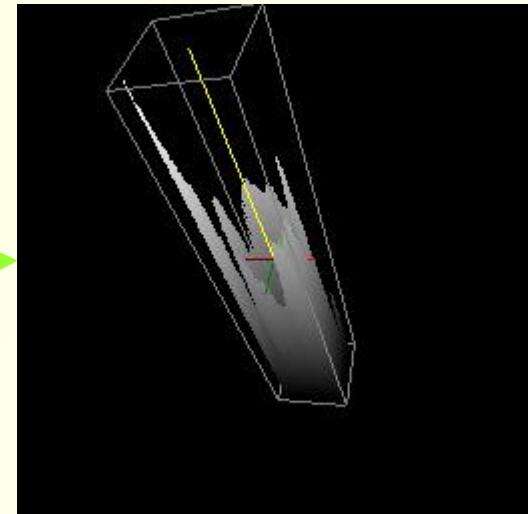
Localización. Histéresis



Histéresis en imágenes 2D

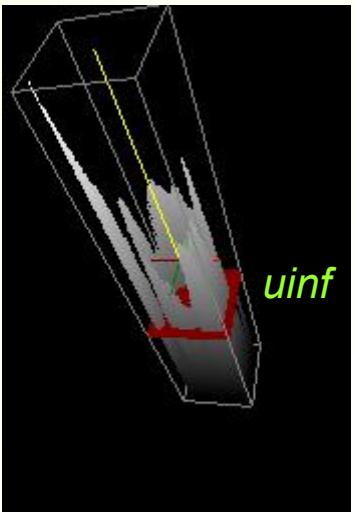
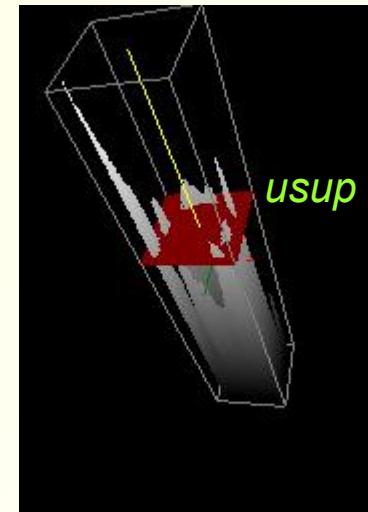


Vista 3D



Vista 3D

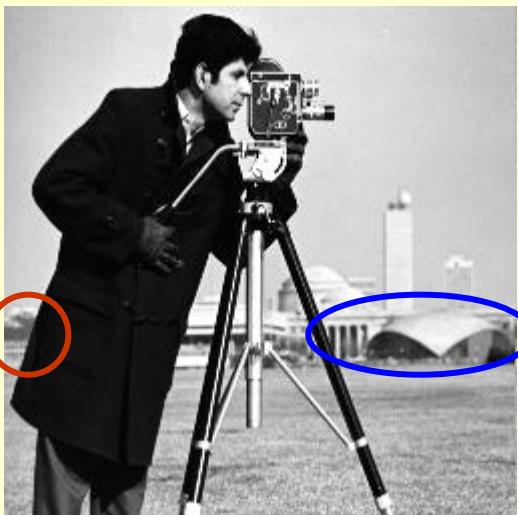
Supresión no máximos



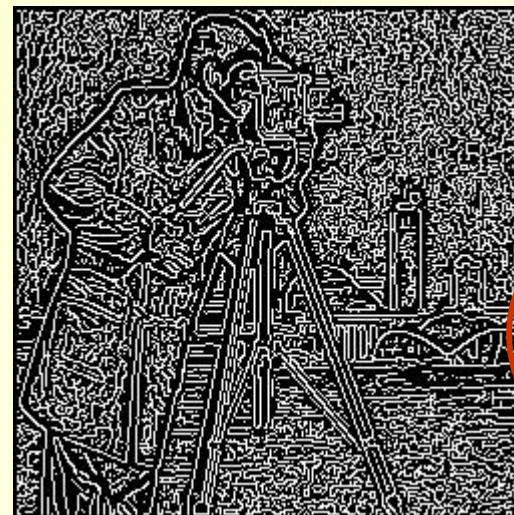


El detector de Canny

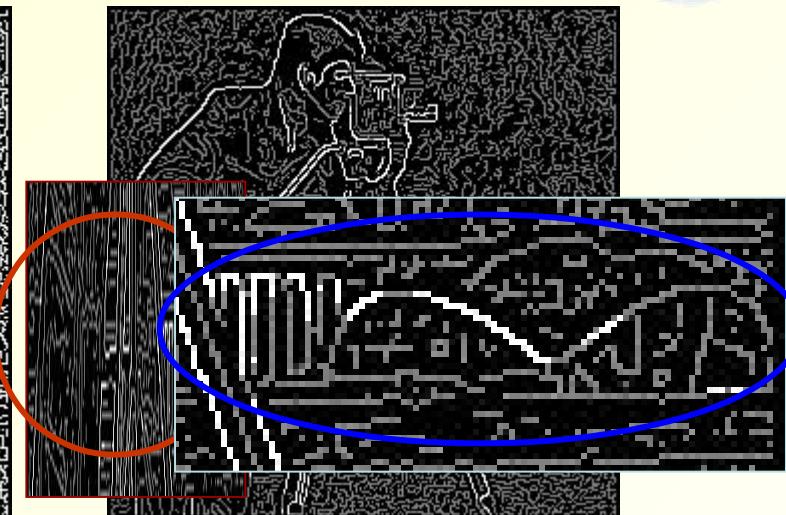
Localización. Histéresis



Original



Máximos locales



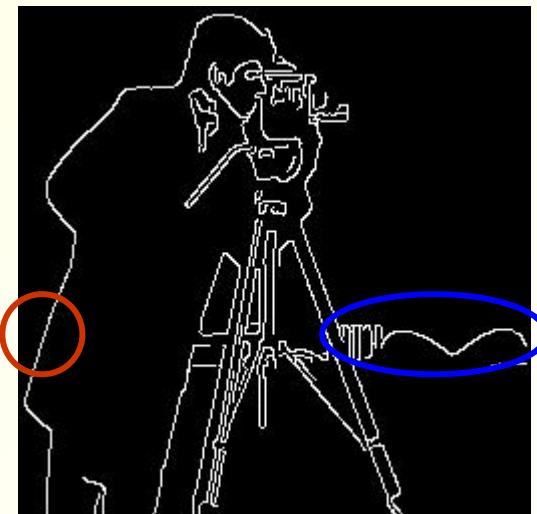
$u_{inf}=0.5$ $u_{sup}=0.9$



Magnitud gradiente



Supresión no máximos



Histéresis

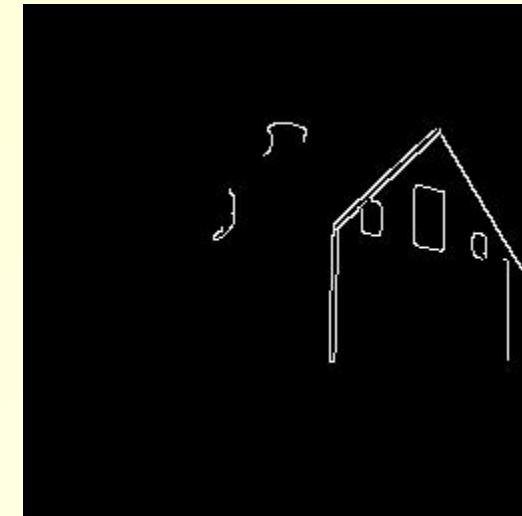


El detector de Canny

Localización. Histéresis



Original

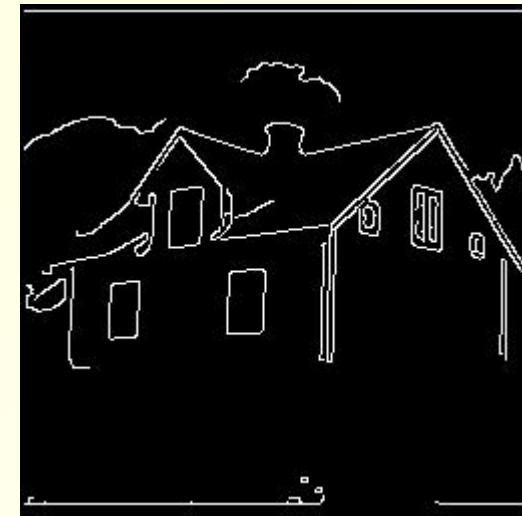


$u_{inf}=0.1$

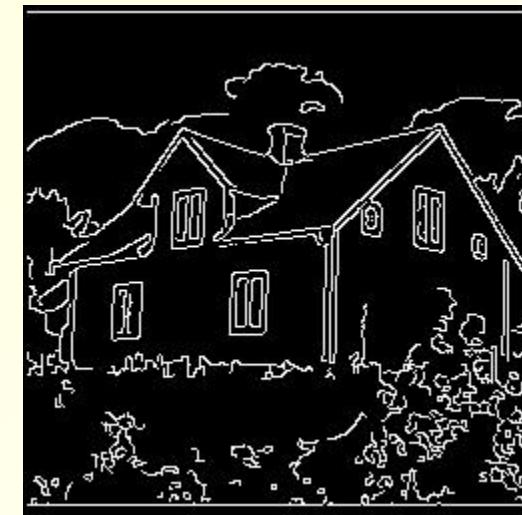
$u_{sup}=0.99$



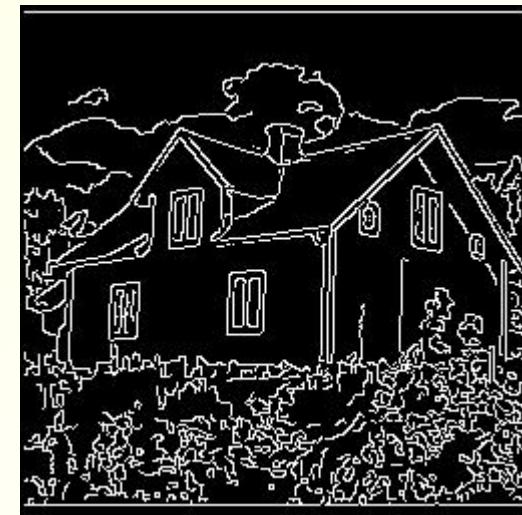
Magnitud gradiente



$u_{sup}=0.9$



$u_{sup}=0.8$



$u_{sup}=0.7$

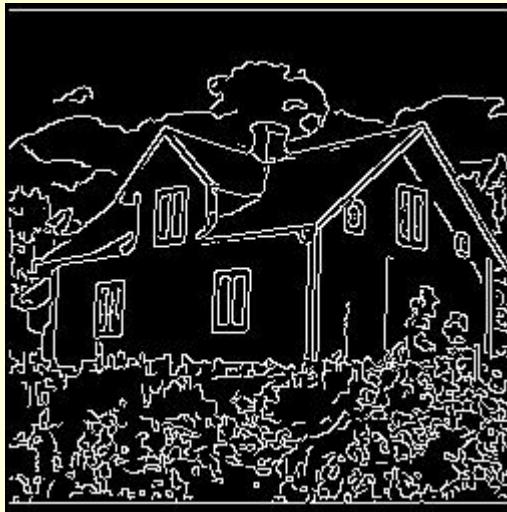


El detector de Canny

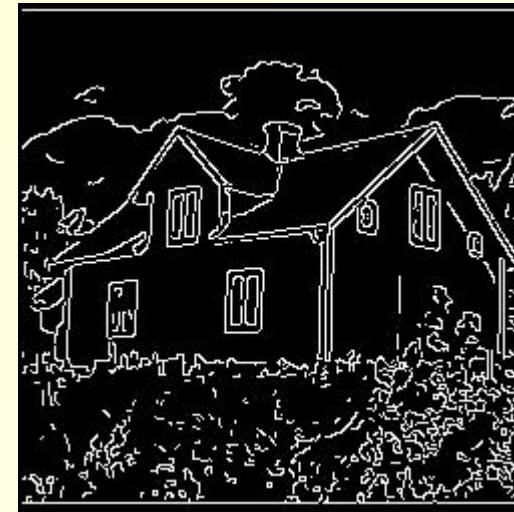
Localización. Histéresis



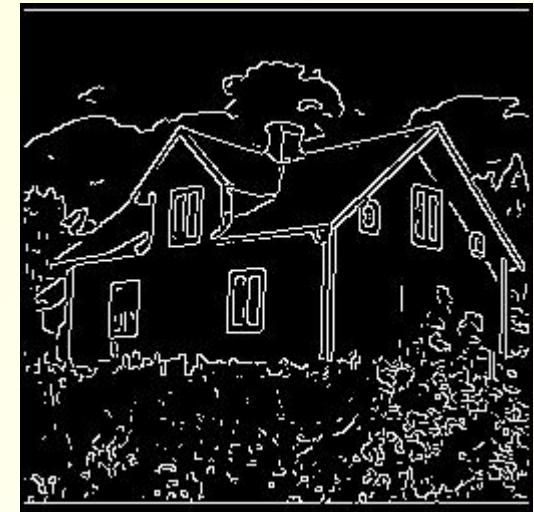
$usup=0.7$



$uinf=0.1$



$uinf=0.6$



$uinf=0.65$

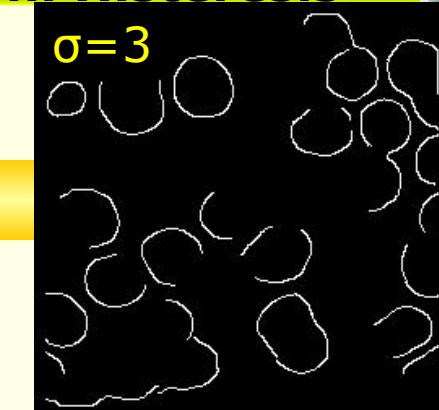
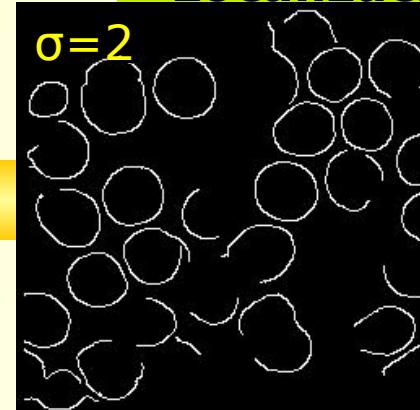
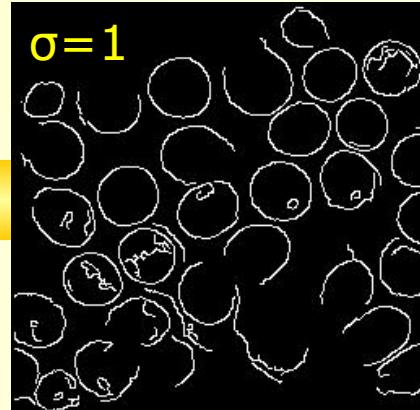
**usup suele ser alto
uinf suele ser bajo**



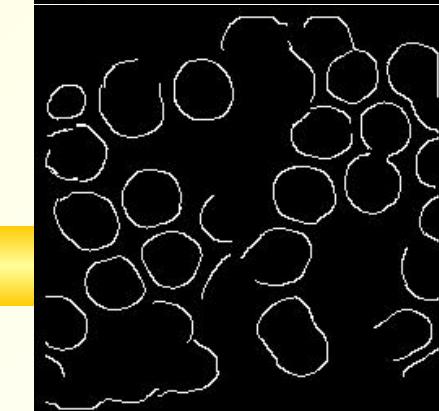
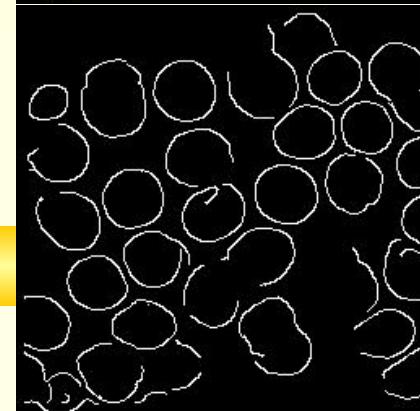
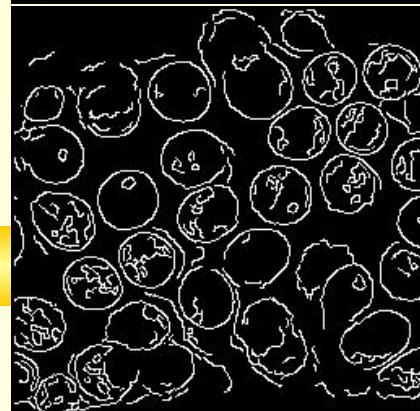
El detector de Canny

Localización. Histéresis

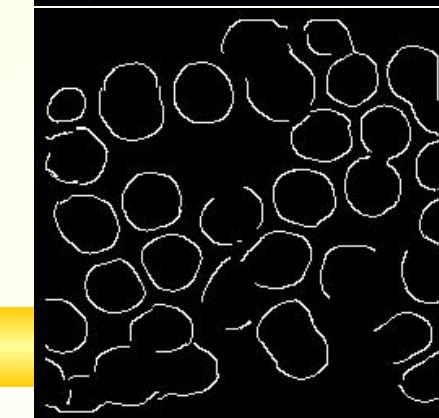
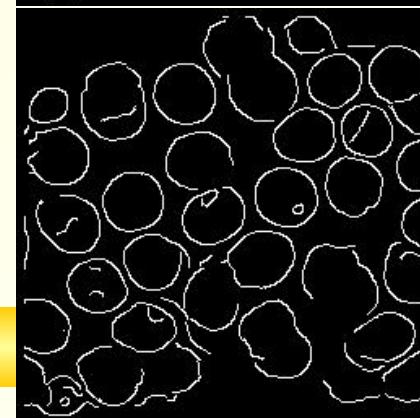
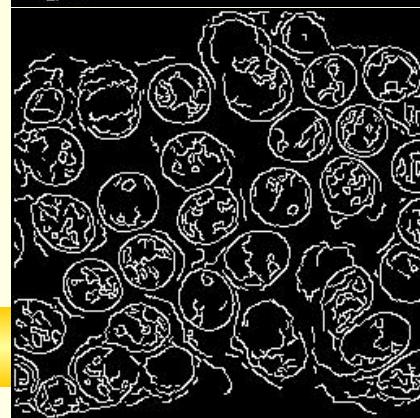
$usup=0.9$



$usup=0.8$



$usup=0.7$





Ejemplo numérico

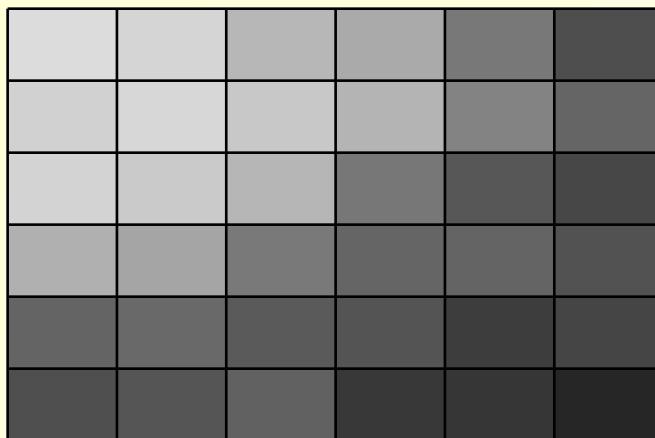


imagen de niv. de gris

220	213	183	170	120	78
209	215	200	180	131	101
211	202	182	119	87	71
176	165	121	101	100	82
100	105	90	84	61	69
79	85	97	56	54	38

Valores de niv. Gris (intensidades)
de la misma imagen



Ejemplo numérico



Aplicar los filtros de gradiente horizontal y vertical nos conduce a los siguientes resultados:

-0.5	-22.5	-16.5	-49.5	-36	0
-1.5	-17.5	-41.5	-40.5	-23	0
-10	-32	-41.5	-16.5	-17	0
-3	-29.5	-13	-12	-5	0
5.5	-1.5	-23.5	-12.5	-4	0
0	0	0	0	0	0

- $P[i, j]$

• P:

• -0.5	• 0.5
• -0.5	• 0.5

- Valores de niv. Gris

220	213	183	170	120	78
209	215	200	180	131	101
211	202	182	119	87	71
176	165	121	101	100	82
100	105	90	84	61	69
79	85	97	56	54	38



Ejemplo numérico



Aplicar los filtros de gradiente horizontal y vertical nos conduce a los siguientes resultados:

4.5	-9.5	-13.5	-10.5	-17	0
-5.5	15.5	39.5	52.5	37	0
36	49	39.5	2.5	-12	0
68	45.5	24	28	26	0
20.5	6.5	10.5	17.5	19	0
0	0	0	0	0	0

220	213	183	170	120	78
209	215	200	180	131	101
211	202	182	119	87	71
176	165	121	101	100	82
100	105	90	84	61	69
79	85	97	56	54	38

- $Q[i, j]$
- Valores de niv. Gris

•Q:

• 0.5	• 0.5
• -0.5	• -0.5



Ejemplo numérico

Aplicar los filtros de gradiente horizontal y vertical nos conduce a los siguientes resultados:

-0.5	-22.5	-16.5	-49.5	-36	0
-1.5	-17.5	-41.5	-40.5	-23	0
-10	-32	-41.5	-16.5	-17	0
-3	-29.5	-13	-12	-5	0
5.5	-1.5	-23.5	-12.5	-4	0
0	0	0	0	0	0

- $P[i, j]$

4.5	-9.5	-13.5	-10.5	-17	0
-5.5	15.5	39.5	52.5	37	0
36	49	39.5	2.5	-12	0
68	45.5	24	28	26	0
20.5	6.5	10.5	17.5	19	0
0	0	0	0	0	0

- $Q[i, j]$

$$M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad \theta[i, j] = \arctan(Q[i, j], P[i, j])$$



Ejemplo numérico



- Basándonos en $P[i, j]$ y $Q[i, j]$, podemos calcular magnitud y orientación del gradiente:

4.5	24.4	21.3	50.6	39.8	0
5.7	23.4	57.3	66.3	43.6	0
37.4	58.5	57.3	16.7	20.8	0
68.1	54.2	27.3	30.5	26.5	0
21.2	6.7	25.7	21.5	19.4	0
0	0	0	0	0	0

$M[i, j]$

•276°	•22°	•39°	•11°	•25°	•180°
•285°	•318°	•316°	•307°	•301°	•180°
•285°	•303°	•316°	•351°	•35°	•180°
•272°	•302°	•298°	•293°	•280°	•180°
•254°	•282°	•335°	•305°	•281°	•180°
•180°	•180°	•180°	•180°	•180°	•180°

$\theta[i, j]$



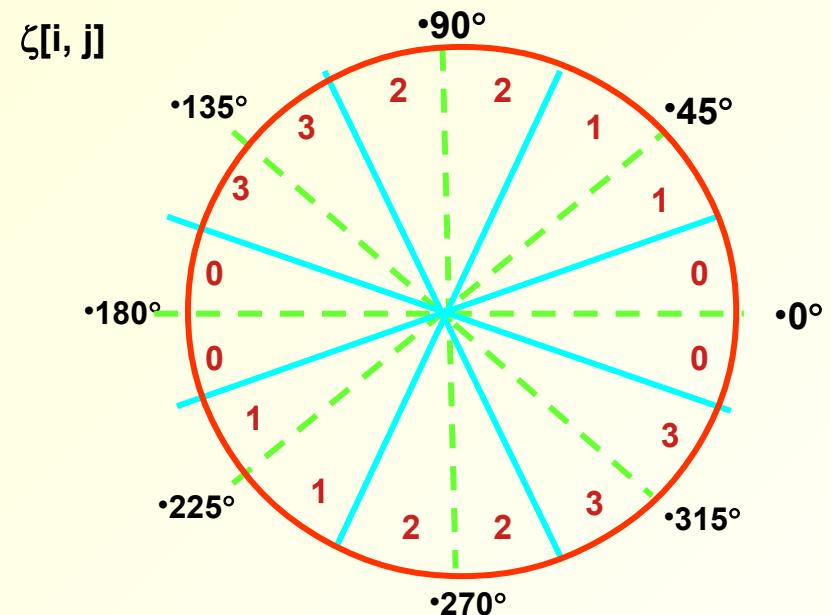
Ejemplo numérico



- Esto nos permite calcular el sector $\zeta[i, j]$ para cada angulo $\theta[i, j]$:

•2	•1	•1	•0	•1	•0
•2	•3	•3	•3	•3	•0
•2	•3	•3	•0	•1	•0
•2	•3	•3	•3	•2	•0
•2	•2	•3	•3	•2	•0
•0	•0	•0	•0	•0	•0

•276°	•22°	•39°	•11°	•25°	•180°
•285°	•318°	•316°	•307°	•301°	•180°
•285°	•303°	•316°	•351°	•35°	•180°
•272°	•302°	•298°	•293°	•280°	•180°
•254°	•282°	•335°	•305°	•281°	•180°
•180°	•180°	•180°	•180°	•180°	•180°





Ejemplo numérico



- Los últimos pasos son: primero una supresión de no-máximos....

•4.5	•24.4	•21.3	•50.6	•39.8	•0
•5.7	•23.4	•57.3	•66.3	•43.6	•0
•37.4	•58.5	•57.3	•16.7	•20.8	•0
•68.1	•54.2	•27.3	•30.5	•26.5	•0
•21.2	•6.7	•25.7	•21.5	•19.4	•0
•0	•0	•0	•0	•0	•0

$M[i, j]$

•2	•1	•1	•0	•1	•0
•2	•3	•3	•3	•3	•0
•2	•3	•3	•0	•1	•0
•2	•3	•3	•3	•2	•0
•2	•2	•3	•3	•2	•0
•0	•0	•0	•0	•0	•0

$\zeta[i, j]$

•3	•2	•1
•0	•[i,j]	•0
•1	•2	•3



Ejemplo numérico



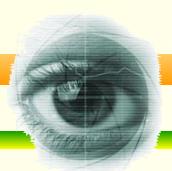
- ...Y segundo una umbralización (en este caso sin histéresis usando un umbral $T = 30$):

•0	•24.4	•0	•50.6	•0	•0
•0	•0	•57.3	•66.3	•0	•0
•0	•58.5	•57.3	•0	•0	•0
•68.1	•54.2	•0	•0	•26.5	•0
•0	•0	•0	•0	•0	•0
•0	•0	•0	•0	•0	•0

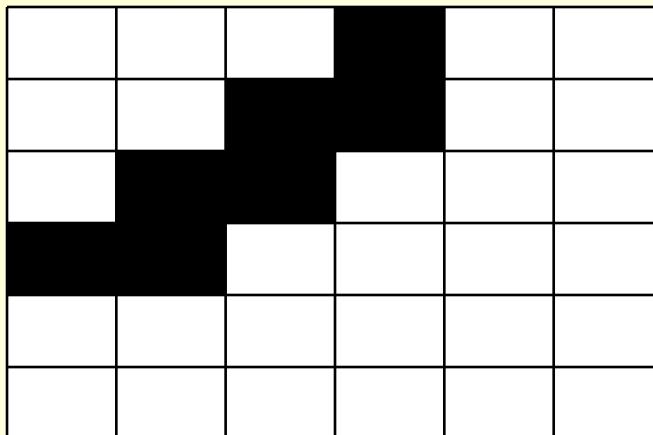
- $E[i, j]$

•0	•0	•0	•1	•0	•0
•0	•0	•1	•1	•0	•0
•0	•1	•1	•0	•0	•0
•1	•1	•0	•0	•0	•0
•0	•0	•0	•0	•0	•0
•0	•0	•0	•0	•0	•0

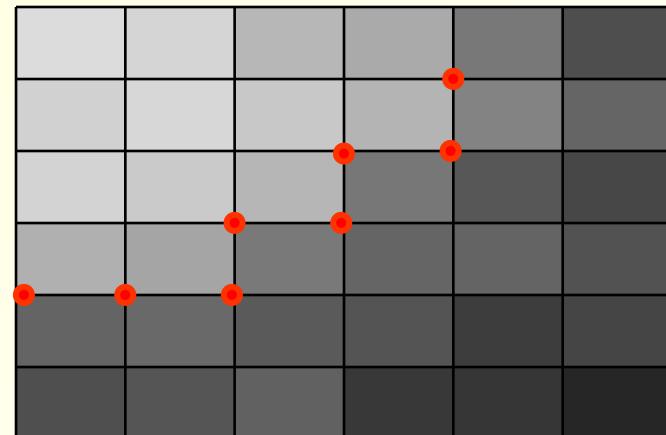
- Resultado final



- Finalmente, visualizamos los píxeles frontera detectados (0 = blanco, 1 = negro) e indicamos las localizaciones precisas de las fronteras (píxeles de las esquinas inferior derecha) en la imagen original:



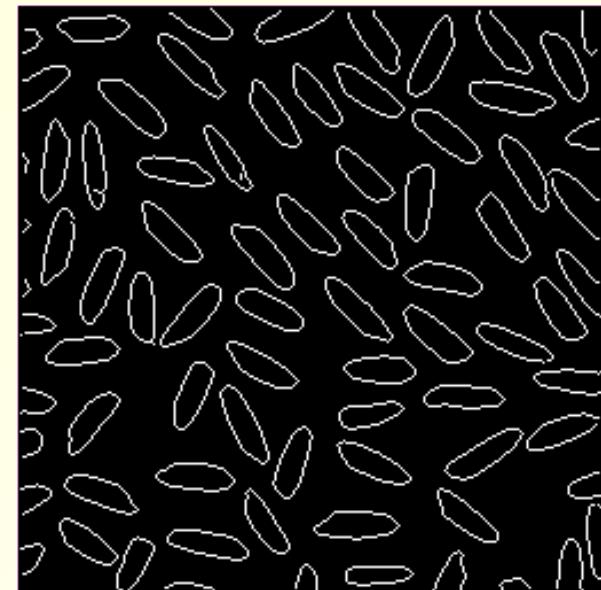
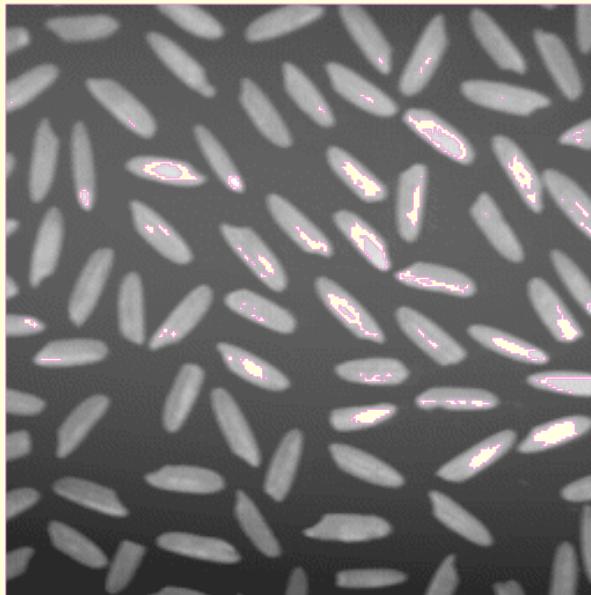
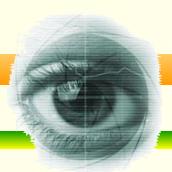
- píxeles frontera



- Localizaciones de los píxeles frontera en la imagen original



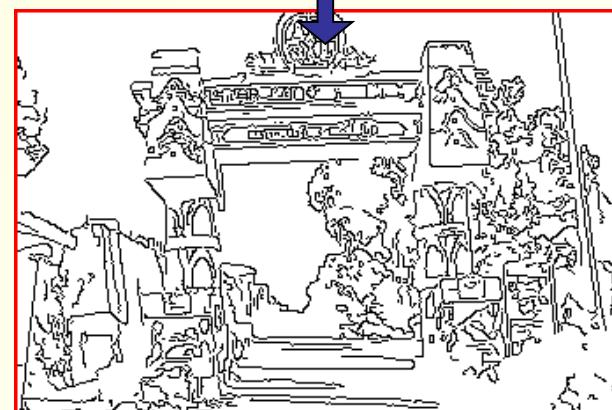
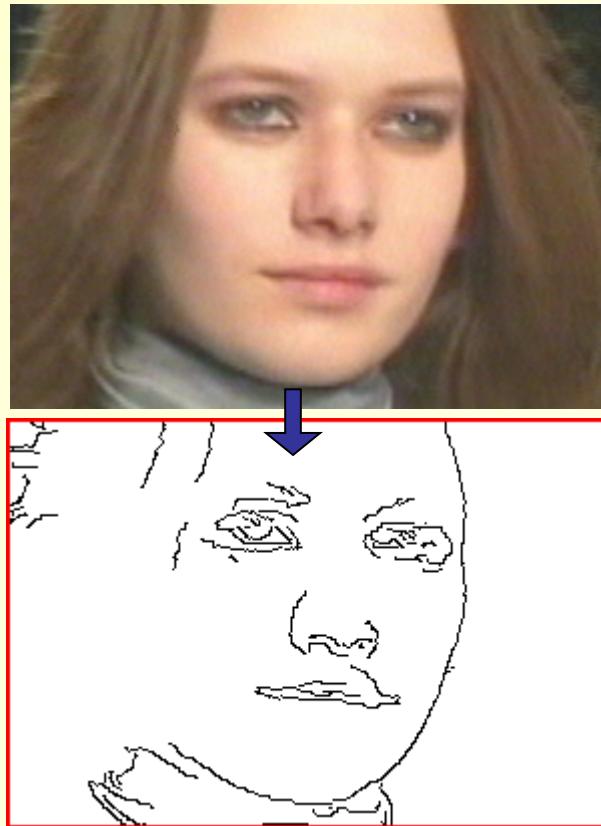
Ejemplo



Detección de bordes por el método de Canny



Ejemplo





Detectores basados en gradiente + umbralización

Sensibles al ruido

Grosor de fronteras superior a 1 pixel

Fronteras cortadas

Algoritmo de Canny

Óptimo

Detección de fronteras (derivada de gaussiana)

Localización de fronteras

Supresión de no máximos

Histéresis

Problema: Puntos donde convergen distintas fronteras

Parámetros del algoritmo

Escala

Umbrales de la histéresis

