Procesamiento Digital de Imágenes

Pablo Roncagliolo B.

Nº 17

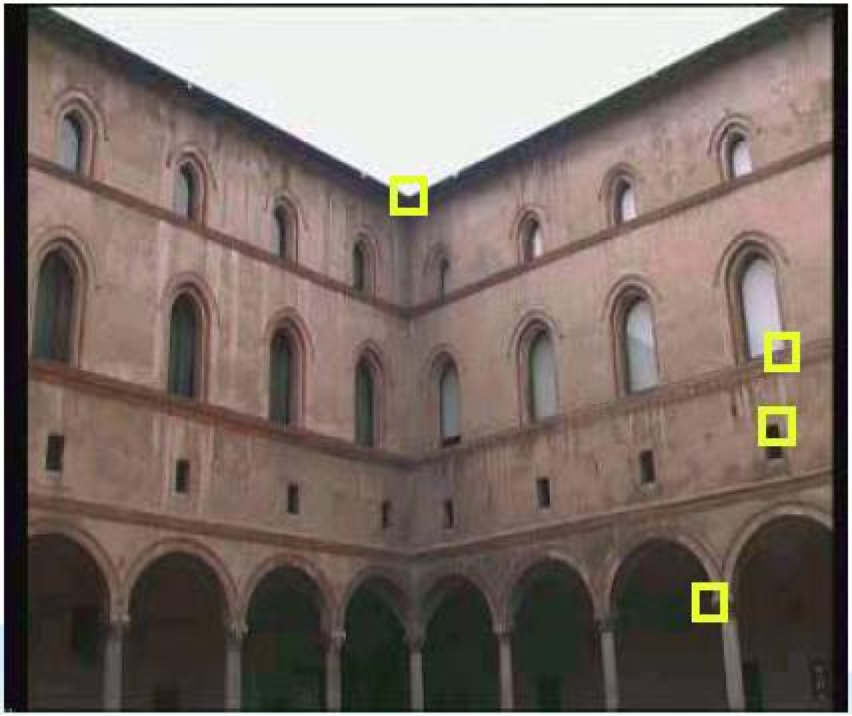
Orden de las clases...

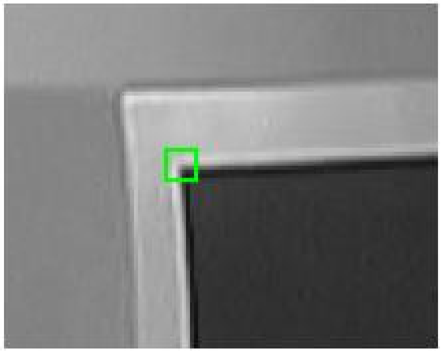
- **CAPTURA, DIGITALIZACION Y ADQUISICION DE IMAGENES** ü
- **TRATAMIENTO ESPACIAL DE IMAGENES** ü
- **TRATAMIENTO EN FRECUENCIA DE IMAGENES** ü
- **RESTAURACION DE IMAGENES** ü
- **OPERACIONES MORFOLOGICAS** ü
- **PROCESAMIENTO DE IMÁGENES EN COLORES** ü
- **COMPRESION DE IMAGENES** ü
- **REPRESENTACION Y DESCRIPCION**
- **SEGMENTACION DE IMAGENES**
- **TOPICOS AVANZADOS**

DETECCION DE ESQUINAS

Extracción de esquinas

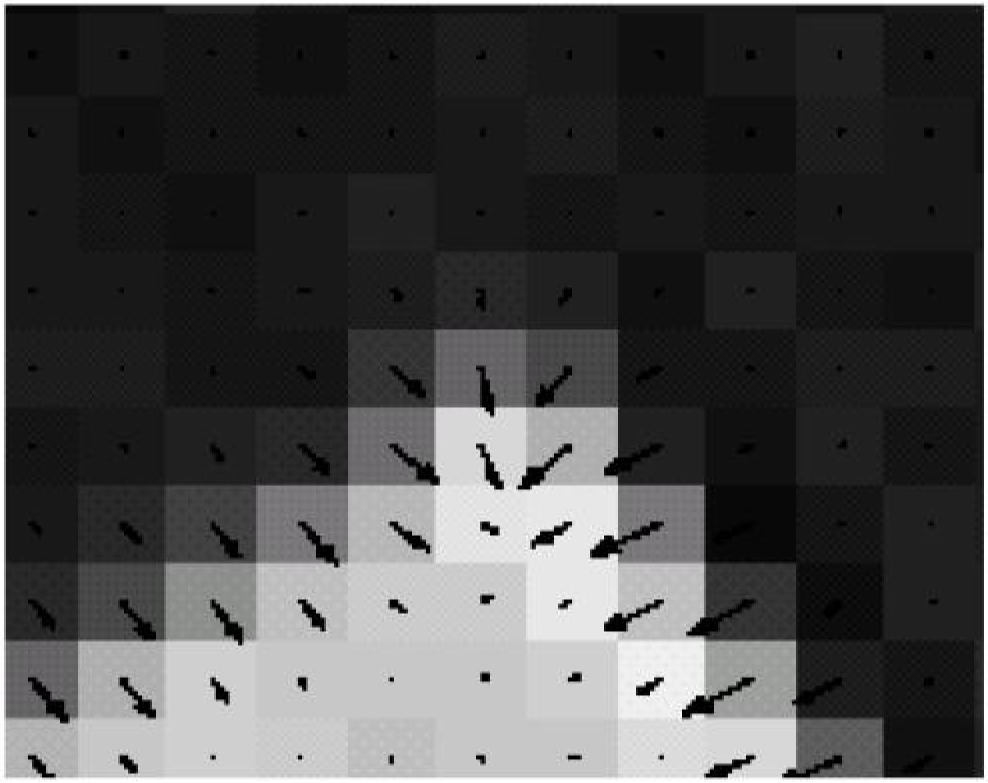
Nivel semántico: intersección de dos bordes rectos è sólo en imágenes ideales.





Extracción de esquinas

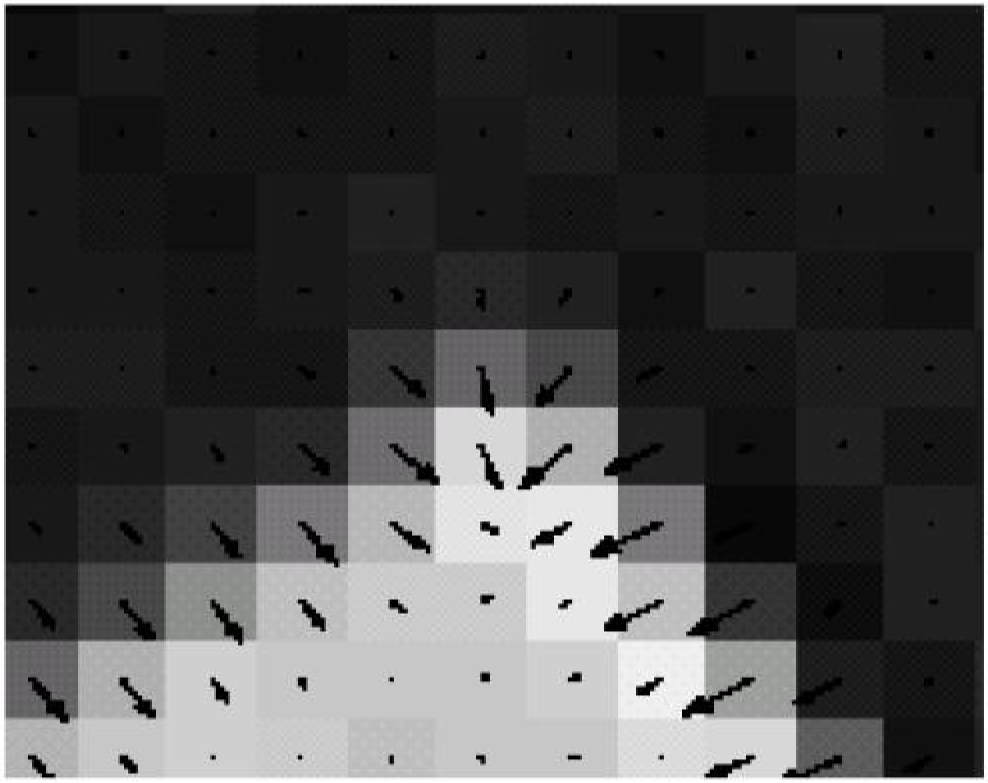
¿Qué caracteriza a una esquina?



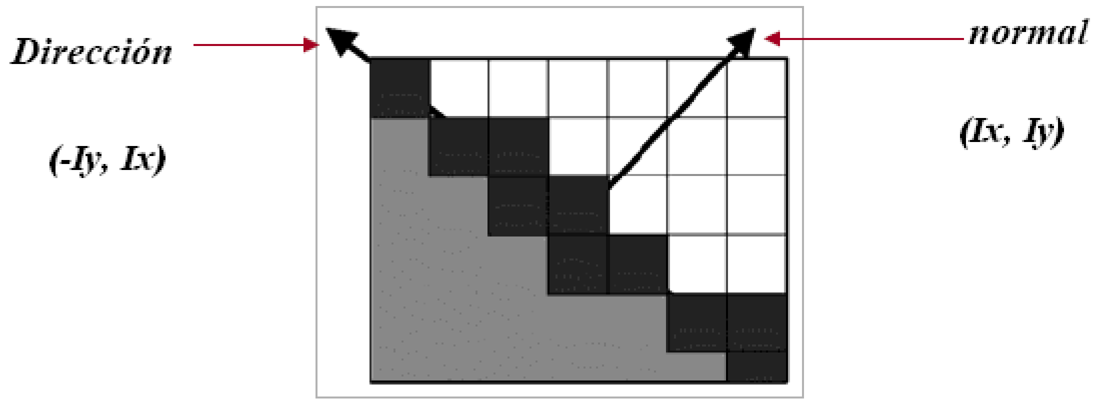
Se observa que en los "bordes" el gradiente aumenta... ¿pero qué ocurre en la esquina?

Extracción de esquinas

Una esquina se caracteriza por un abrupto cambio en la dirección del gradiente.



Extracción de esquinas



Sea

I(x,y) *= imagen*
I_x(x,y), I_y(x,y) *= Imágenes "gradientes" en eje x e y (derivada en x e y)*
Vg=(Ix, Iy) ... *el vector gradiente*
Vmin=(-Iy, Ix) ... *el vector con mínima variación (borde)*

$q = \tan^{-1}(I_y / I_x)$ *= razón de gradientes... dirección del gradiente...*
q_x *y* q_y *las derivadas de la dir. del gradiente... entonces:*

$$K = (q_x , q_y) \cdot Vminm > U$$

K representa la variación de la dirección del gradiente en la dirección de Vmin.

Extracción de esquinas

$$\frac{\partial \tan^{-1}(u)}{\partial x} = \frac{1}{u^2+1}$$

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

$$\frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial x} = \frac{1}{\frac{I_y{}^2}{I_x{}^2}+1} \cdot \frac{I_x I_{xy} - I_y I_{xx}}{I_x{}^2} = \frac{I_x I_{xy} - I_y I_{xx}}{I_x{}^2 + I_y{}^2}$$

$$\frac{\partial \tan^{-1}\left(\frac{I_y}{I_x}\right)}{\partial y} = \frac{1}{\frac{I_y{}^2}{I_x{}^2}+1} \cdot \frac{I_x I_{yy} - I_y I_{yx}}{I_x{}^2} = \frac{I_x I_{yy} - I_y I_{yx}}{I_x{}^2 + I_y{}^2}$$

Extracción de esquinas

$$= \left(\frac{I_x I_{xy} - I_y I_{xx}}{I_x{}^2 + I_y{}^2}, \frac{I_x I_{yy} - I_y I_{yx}}{I_x{}^2 + I_y{}^2} \right) \cdot \frac{(-I_y, I_x)}{\sqrt{I_x^2 + I_y^2}}$$

$$= \left(\frac{I_x I_y I_{xy} - I_y^2 I_{xx}}{I_x{}^2 + I_y{}^2} + \frac{I_x^2 I_{yy} - I_x I_y I_{yx}}{I_x{}^2 + I_y{}^2} \right) \frac{1}{\sqrt{I_x^2 + I_y^2}}$$

$$= \frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2 I_x I_y I_{yx}}{\left(I_x{}^2 + I_y{}^2\right)^{3/2}}$$

Extracción de esquinas 1

Este método habitual consiste en el uso de derivadas de segundo orden.

Consiste en la medición de la razón de cambio de la dirección del gradiente respecto de la magnitud del gradiente. (Kitchen&Rosenfeld)

$$E = \frac{f_{xx} f_y^2 + f_{yy} f_x^2 - 2 f_{xy} f_x f_y}{(f_x^2 + f_y^2)^{3/2}} \geq U$$

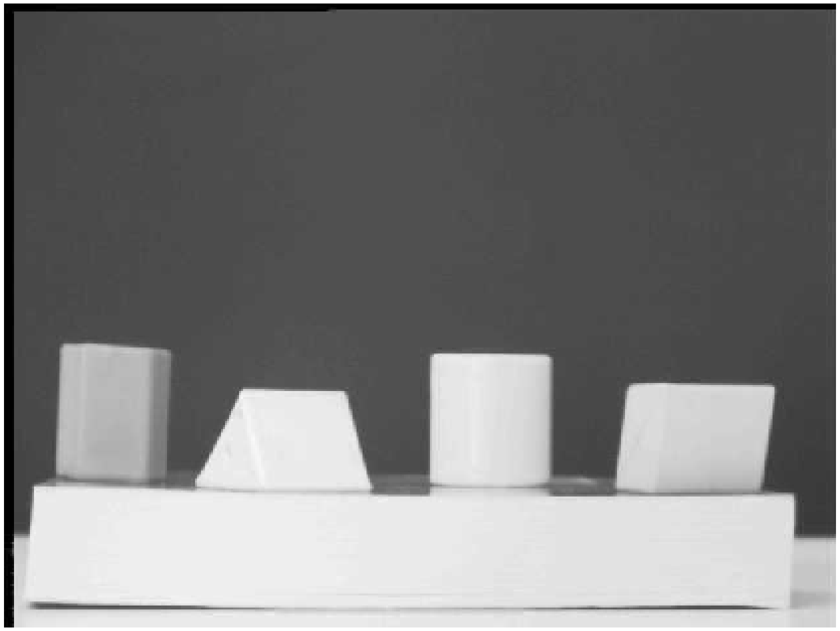
Extracción de esquinas 1

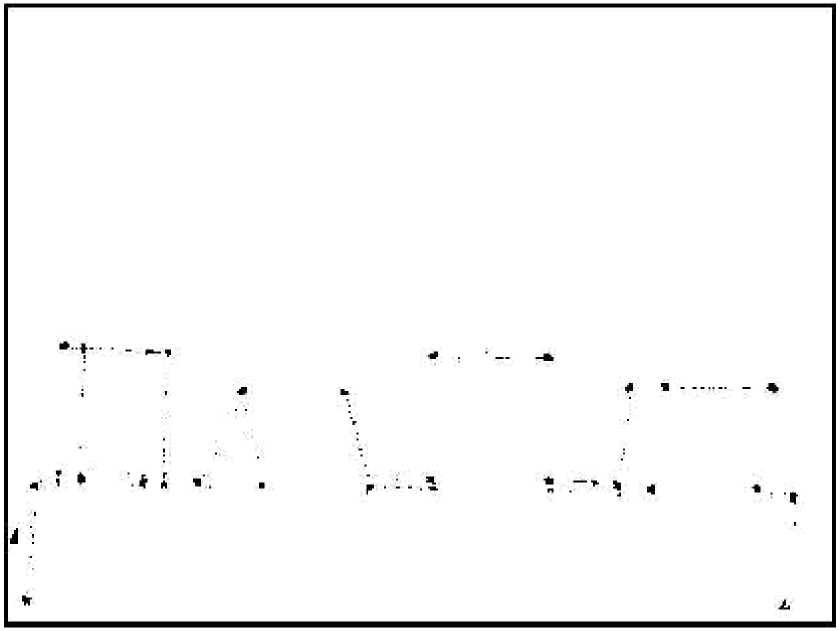
Las operaciones se realizan sobre cada píxel. Para el calculo de las derivadas se utiliza la convolución con los siguientes filtros:

$$f_x = \frac{\partial f}{\partial x} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad f_y = \frac{\partial f}{\partial y} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$f_{xx} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) \quad f_{yy} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) \quad f_{xy} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right)$$

Extracción de Esquinas 1





Algoritmo

```
I=double(imread('figura.bmp'));
subplot(1,2,1); imshow(I);
DX=[-1  0  1;
    -1  0  1;
    -1  0  1];
DY=[-1 -1 -1;
     0  0  0;
     1  1  1];

Ix=conv2(I,DX,'same');      Iy=conv2(I,DY,'same');
Ixx=conv2(Idx,DX,'same');  Iyy=conv2(Iy,DY,'same');
Ixy=conv2(Iy,DX,'same');

E=((Iy.^2).*Iyy+(Iy.^2).*Ixx-2*Ixy.*Ix.*Iy)./((Ix.^2+Iy.^2).^(3/2));

E=abs(E);E=255*(E/max(max(E)));
subplot(1,2,2);surf(E);
```

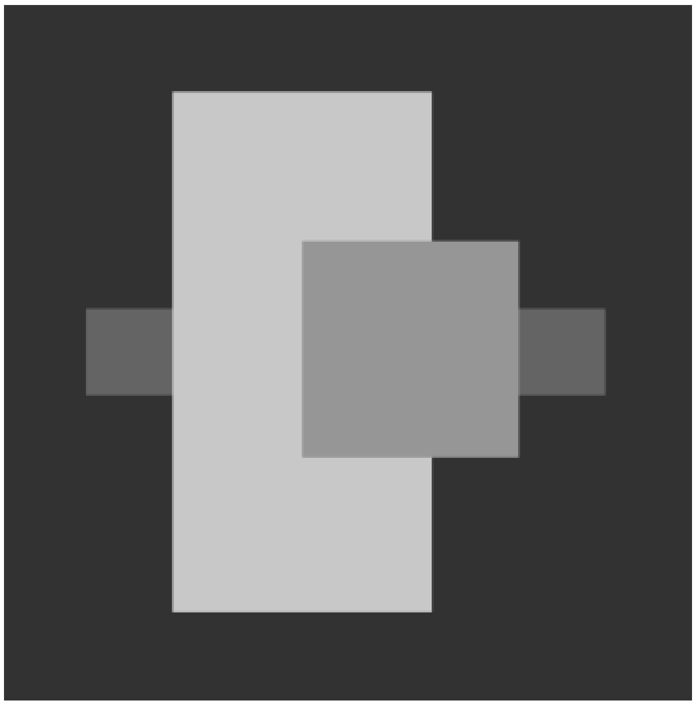
Extracción de esquinas 2

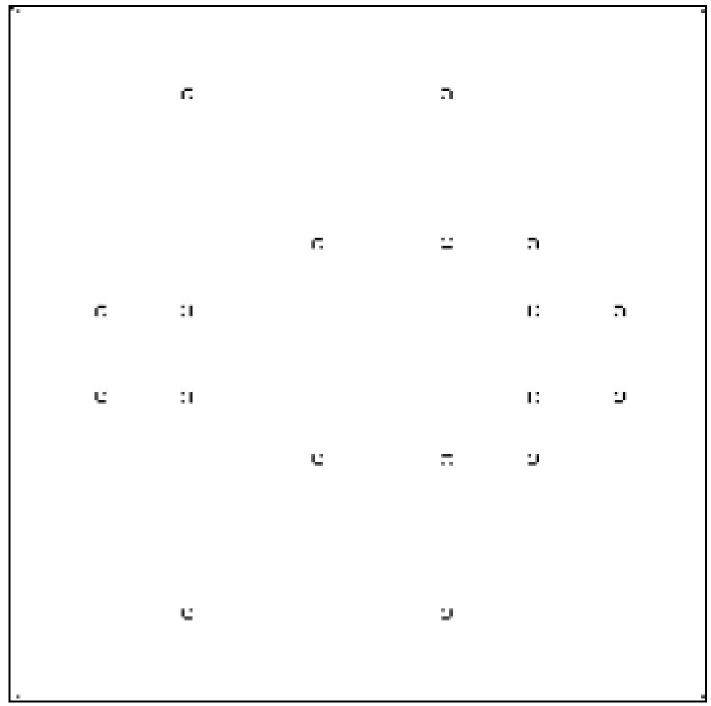
Otro método consiste en medir la curvatura Gaussiana de una determinada superficie (imagen).

La curvatura K se expresa en términos de derivadas parciales (Lipschutz):

$$K = \frac{f_{xx} f_{yy} - f_{xy}^{2}}{(1 + f_x^{2} + f_y^{2})^{2}} \geq U$$

Extracción de Esquinas 2





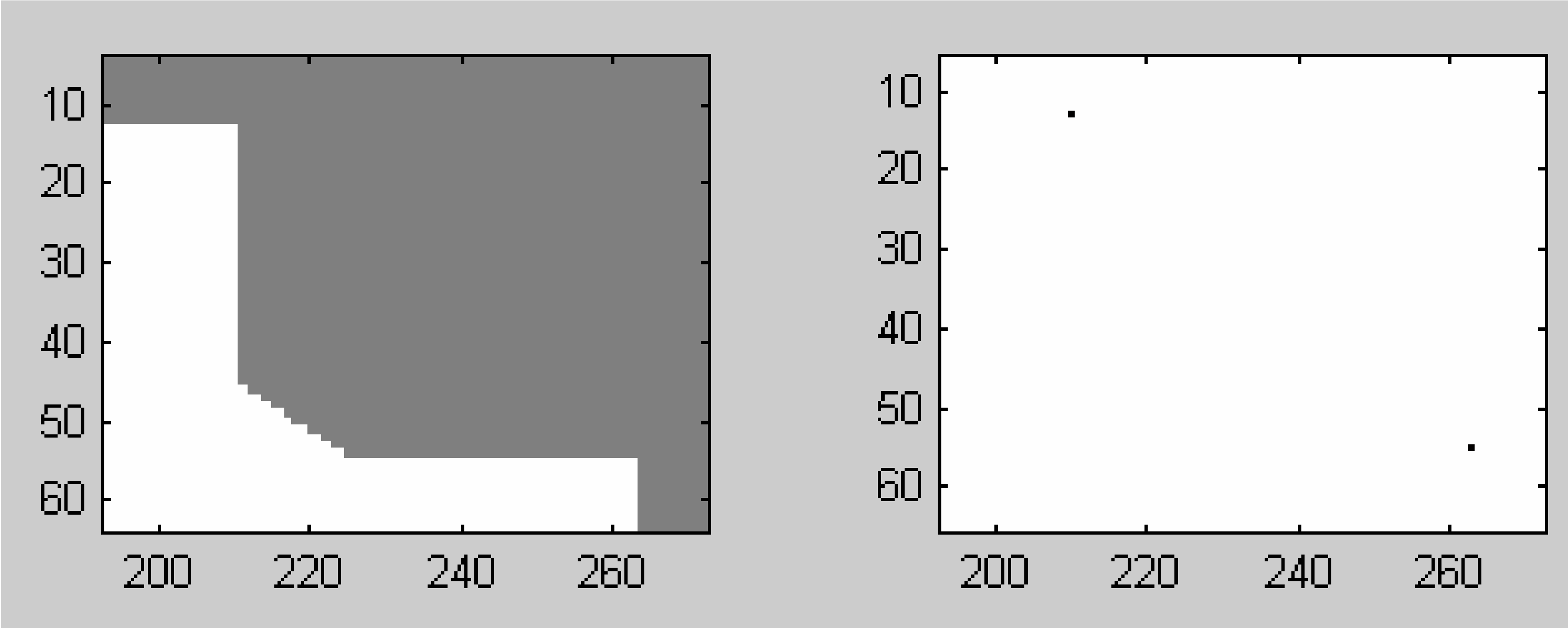
Extracción de esquinas 3

Finalmente se pueden mencionar los métodos clásicos de detección de esquinas específicas. Consisten simplemente en la convolución con núcleos que "representen" la esquina buscada:

$$E_{\mathrm{sup\_der}} = \begin{bmatrix} -4 & 5 & 5 \\ -4 & 5 & 5 \\ -4 & -4 & -4 \end{bmatrix} \qquad E_{\mathrm{inf}} = \begin{bmatrix} 5 & 5 & 5 \\ -4 & 5 & -4 \\ -4 & -4 & -4 \end{bmatrix}$$

Son ocho posible rotaciones del núcleo.

Extracción de esquinas 3



Algoritmo

```
 map1=(0:255)/255;
 map=[map1' map1' map1'];
 COLORMAP(map);
 A1=double(imread('figura.bmp'));
 subplot(2,2,1);
 image(A1);

D1=[-4  5  5;
    -4  5  5;
    -4 -4 -4];

 E=conv2(A1,D1,'same');

 E=255*(E/max(max(E)));
 E=255*(E<220);
 subplot(2,2,2);image(E);
```

DETECCION DE PUNTOS DE INTERÉS

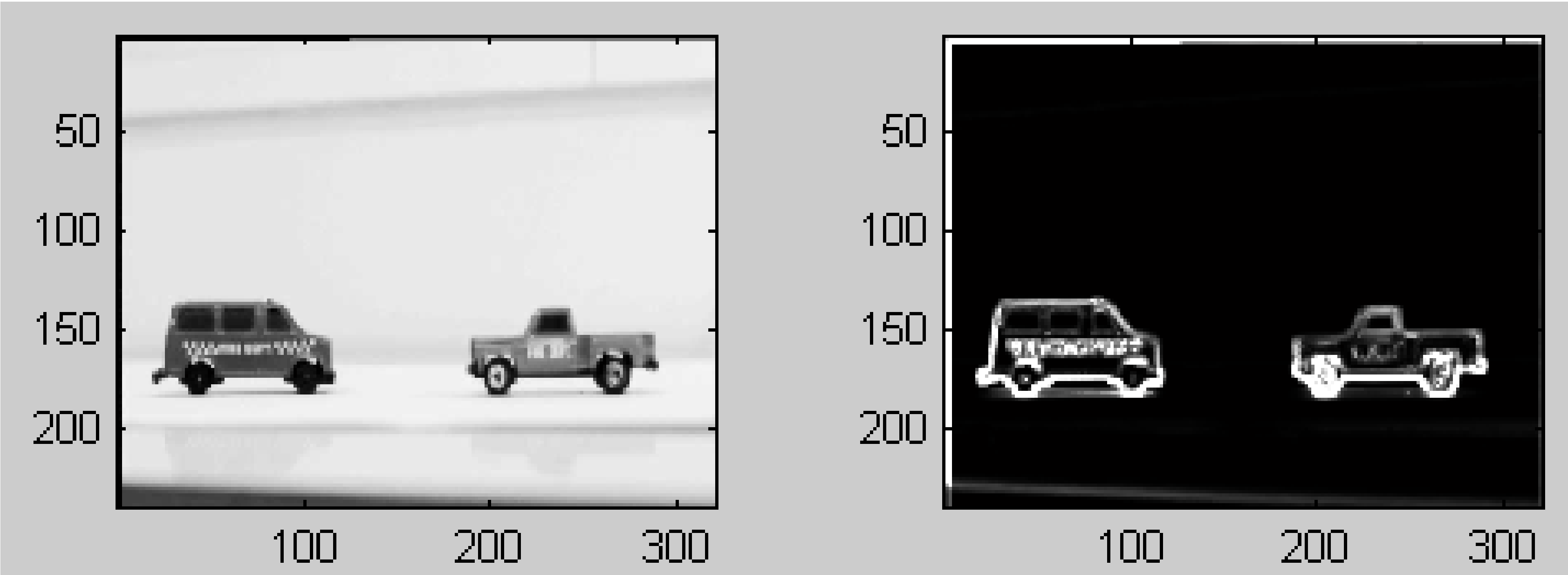
Extracción de puntos de interés

Los puntos de interés o zonas de interés, generalmente se delimitan por grupos de píxeles con una gran varianza.

$$VAR(x,y) = \sum_{i,j \in \text{vecinos}} \left[I(x,y) - I(x+i, y+j) \right]^2$$

El resultado se puede binarizar o "amplificar" para filtrar los "puntos de interés".

Extracción de puntos de interés



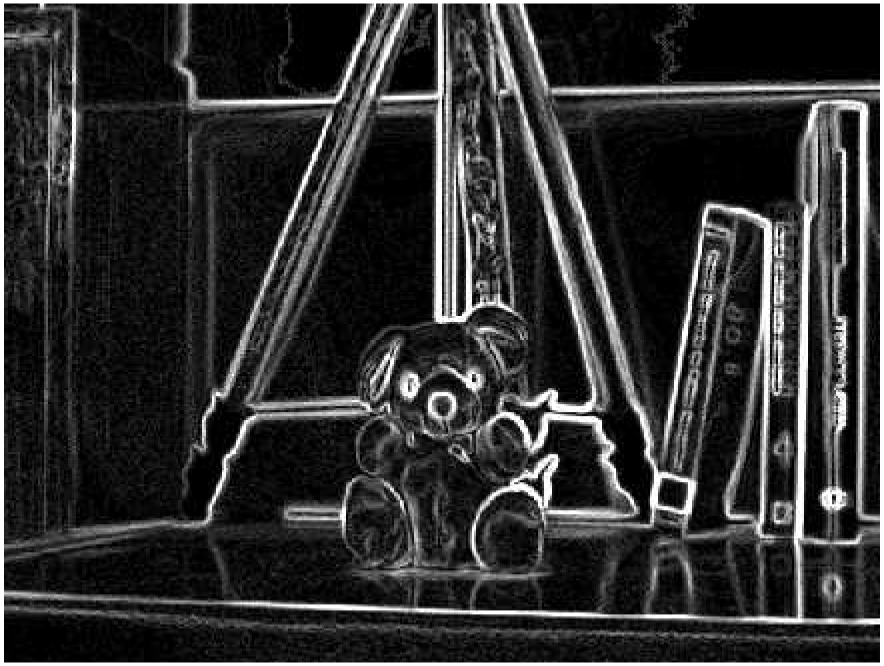
Algoritmo

```
map1=(0:255)/255;
map=[map1' map1' map1'];  COLORMAP(map);
A1=double(imread('autos.bmp'));
subplot(2,2,1);image(A1);
[H,W]=size(A1);
V=A1;

D=2;
for x=1+D:W-D
   for y=1+D:H-D
      var=0;
      for xx=-D:D
         for yy=-D:D
           var=var+(A1(y,x)-A1(y+yy,x+xx))^2;
         end;
      end;
      V(y,x)=var;
   end;
end;

V=V-min(min(V)); V=255*(V/max(max(V))); %normaliza
V=V*6; %amplifica
subplot(2,2,2);  image(V);
```

DETECCION DE LINEAS

Descripción de Líneas...

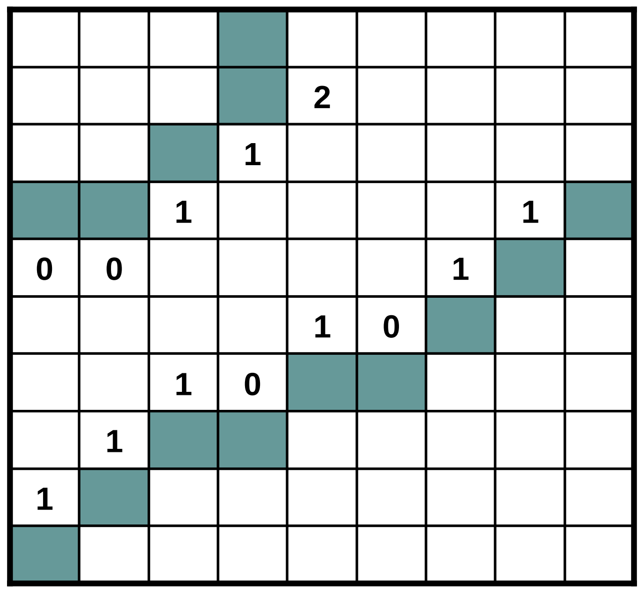
Generalmente en una imagen de bordes se desea "filtrar" ciertos bordes especiales, por ejemplo, dejar sólo las líneas rectas



1. Códigos de Cadena

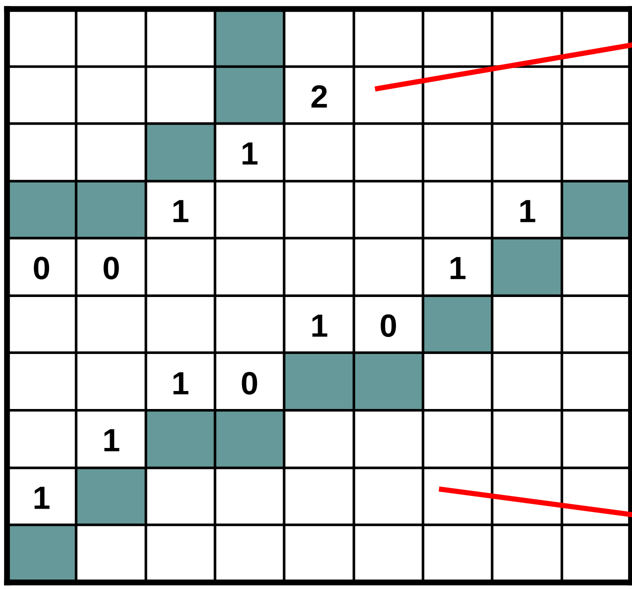
Un método corresponde a buscar “códigos de cadenas”. Corresponde a etiquetar los puntos de un borde, con el ángulo (índice) de su vecino siguiente, siguiendo el sentido del borde.

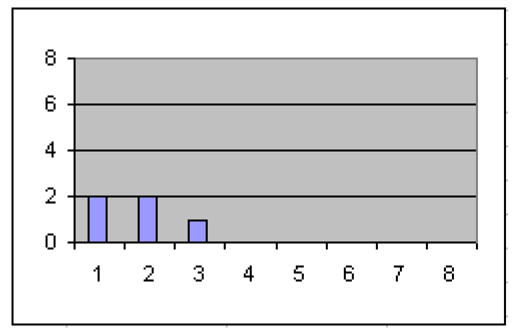
3	2	1
4		0
5	6	7



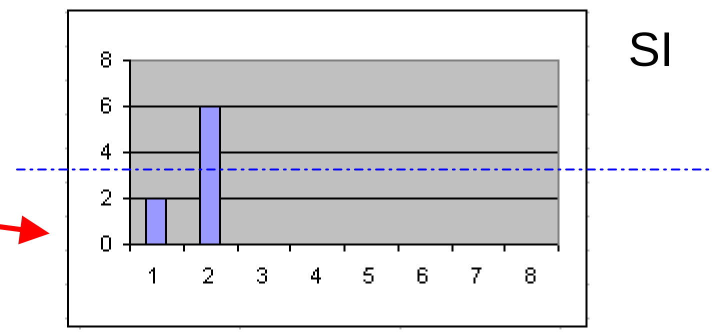
1. Códigos de Cadena

Para cada borde o segmento, luego de etiquetar con el número de vecino, se determina un histograma de los índices.





NO



SI

Si en dicho histograma existe algún índice que supere notablemente al resto, se puede considerar una recta.

1. Códigos de Cadena

Algunas condiciones:

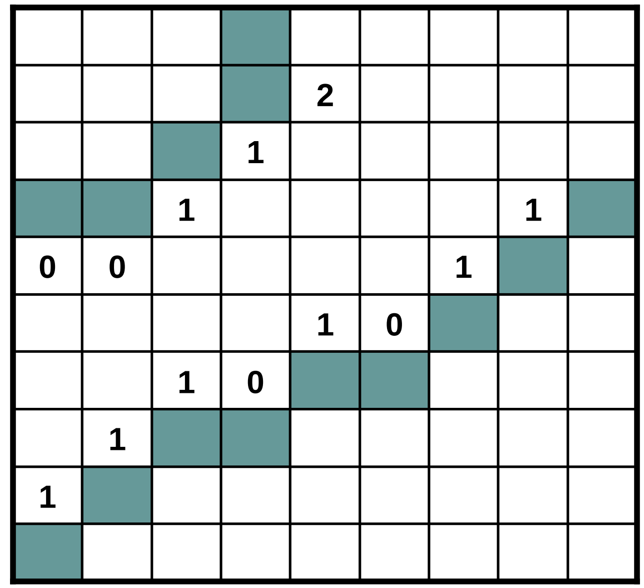
1.- Si el histograma tiene valores en más de 4 índices è no es recta

2.- Si el histograma tiene valores sólo de 1 índice è es una recta pura.

3.- Si el hist. tiene 2 barras. Significativas:

.- Si son adyacentes, pero una supera notoriamente a la otra **è** es recta.

.- Si no son adyacentes, **è** no es recta.



2. Ajuste de líneas: mínimos cuadrados

Para un conjunto de puntos se desea encontrar *a* y *b* tal que la siguiente expresión sea mínima:

$$\sum_{i=1}^{n}\left[(b + ax_i) - y_i\right]^2$$

Una solución para este problema clásico es derivando e igualando a cero para buscar el mínimo.

2. Ajuste de líneas: mínimos cuadrados

$$\sum_{i=1}^{n}\left[(b+ax_i)-y_i\right]^2 = \sum_{i=1}^{n}\left[(b+ax_i)^2-2(b+ax_i)y_i+y_i^{\,2}\right]$$

$$=\sum_{i=1}^{n}\left[b^2+2abx_i+a^2x_i^{\,2}-2by_i-2ax_iy_i+y_i^{\,2}\right]$$

derivando parcialmente...

$$\frac{\partial}{\partial a}=2b\sum_{i=1}^{n}x_i+2a\sum_{i=1}^{n}x_i^2-2\sum_{i=1}^{n}x_iy_i$$

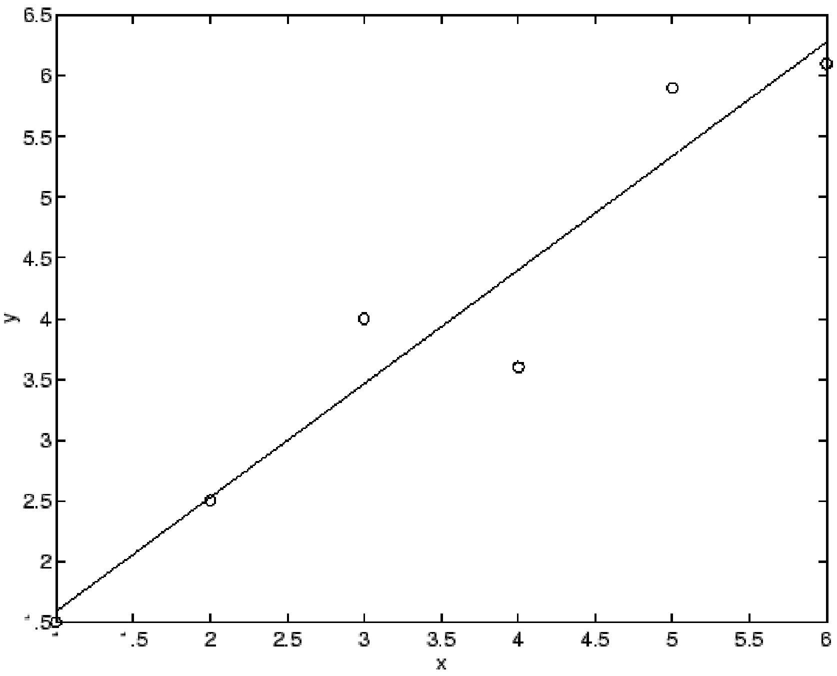
$$\frac{\partial}{\partial b}=2b\cdot n+2a\sum_{i=1}^{n}x_i-2\sum_{i=1}^{n}y_i$$

2. Ajuste de líneas: mínimos cuadrados

$$despejando:$$

$$a = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

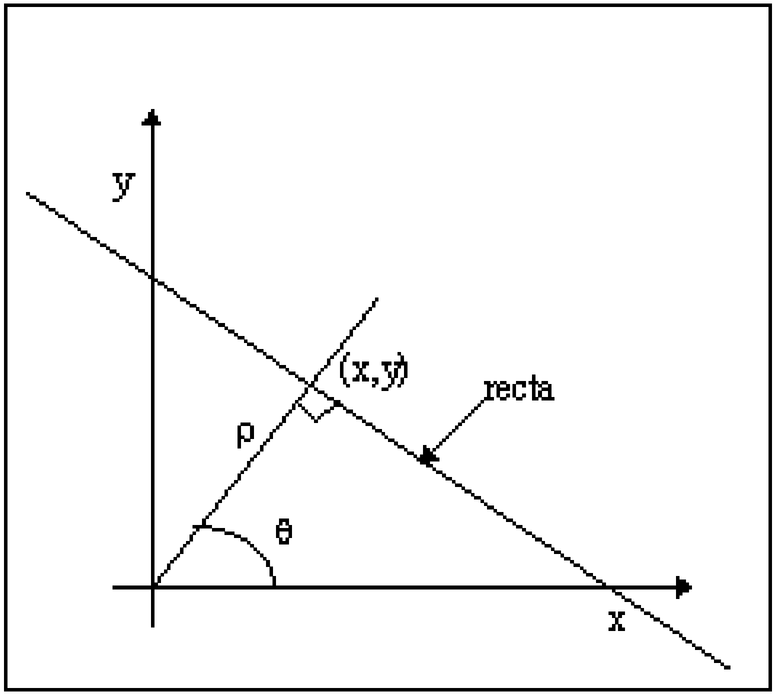
$$b = \frac{\sum y - a\sum x}{n}$$



3. Transformada Hough

La transformada Hough se utiliza para el enlace de puntos de borde y la extracción de rectas. Implica la transformación de coordenadas Cartesianas a coordenadas polares de la forma:

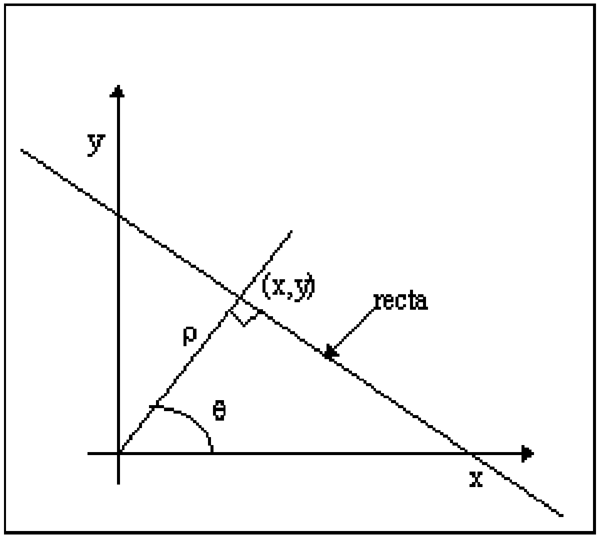
$$r = x\cos q + y\sin q$$



Los dos inconvenientes principales de esta transformada son que no es capaz de encontrar los extremos de la recta, y que la transformada ha sido patentada, por lo que su uso en algún proyecto requiere el pago de royalties a la familia Hough.

3. Transformada Hough

Desarrollo:



$$\text{Sea}: (x_1, y_1)(x_2, y_2)$$

$$\text{Recta}:$$

$$y = ax + b$$

$$\text{Donde}:$$

$$a = \frac{(y_2 - y_1)}{(x_2 - x_1)}, \qquad b = y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} x_1$$

Para obtener el punto x_0,y_0, se puede minimizar la función de distancia desde el origen hasta los puntos de la recta.

3. Transformada Hough

Desarrollo:

Distancia al origen:

$$d = \sqrt{(x-0)^2 + (y-0)^2}$$

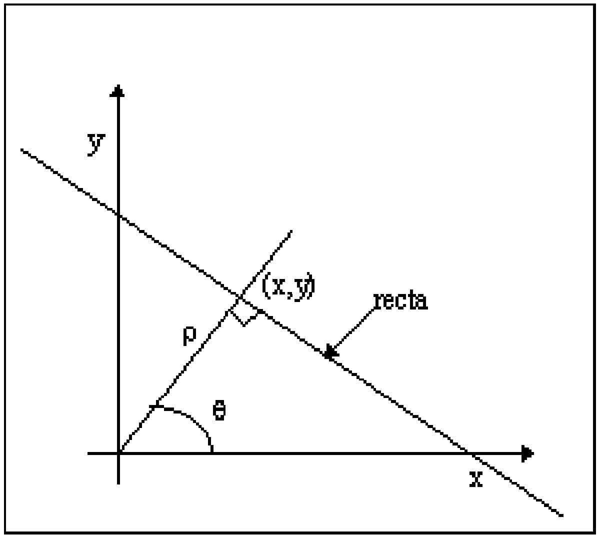
$$d' = x^2 + (ax+b)^2 = x^2 + a^2x^2 + 2abx + b^2$$

Derivando:

$$2x + 2xa^2 + 2ab = 0$$

Entonces:

$$x_0 = \frac{-ab}{1+a^2}, \qquad y_0 = b - \frac{a^2 b}{1+a^2}$$



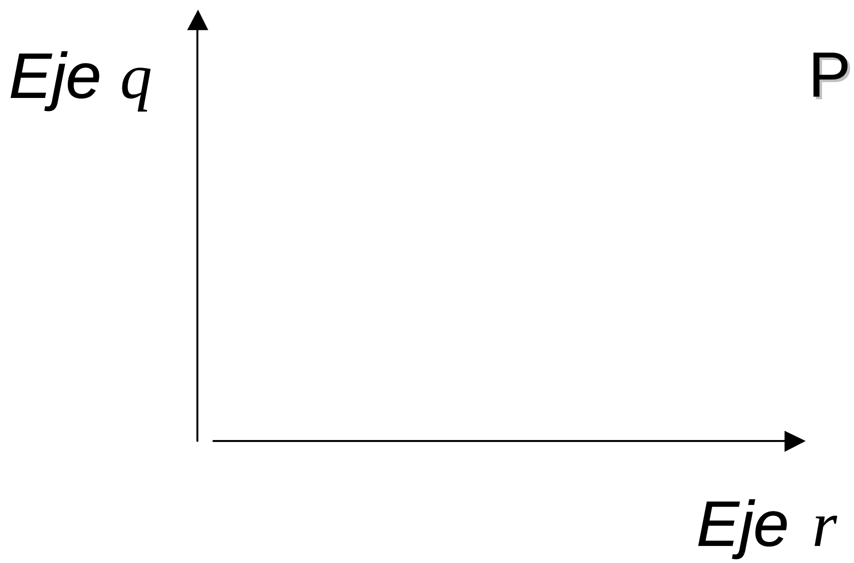
3. Transformada Hough

Desarrollo:

$$\text{Ecuación polar}:$$

$$r = \sqrt{x_0 + y_0}, \qquad q = \arctan(y_0 / x_0)$$

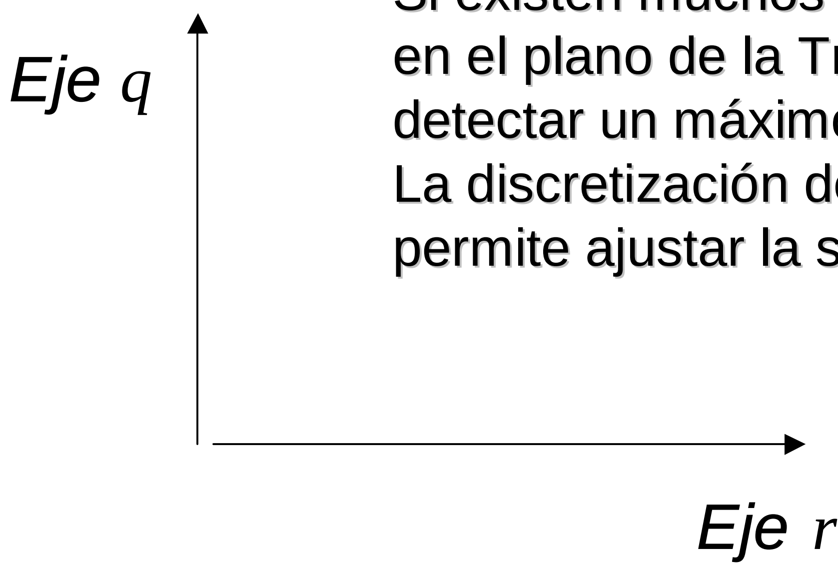
Plano de la Transformada



3. Transformada Hough

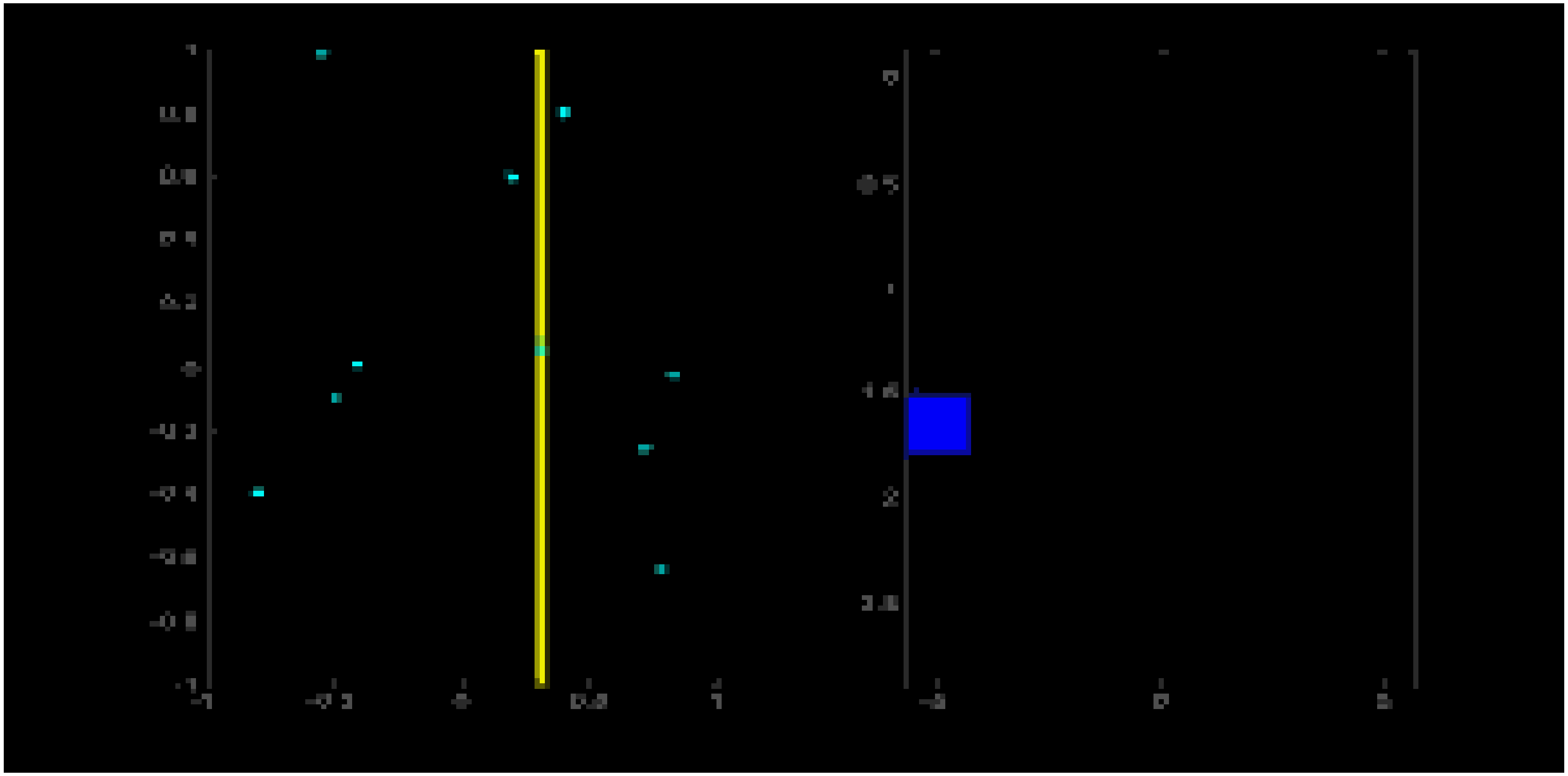
La transformada consiste en contabilizar en una matriz o "plano de la transformada" todas las rectas en su representación polar.

Si existen muchos pares de puntos en una "recta", en el plano de la Transformada se debería detectar un máximo.
La discretización de los ejes de la transformada permite ajustar la sensibilidad de la detección.



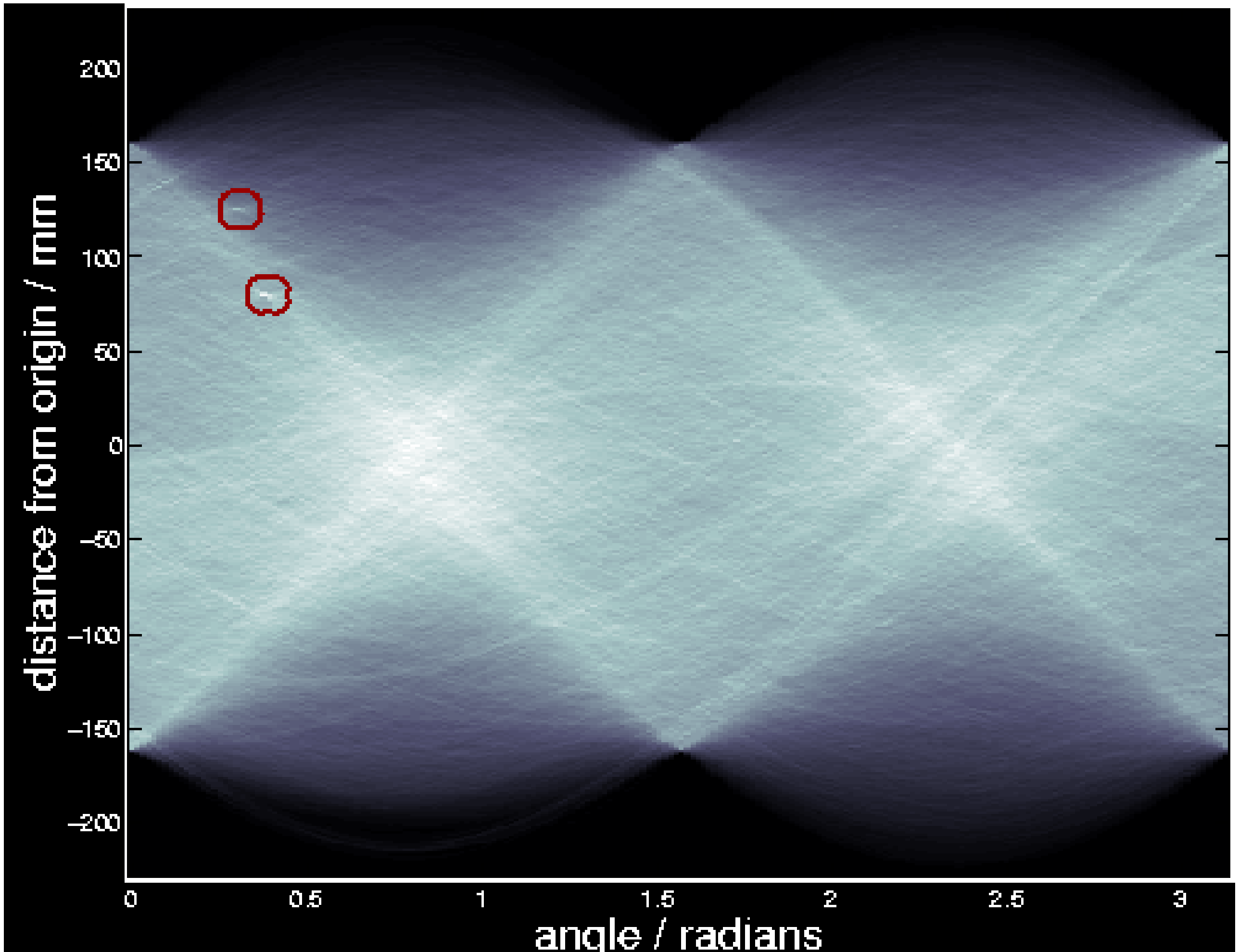
3. Transformada Hough

Transformada...

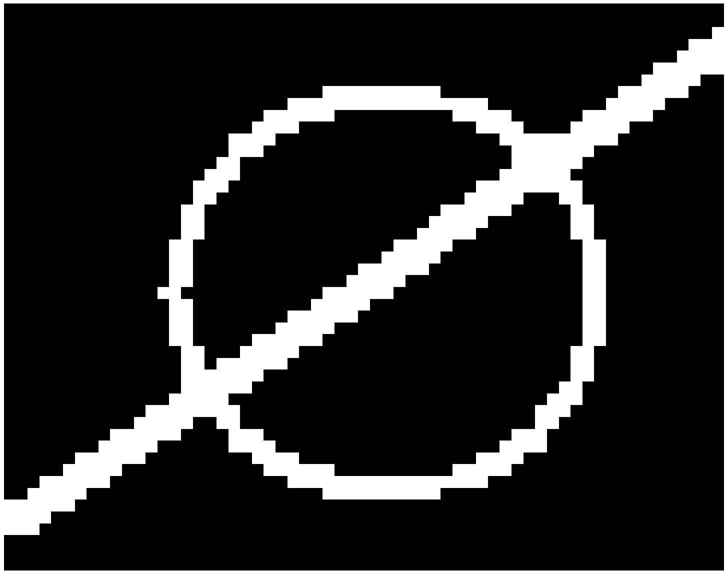


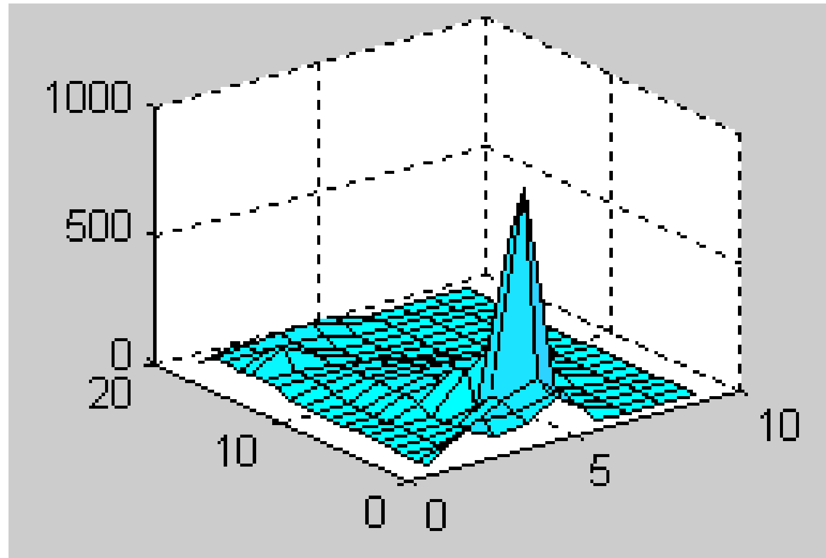
3. Transformada Hough

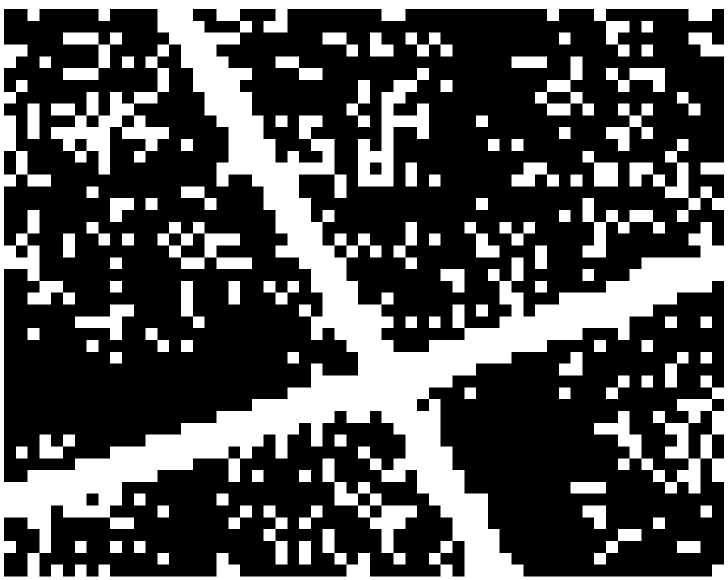
Transformada...

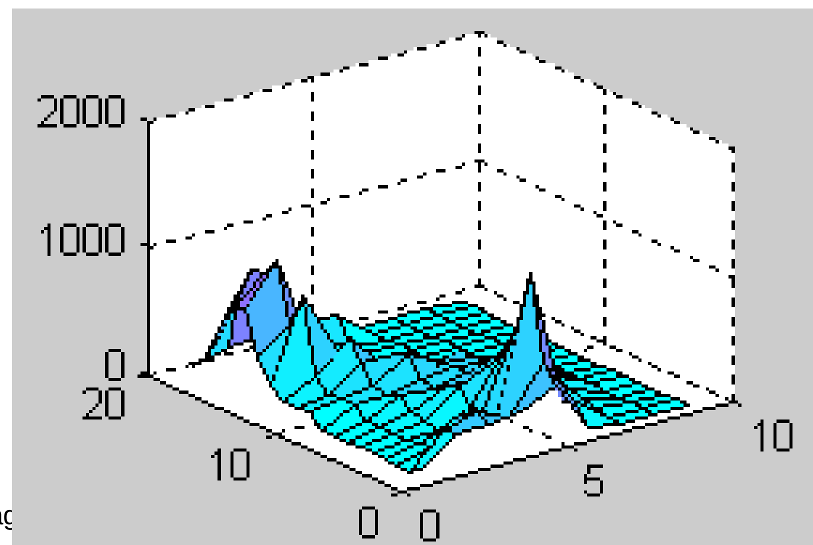


3. Transformada Hough (Delta=10)

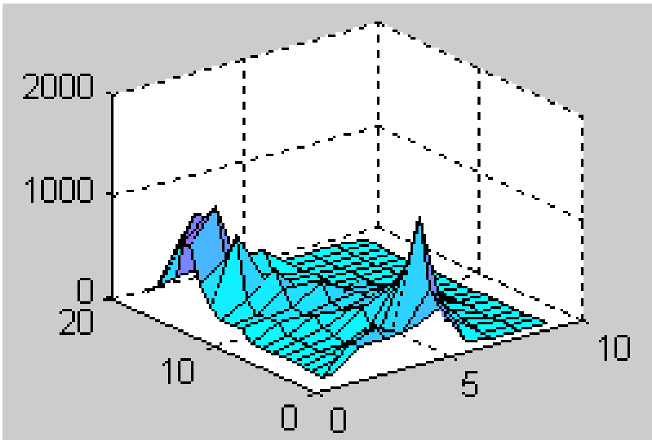


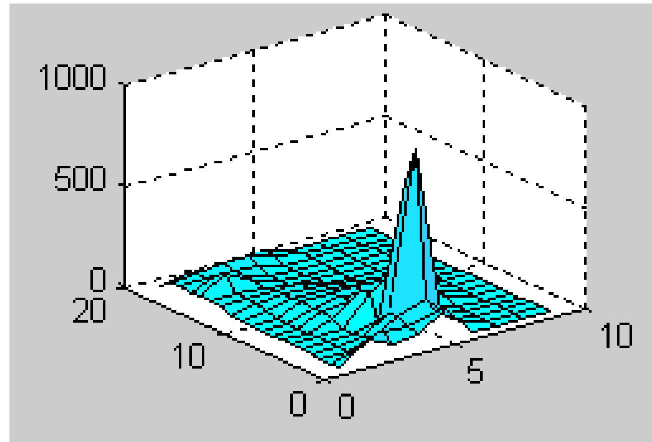


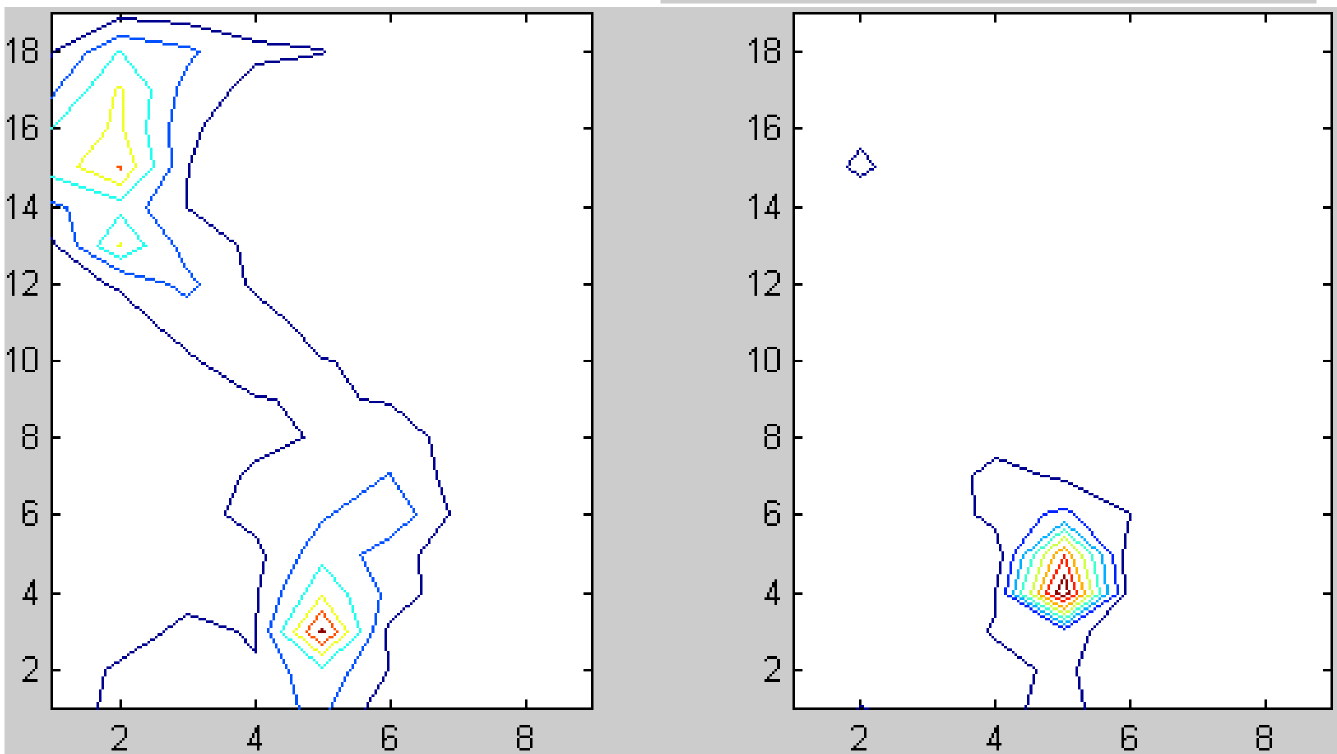


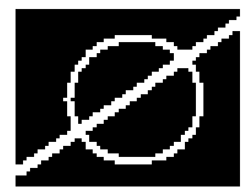


3. Transformada Hough (Delta=10)

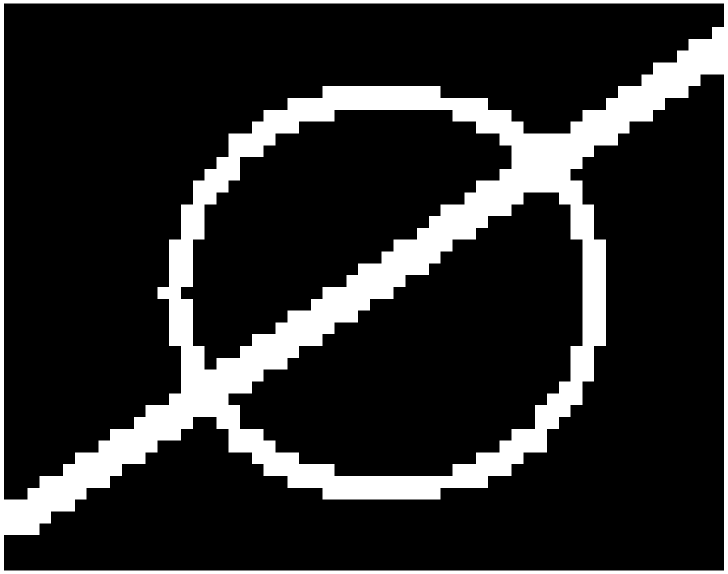


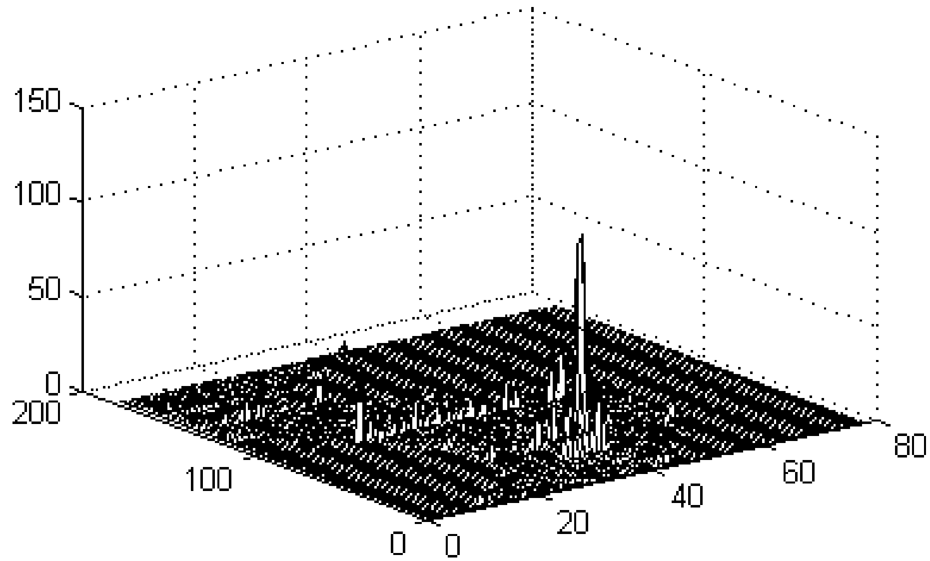


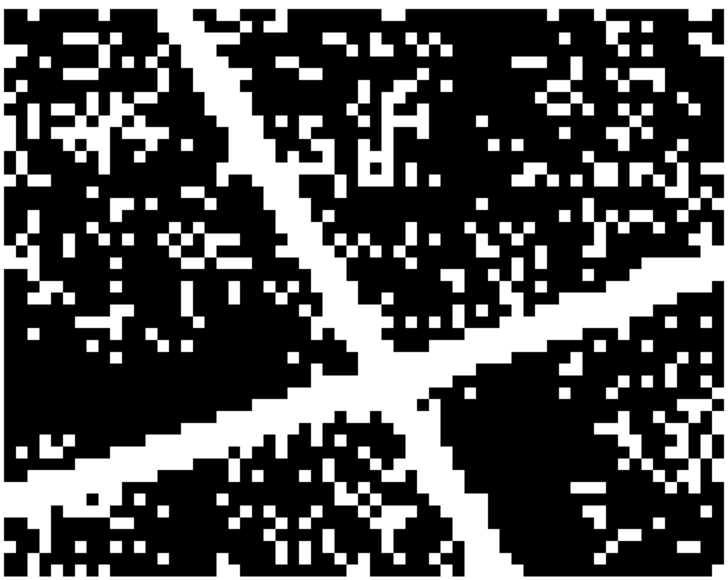




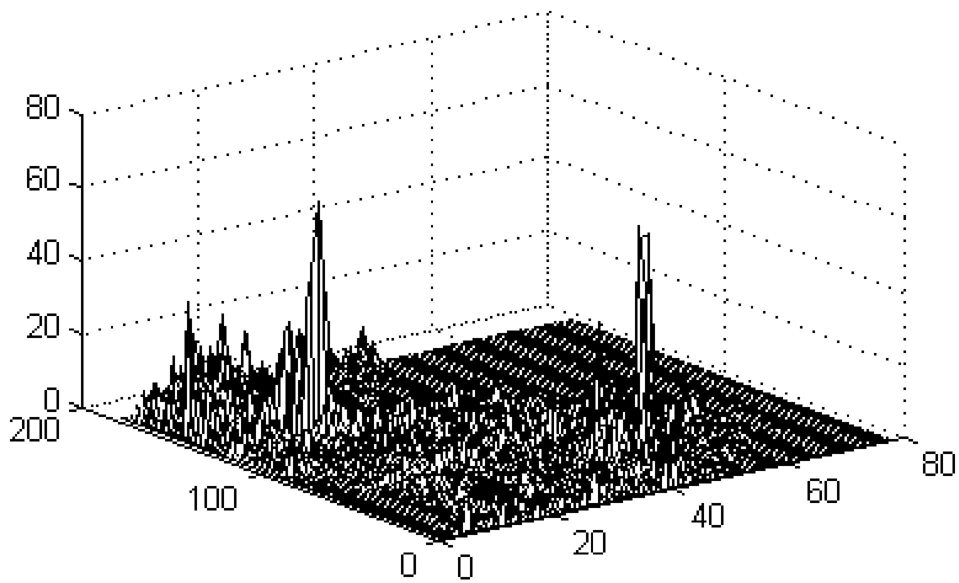
3. Transformada Hough (Delta=1)



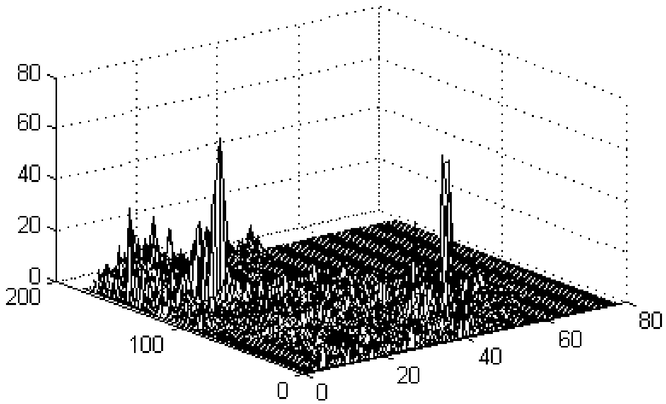


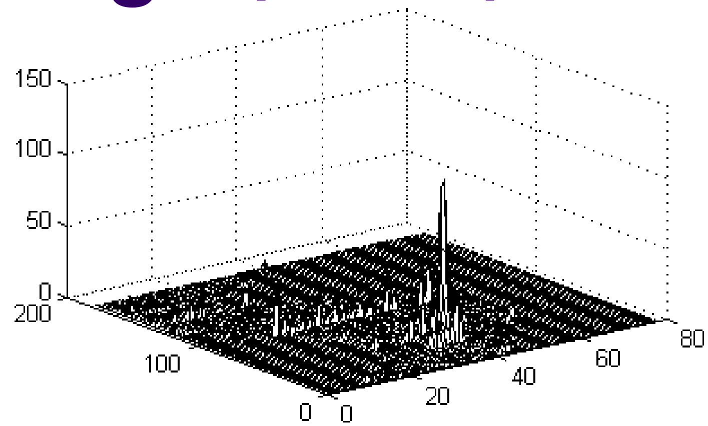


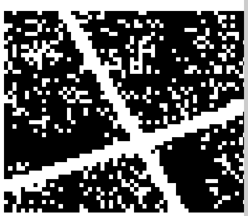
Imág

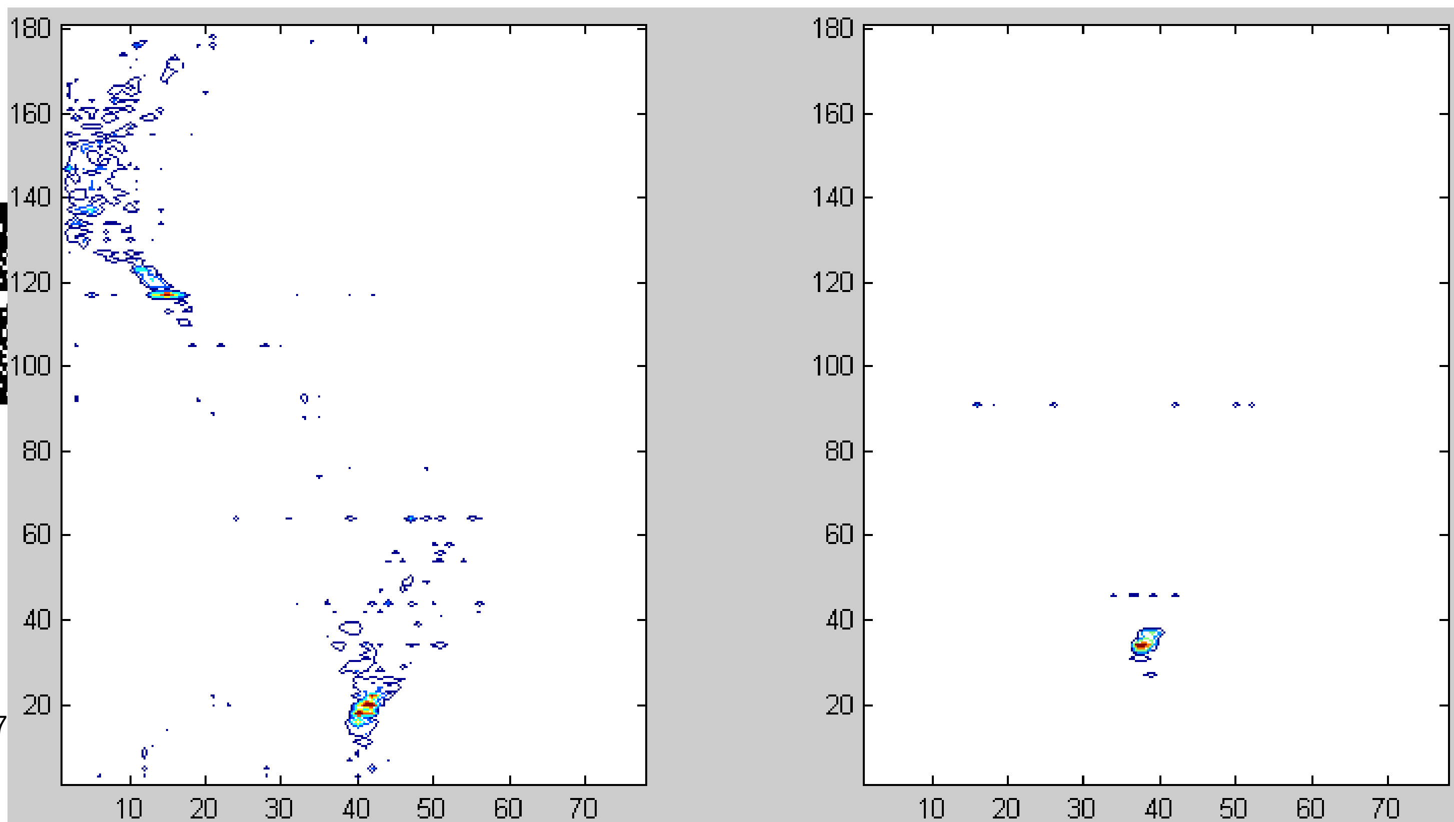


3. Transformada Hough (Delta=1)









3. Algoritmo

```
for f=1:4:nf
 for c=1:4:nc
  if IM(f,c)==1
   for ff=1:4:nf
    for cc=1:4:nc
              if ff~=f & cc~=c & IM(ff,cc)==1
                       %ec recta
                       if cc==c, x0=c;y0=0;
                       else
                               a=(ff-f)/(cc-c);
                               b=ff-a*cc;
                               x0=-a*b/(1+a*a);
                               y0=b-a*a*b/(1+a*a);
                       end;
                      r=sqrt(x0^2+y0^2)+1;
                      if x0==0, ang=90;else ang=atan(y0/x0)*180/pi+90;end;
                      ang=round(ang/DELTA)+1;
                      r=round(r/DELTA)+1;
                      H(ang,r)= H(ang,r)+1;
             end;%if
    end;%for cc
   end;%for ff
  end;%if
 end;%for c
end;%foc f
```

3. Transformada Hough modificadas

Esta transformada también se puede modificar para detectar otras “figuras” o funciones.

Por ejemplo, círculos. Para cada par de puntos existe sólo 1 círculo que los contiene en su borde en posiciones diametrales.

El círculo se puede representar por:

.- RADIO
.- Angulo desde origen hasta el centro
.- Distancia desde el origen hasta el centro

El espacio de la transformada tiene 3 dimensiones **è** se deben detectar “aglomeraciones” de puntos en el espacio.

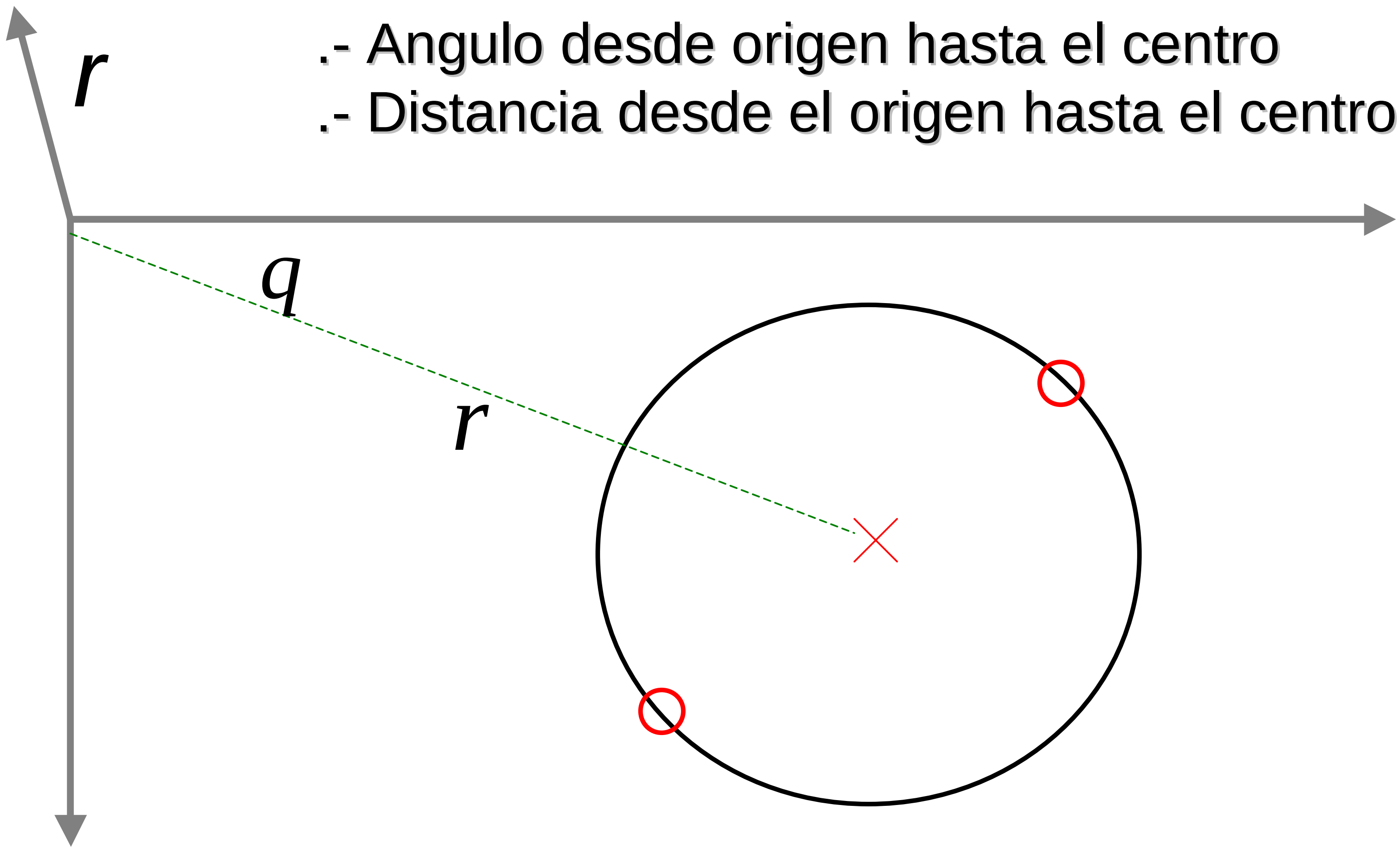
3. Transformada Hough modificadas

El círculo se puede representar por:

.- RADIO

.- Angulo desde origen hasta el centro

.- Distancia desde el origen hasta el centro



3. Transformada Hough modificadas

Una simplificación consiste en representar los círculos sólo por el:

.- Angulo desde origen hasta el centro

.- Distancia desde el origen hasta el centro

Es decir no se considera el radio **è** si existen muchos círculos de diferentes radios, pero con un mismo centro, se detectan como uno solo.

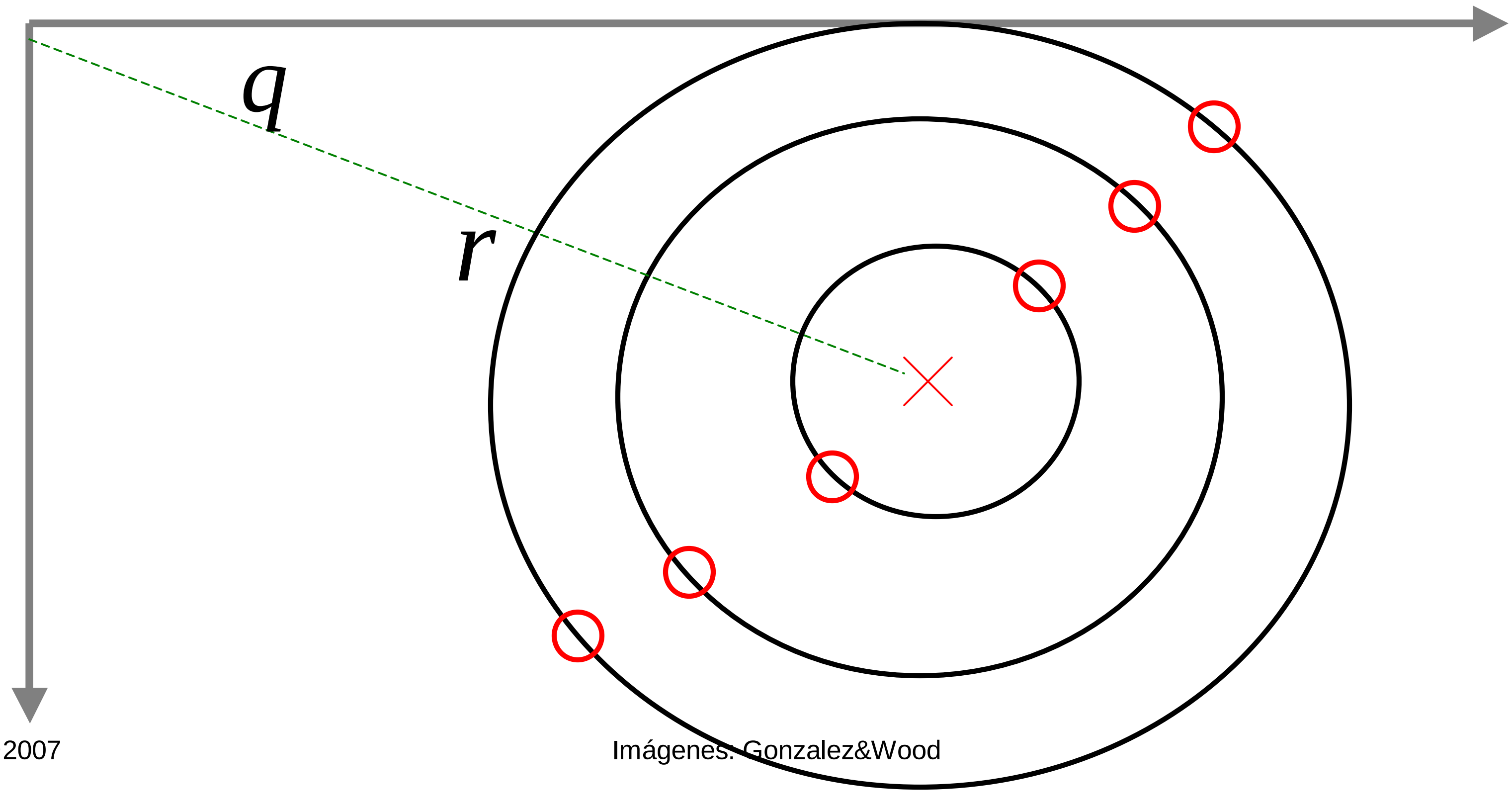
La transformada se puede extender para cualquier función y=f(x).

3. Transformada Hough modificadas

Si el círculo se representa por:

.- Angulo desde origen hasta el centro

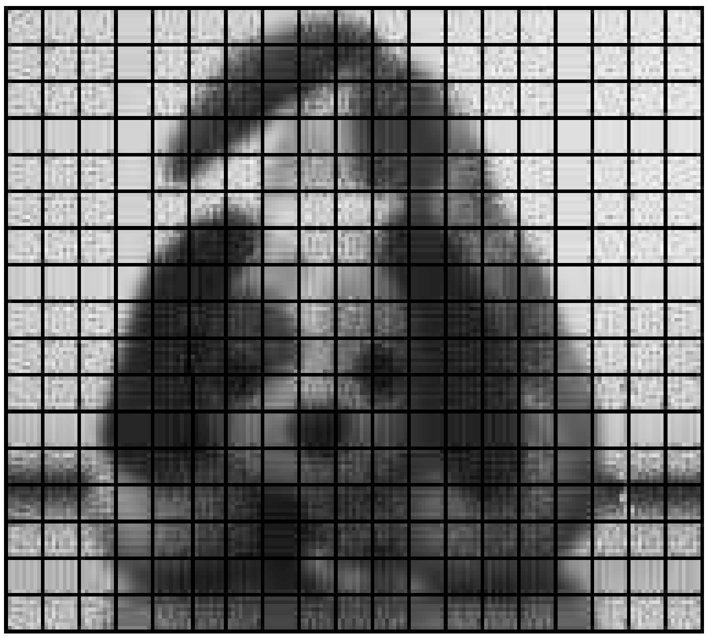
.- Distancia desde el origen hasta el centro

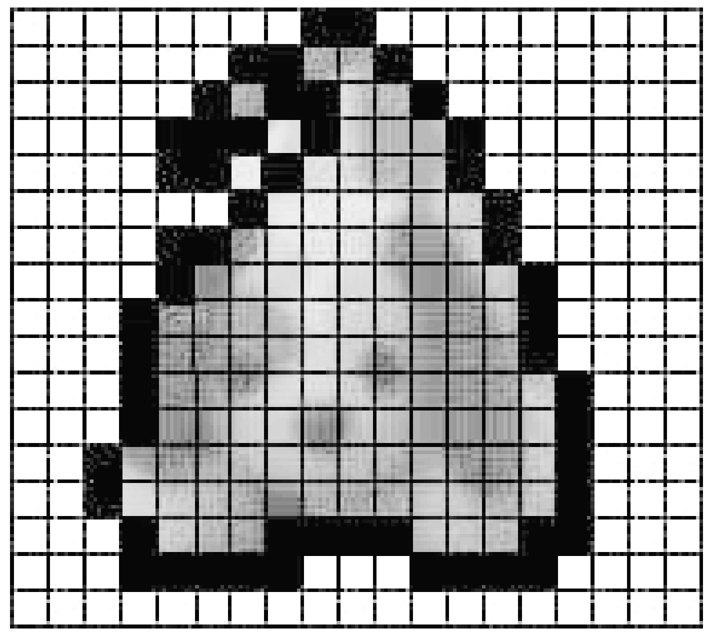


DESCRIPTORES DE FRONTERAS (BORDES)

Descripción de Fronteras...

Generalmente se desea reconocer o describir un cierto objeto delimitado por un borde. Una solución es encontrar el "código de cadena" que representa dicho borde.



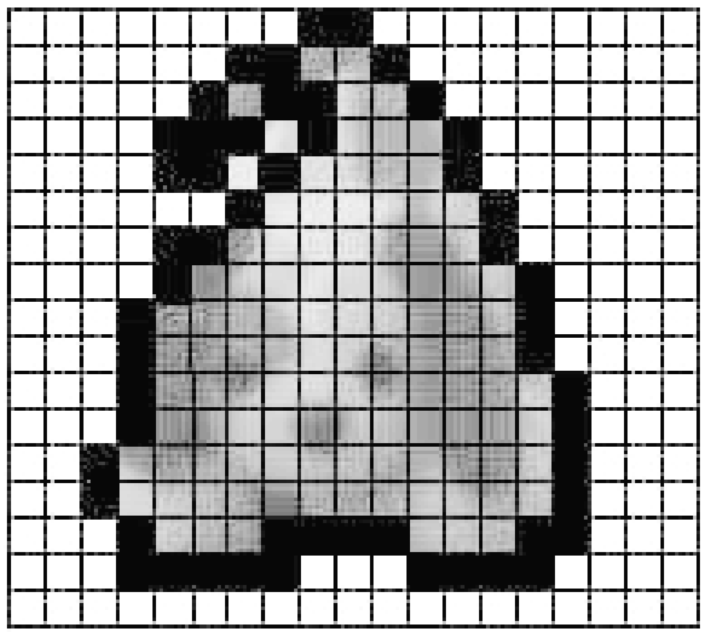


1. Códigos de Cadena

Se puede utilizar vecindad tipo 4 u 8.

Ej. Aplicando código según vecindad tipo 8:

3	2	1
4		0
5	6	7

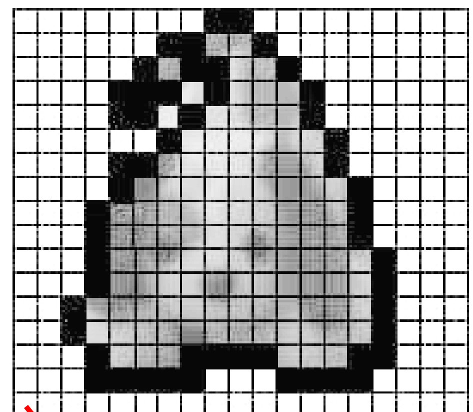


1. Códigos de Cadena

Para comparar diferentes códigos de cadena se debe normalizar a un tamaño fijo. Para ello se puede considerar la “moda” en segmentos.

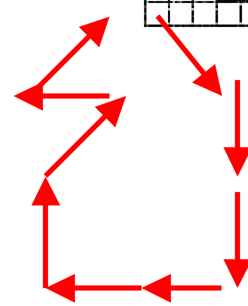
3	2	1
4		0
5	6	7

Ej. Aplicando segmentos de 5:



0-7-7-7-6-7-6-7-6-6-6-6-4-6-4-4-4-3-4-4-4-6-4-4-4-2-3- 2-1-2-0-1-1-1-2-4-5-5-4-2-1-1-0-1

7-6-6-4-4-2-1-4-1



1. Códigos de Cadena

Para comparar dos códigos de cadena normalizados, se puede determinar la correlación cruzada.

3	2	1
4		0
5	6	7

135	90	45
180		0
225	270	315

Sea :

$$C_{ab} = \frac{1}{n}\sum_{i=1}^{n} a_i b_i$$

Donde :

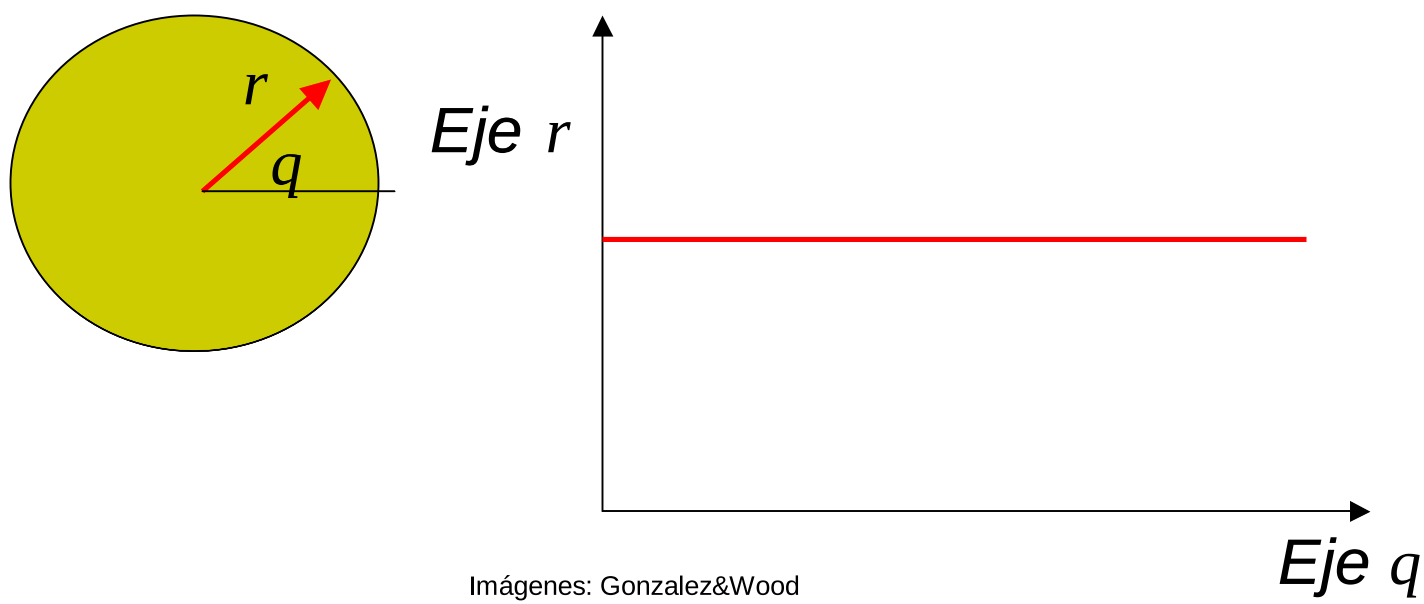
$$a_i b_i = \cos(\angle a_i - \angle b_i)$$

Si las cadenas son similares el coeficiente C tiende a 1.

2. Signaturas

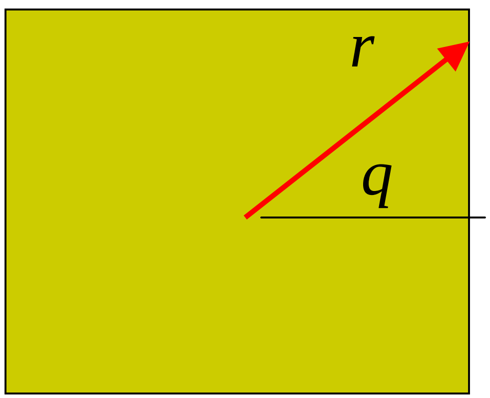
Otro método consiste en determinar la función o curva de signatura de un objeto.

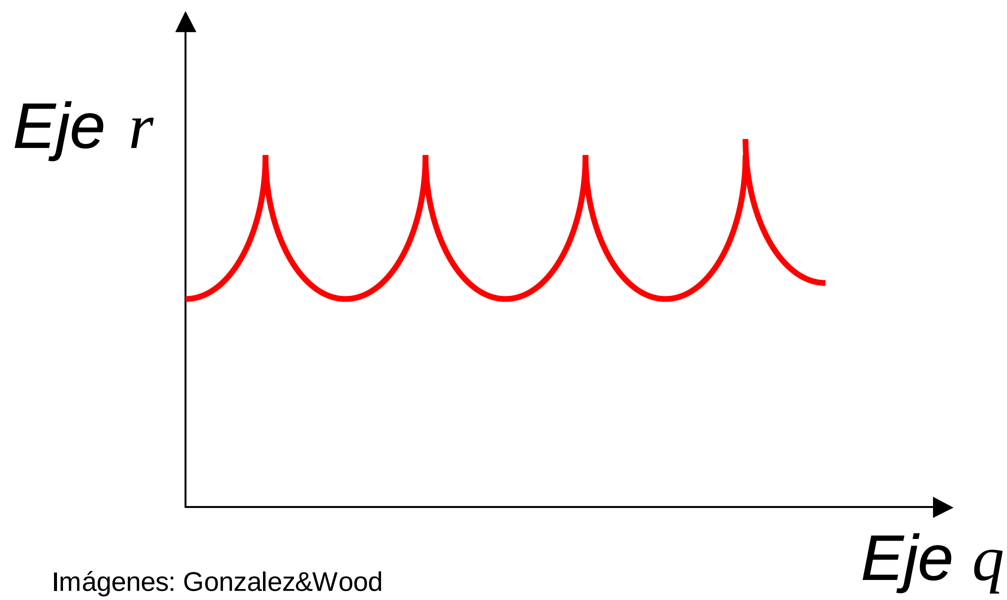
Una de las funciones de signatura más comunes es la distancia de los puntos del borde al centro del objeto



2. Signaturas

Las signaturas (firmas) de los objetos se deben normalizar en el eje y, para comparar objetos de igual forma pero distinto tamaño.



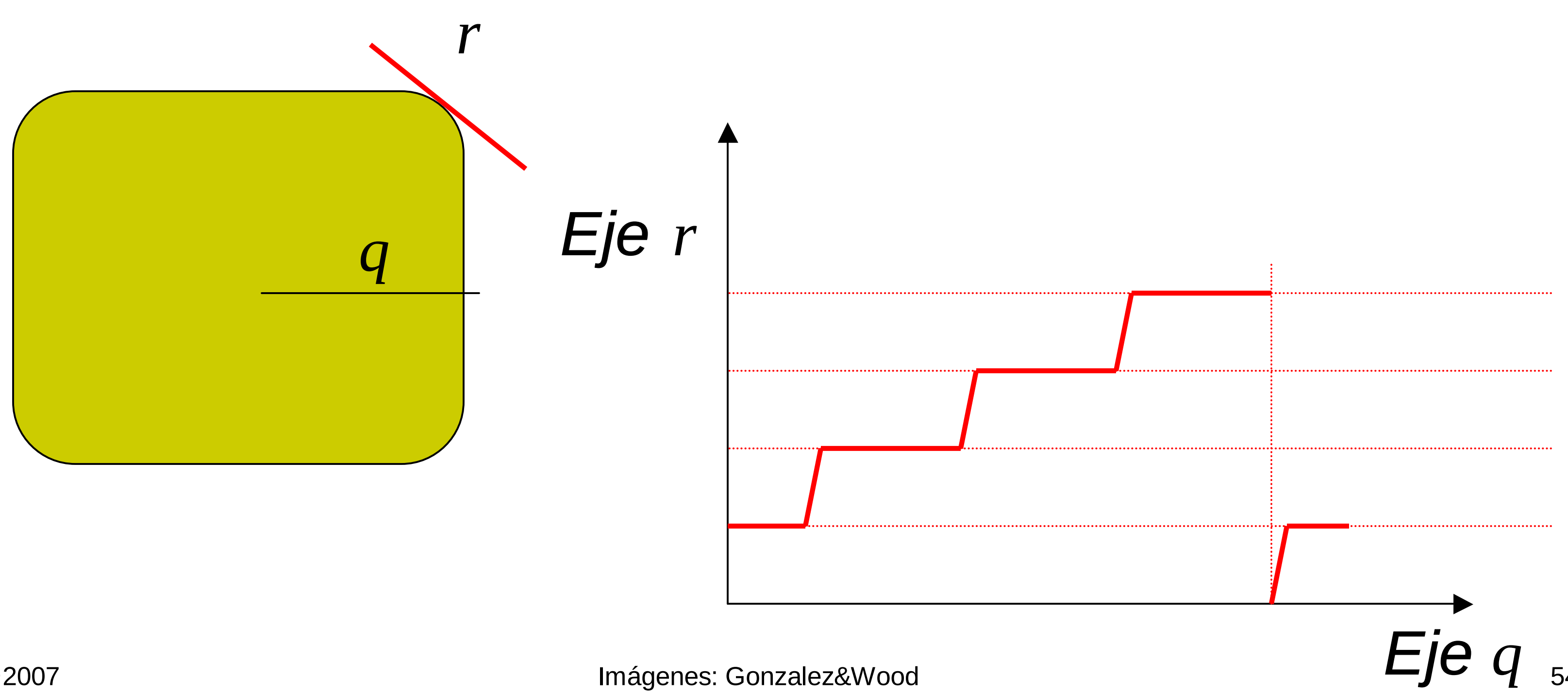


2. Signaturas

Existen otras funciones de signatura:

.- Desplazar una recta tangente en el borde

.- Histograma de los valores del ángulo tangente



3. Descriptores de Fourier

Los Descriptores de Fourier representan la forma del objeto.

Los primeros descriptores indican la forma general del objeto y los últimos descriptores los más pequeños detalles

Para una clasificación un pequeño conjunto de descriptores puede ser suficiente.

3. Descriptores de Fourier

La TF Unidimensional se aplica sobre una función continua que describe el contorno de una imagen.

Cada punto del borde x,y se considera un punto complejo p=x+jy.

De esta manera se aplica la transformada en una dimensión.

3. Descriptores de Fourier

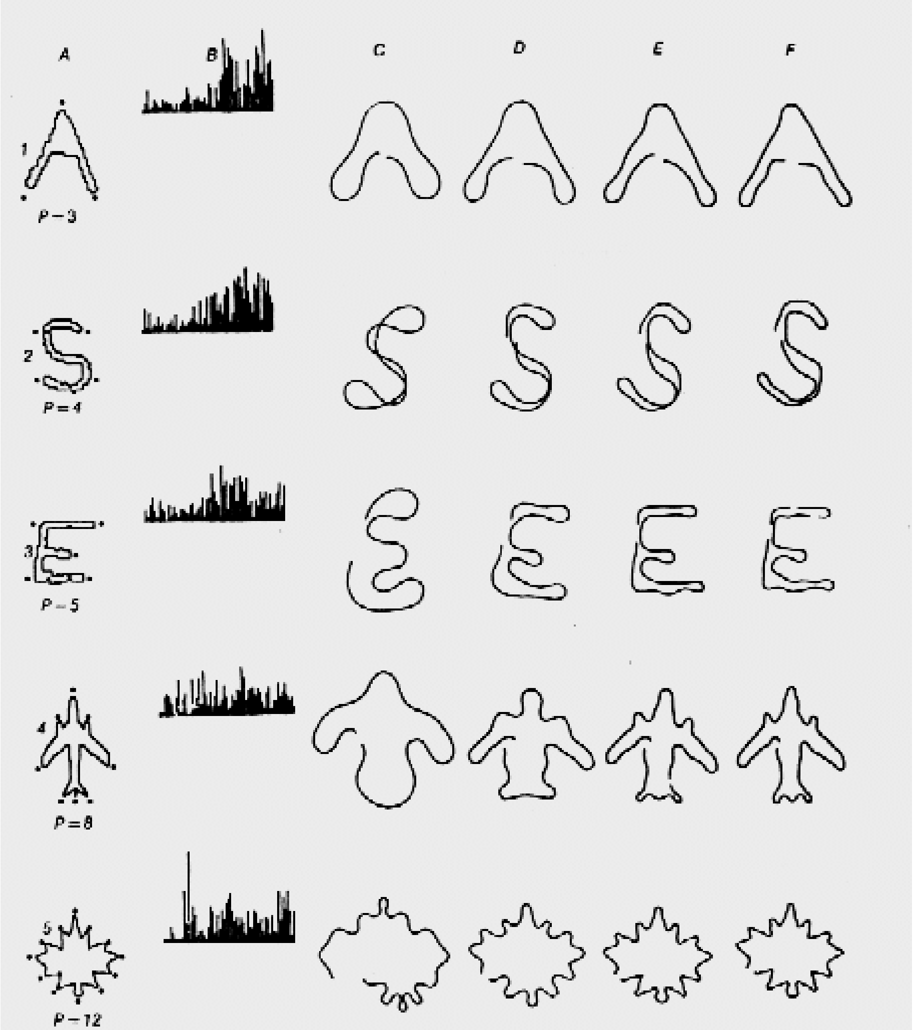


FIGURE 19.5-2. Examples of Fourier expansions of curvature function.

Con mas descriptores, mas aproximada será la imagen a a la original.

La compresión busca el máximo número de descriptores que se puedan eliminar de forma que se recupere la imagen original sin problemas.

Permite dar mayor estabilidad a la comparación de códigos de cadena.