



UNIVERSIDAD DE JAÉN

# Segmentación. Transformada de Hough.

Departamento de Ingeniería electrónica, Telecomunicación y Automática.  
Área de Ingeniería de Sistemas y Automática

## **OBJETIVOS:**

- Uso de la transformada de Hough como técnica de segmentación.

### **1. SEGMENTACIÓN DE LA IMAGEN.**

La segmentación es el proceso que divide una imagen en regiones u objetos cuyos píxeles poseen atributos similares. Cada región segmentada suele tener un significado físico dentro de la imagen. Es uno de los procesos más importantes de un sistema automatizado de visión ya que permite extraer los objetos de la imagen para su posterior descripción y reconocimiento.

Las técnicas de segmentación pueden encuadrarse en tres grupos fundamentales: técnicas basadas en la detección de la frontera, técnicas de umbralización y técnicas basadas en el agrupamiento de píxeles.

En las técnicas basadas en la frontera, la segmentación de una imagen puede llevarse a cabo mediante la detección de los límites de cada región, es decir, detectando los bordes de la imagen. Las otras dos técnicas (umbralización y técnicas basadas en el agrupamiento de píxeles), enfocan la segmentación como un problema de clasificación de píxeles o grupos de píxeles, donde: píxeles de una región deben ser similares, píxeles de regiones distintas deben ser no similares, las regiones resultantes deben tener cierto significado para el procesamiento posterior.

Veremos una técnica “La transformada de Hough”, perteneciente al primer bloque de las técnicas de segmentación (técnicas basadas en la frontera).

#### **1.1. La transformada de Hough.**

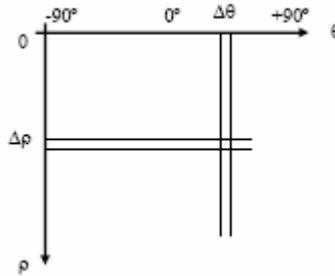
La transformada de Hough es una herramienta que permite detectar curvas en una imagen. Es una técnica muy robusta frente al ruido y a la existencia de huecos en la frontera del objeto. A la hora de aplicar la transformada de Hough a una imagen es necesario obtener primero una imagen binaria de los píxeles que forman parte de la frontera del objeto.

A continuación se explicará de forma muy simplificada la transformada de Hough para detectar rectas. El objetivo de la transformada de Hough es encontrar puntos alineados que puedan existir en la imagen, es decir, puntos en la imagen que satisfagan la ecuación de la recta, para distintos valores de  $\rho$  y  $\theta$ .

Ecuación de la recta en forma polar:  $\rho = x \cdot \cos\theta + y \cdot \sin\theta$ . Por tanto hay que realizar una transformación entre el plano imagen (coordenadas x-y) y el plano o espacio de parámetros ( $\rho, \theta$ ).

Para aplicar la transformada de Hough es necesario discretizar el espacio de parámetros en una serie de celdas denominadas celdas de acumulación. Esta discretización se realiza sobre los intervalos ( $\rho_{\min}, \rho_{\max}$ ) y ( $\theta_{\min}, \theta_{\max}$ ).

## Práctica 4ª.Segmentación. Transformada de Hough.



El siguiente paso es evaluar la ecuación de la recta para cada punto de la imagen  $(x_k, y_k)$ , si se cumple esta ecuación se incrementa en uno el número de votos de la celda. Un número de votos elevado indica que el punto pertenece a la recta.

En primer lugar utilizaremos la transformada de Hough para detectar rectas, trabajando con los siguientes scripts de Matlab: *hough.p*, *houghpeaks.p*, *houghlines.p*.

***hough.p*** → Este script realiza la transformada de Hough de una imagen, la función .m tendrá los siguientes parámetros de entrada y salida:

***function [h,theta,rho]=hough(f,dtheta,drho)***

f → imagen de bordes binaria.

dtheta → espaciado (en grados) de la transformada de Hough a lo largo del eje theta.

drho → espaciado de la transformada de Hough en el eje rho.

h → transformada de Hough.

theta → vector que contiene el ángulo (en grados) correspondiente a cada columna de h.

rho → vector que contiene los valores de rho correspondiente a cada fila de h.

Utilizaremos este script con el siguiente **ejemplo**:

1. Crear una imagen de tamaño 101x101, toda de ceros, exceptuando los siguientes píxeles que valen 1: (1,1), (101,1), (1,101), (51,51).
2. Calcula la transformada de Hough con el script *hough.p*.
3. Representar la transformada con *imshow*.
4. Repetir el apartado anterior pero con los siguientes comandos para representar la transformada:  
***[H, theta, rho] = hough(f);***  
***figure, imshow(theta, rho, H, [], 'notruesize');***  
***axis on;***  
***axis normal;***  
***xlabel('\theta')***  
***ylabel('\rho')***
5. Representar los picos más importantes de esta función, utilizando el script a continuación expuesto.

***houghpeaks.p*** → Detecta los picos que hay en la matriz H de la transformada de Hough. La función .m tendrá los siguientes parámetros de entrada y salida.

***function [r,c,hnew]=houghpeaks(h,numpeaks,threshold,nhood)***

h → matriz con la transformada de Hough.

numpeaks → máximo número de picos que debe buscar.

threshold → umbral a partir del cual una celda no será considerada como pico.

nhood → vector de dos elementos que especifica el tamaño para suprimir píxeles vecinos (debe ser positivo entero y par). El entorno de vecindad alrededor de cada pico identificado se pone a cero.

r,c → vectores con las filas y las columnas de las coordenadas de los picos identificados.

hnew → transformada de hough con los píxeles vecinos a un pico suprimidos.

***houghlines.p*** → Extrae segmentos de línea basándose en la transformada de Hough.

***function lines=houghlines(f,theta,rho,rr,cc,fillgap,minlength)***

Donde:

f → Imagen de bordes binaria a la que se le aplicará la transformada de Hough.

theta, rho → vectores que devuelve el script *hough.p*.

rr,cc → filas y columnas en los que buscar los segmentos de la transformada de Hough.

fillgap → Separación de píxeles a partir de la cual considera segmentos distintos (valor por defecto 20).

minlength → Número de píxeles mínimo que debe tener un segmento para ser considerado como tal (valor por defecto 40).

lines → estructura que devuelve esta función de longitud igual al número de segmentos de línea encontrados. Cada elemento de esta estructura tiene los siguientes campos.

point1 → punto inicial del segmento; vector de dos elementos.

point2 → punto final del segmento; vector de dos elementos.

length → distancia entre point 1 y point 2.

theta, rho → parámetros theta (en grados) y rho de la ecuación de la recta en forma polar en la que se encuentra contenido el segmento hallado.

Para probar estos scripts realizaremos el siguiente ejemplo (ejecutar todos los apartados juntos).

1. Calcular los bordes de la imagen *casa.jpg* mediante la función de Matlab ***edge***, utilizando el parámetro *canny* con límite [0.04 0.10] y sigma 1.5.
2. Realizar la transformada de Hough utilizando el script ***hough.p***, usando un espaciado angular de 0.5 grados. Representar dicho resultado de la forma especificada en el apartado 4 del ejemplo anterior.
3. Calcular los picos más importantes de la transformada de Hough anterior utilizando ***houghpeaks.p***.

Para representarlos utilizar el código:

***hold on;***

***plot(theta(c), rho(r), 'linestyle', 'none', 'marker', 's', 'color', 'w');***

4. Representar las cinco líneas seleccionadas en el apartado anterior mediante el script *houghlines.p*

```
figure, imshow(g_canny), hold on  
for k = 1:length(lines)  
    xy = [lines(k).point1 ; lines(k).point2];  
    plot(xy(:,2), xy(:,1), 'LineWidth', 4, 'Color', [.6 .6 .6]);  
end
```

*g\_canny* → es la imagen extraída del apartado 1 y *lines* es el resultado obtenido a través de *houghlines*.