

## PRÁCTICA 6

### "Segmentación de imágenes: aplicación de la Transformada de Hough"

**Objetivo:** Comprensión y aplicación de la Transformada de Hough para detectar líneas rectas y circunferencias en imágenes.

#### Instrucciones y Funciones Matlab:

- Instrucciones Matlab de utilidad: `bwlabel`, `find`, `cat`, `imfill`
- Instrucciones Matlab para la implementación de la Transformada de Hough para detectar líneas rectas: `hough`, `houghpeaks`, `houghlines`
- Funciones Matlab facilitadas para la implementación de la Transformada de Hough para detectar contornos circulares: `circle_hough`, `circle_houghpeaks`, `circlepoints`

#### PRIMERA PARTE: Segmentación de carreteras mediante la aplicación de la Transformada de Hough para detectar líneas rectas.

1. Lee la imagen `P6_1.tif` (incluida en la carpeta `ImagenesPractica/PrimeraParte`) y genera y visualiza su imagen de intensidad  $I$ . Utiliza para ello la función de Matlab `rgb2gray`.
2. A partir de  $I$ , genera una imagen binaria de bordes  $I_b$ . Para ello, aplica un detector de bordes verticales dado por la máscara correspondiente de Sobel; considera como umbral el 30% del valor máximo de la matriz magnitud del vector gradiente obtenida por el detector de bordes verticales. Describe los distintos pasos metodológicos que se pueden distinguir en el procedimiento de generación de  $I_b$ , explicando el significado de las matrices de información involucradas.
3. Aplica a  $I_b$  la Transformada de Hough para detectar líneas rectas utilizando la función de matlab `hough` de la siguiente forma:

```
[H,theta,rho] = hough(Ib, 'Theta', -90:89)
```

Utiliza la ayuda de matlab (`help hough`) para entender el funcionamiento de la función anterior y contesta:

- ¿Qué representan `theta` y `rho`?
- ¿Cuál es el significado de los valores almacenados en `H`?
- ¿Cómo es la discretización que se realiza del espacio de parámetros en esta configuración por defecto?
- Ayudándote de la función de Matlab `find`, escribe la ecuación de la recta que pasa por más puntos en la imagen binaria `Ib`.

4. Encuentra los parámetros representativos de las 5 rectas más votadas. Para ello aplica la función de matlab `houghpeaks` con la siguiente configuración:

```
NumRectas = 5; Umbral = ceil(0.3*max(H(:)));  
P = houghpeaks(H,NumRectas,'threshold', Umbral)
```

Utiliza la ayuda de matlab (`help houghpeaks`) para entender el funcionamiento de la función anterior y contesta:

- ¿Qué información contiene el parámetro de salida `P`?
- ¿Qué significado tiene la inclusión del parámetro de entrada `Umbral` en la función?
- ¿Qué efecto tiene en los resultados finales fijar `Umbral` con un valor `ceil(0.5*max(H(:)))`?

5. Muestra los segmentos de puntos de `Ib` que incluyen las 5 rectas detectadas. Para ello utiliza el código que se adjunta a continuación:

```
lines = houghlines(Ib,theta,rho,P,'FillGap',5,'MinLength',7);  
  
figure, imshow(Ibordes), hold on  
max_len = 0;  
for k = 1:length(lines)  
    xy = [lines(k).point1; lines(k).point2];  
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');  
  
    % Plot beginnings and ends of lines  
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');  
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');  
  
    % Determine the endpoints of the longest line segment  
    len = norm(lines(k).point1 - lines(k).point2);  
    if ( len > max_len)  
        max_len = len;  
        xy_long = xy;  
    end  
end
```

```
% highlight the longest line segment  
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red');
```

Utiliza la ayuda de matlab (help houghlines) para entender el funcionamiento de la función anterior y contesta:

- ¿Qué información contiene la variable `lines`?
- ¿Qué significado tienen las opciones elegidas en la llamada de la función `"'FillGap',5,'MinLength',7"`?

6. A partir de la información obtenida en los pasos anteriores, realiza la segmentación de la carretera siguiendo los siguientes pasos:

- Sobre una **imagen binaria inicializada a 1** (inicialmente "blanca") de las mismas dimensiones que la imagen de intensidad original, **asigna un valor 0 a todos los píxeles presentes en las rectas detectadas en P**. Visualiza la imagen binaria resultante.
- **Aplica un filtro de mínimos 3x3** a la imagen binaria anterior para unir los puntos de las líneas detectadas y delimitar regiones candidatas a ser la zona principal de la carretera. Visualiza la imagen binaria resultante.
- **Genera y visualiza la imagen binaria que representa la segmentación de la carretera asumiendo que es la región que contiene al píxel central de la imagen.**
- **Visualiza el resultado a través de una imagen que muestre únicamente los valores de intensidad de los píxeles detectados** (el resto de los píxeles se visualizarán en negro).

7. Aplica todos los pasos anteriores, con idéntica configuración, para segmentar las imágenes

`P6_2.tif` y `P6_3.tif`, incluidas en la carpeta facilitada:  
`ImagenesPractica/PrimeraParte`.

## SEGUNDA PARTE: Segmentación de señales de tráfico mediante la aplicación de la Transformada de Hough para detectar circunferencias.

### Segmentación de señales de limitación de velocidad:

8. A partir de la imagen en color `Signal1_1.tif` (incluida en la carpeta `ImagenesPractica/SegundaParte`), genera y visualiza una imagen binaria `Ib` de puntos "rojos" de la imagen en color, candidatos a pertenecer al contorno de la región que se pretende segmentar (interior de la señal de tráfico). Para ello, utiliza un método de umbralización global, tomando como umbrales de referencia para cada componente de color los siguientes:

```
factor = 0.35;
UmbralRojo = factor*(Rmin+Rmax);
UmbralVerde = factor*(Gmax+Gmin);
UmbralAzul = factor*(Bmax+Bmin);
```

donde `Rmin`, `Rmax`, `Gmin`, `Gmax`, `Bmin` y `Bmax`, son los valores de intensidad mínimo y máximo en las componentes roja (R), verde (G) y azul (B) de la imagen en color que está siendo analizada.

9. Aplica a `Ib` la Transformada de Hough para detectar contornos circulares utilizando la función facilitada `circle_hough` de la siguiente forma:

```
radii = 5:2:35; % Radios posibles de las circunferencias buscadas.
% Todas las imágenes tienen la misma resolución. Conocimiento a priori:
los objetos circulares buscados tienen sus radios limitados en ese
rango.

H = circle_hough(Ib, radii, 'same');

% Opción 'same': los centros de las circunferencias buscadas deben ser
puntos de la imagen, no se permiten circunferencias cuyos centros se
sitúan fuera de la imagen.
```

- Interpreta las dimensiones de la matriz de salida `H`. ¿Cuál es el significado de los valores almacenados en dicha matriz?
- Calcula “manualmente” el valor almacenado en `H(180,90,6)`. ¿Cómo interpretas el número obtenido? Para realizar el cálculo, utiliza la función facilitada `circlepoints`:

```
[x,y] = circlepoints(r); % genera las coordenadas (x,y) que pertenecen a  
la circunferencia de radio r y centro en el origen del sistema de  
coordenadas (0,0) .
```

- Ayudándote de la función de Matlab `find` , escribe la ecuación de la circunferencia que pasa por más puntos en la imagen binaria `Ib`, especificando su radio y las coordenadas `(x,y)` de su centro. ¿Cuántos puntos de `Ib` contiene esta circunferencia?

10. Encuentra los parámetros representativos de la circunferencia más votada aplicando la función facilitada `circle_houghpeaks` con la siguiente configuración:

```
P = circle_houghpeaks(H, radii, 'npeaks',1);
```

- ¿Qué información contiene el parámetro de salida `P`?
11. A partir de la información contenida en `P`, genera una imagen binaria, `Ib_circunf`, de las mismas dimensiones que las matrices que componen la imagen original que especifique los píxeles correspondientes a la circunferencia detectada. Para ello, utiliza la función facilitada `circlepoints`.

12. Genera y visualiza la imagen binaria que representa la segmentación de la señal de tráfico, `Ib_circulo`. Utiliza para ello la función de matlab `imfill`:

```
Ib_circulo = imfill(Ib_circunf, 'holes');
```

Visualiza el resultado a través de una imagen que muestre los píxeles detectados de la misma forma que aparecen en la imagen de color de entrada (el resto de los píxeles se visualizarán en negro).

13. Aplica todos los pasos anteriores, con idéntica configuración, para segmentar las señales de limitación de velocidad presentes en las imágenes `Signal1_2.tif` , `Signal1_3.tif` , `Signal1_4.tif` , `Signal2_1.tif` , `Signal2_2.tif` , `Signal3_1.tif` y `Signal3_2.tif`, facilitadas en la carpeta `ImágenesPractica/SegundaParte`.

### **Segmentación de señales de obligación (señales circulares de fondo azul):**

14. Genera y visualiza la imagen binaria que representa la segmentación de la señal de tráfico presente en las imágenes "`Signal4_1.tif` y `Signal4_2.tif`" facilitadas en la carpeta

ImágenesPractica/SegundaParte. Visualiza el resultado a través de una imagen que muestre los píxeles detectados de la misma forma que aparecen en la imagen de color de entrada (el resto de los píxeles se visualizarán en negro).