



Universidad Veracruzana

Maestría en Inteligencia Artificial

Visión por Computadora

**Examen 2. Segmentación y trnasformada de
Hough**

Ángel García Báez

Profesor: Dr. Héctor Acosta Mesa

June 19, 2025

Contents

1	Ejercicio practico 1	2
2	Ejercicio practico 2	4
3	Anexos	7
3.1	Implementación en matlab para la resolución de los ejercicios	
	1 y 2	7

1 Ejercicio practico 1

Se tiene la siguiente imagen y se quiere poder segmentar en sus 3 colores principales predominantes (los cuales se observan que son rojo, verde y blanco).



Imagen 1. Verduras con colores muy marcados.

Como propuesta para realizar dicha segmentación, se propone usar los pixeles de la imagen dentro del espacio de colores RGB y usar esos datos para realizar un proceso de agrupación por K-medias, donde de antemano se quieren obtener 3 grupos, por lo que el K es igual a 3. Esto se propone así por su sencillez de implementación y porque los colores de la imagen se muestran muy bien delimitados, es decir, no es muy compleja la composición de la imagen.

A continuación se muestra el resultado de aplicar este proceso con $k = 3$ y coloreando los pixeles de la imagen con el color del centroide del grupo al que corresponden.

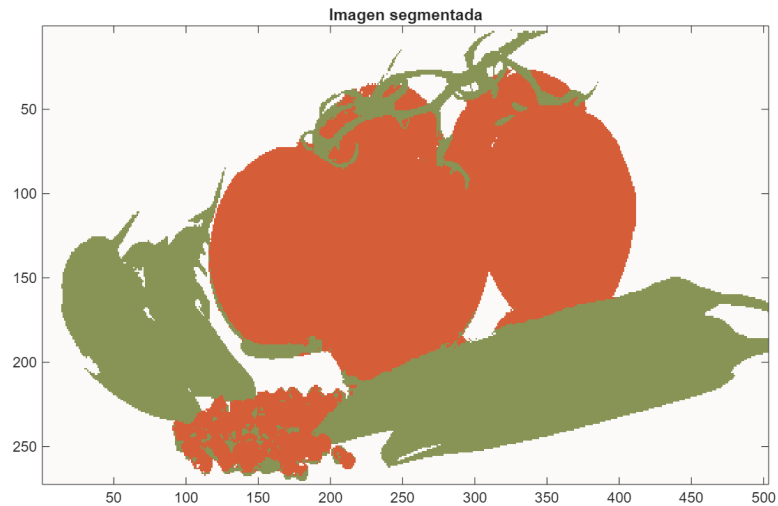


Imagen 1. Verduras con colores muy marcados.

Se observa que el resultado de la segmentación cumple su propósito, logra diferenciar bien los chiles y los tallos de los tomates, diferencia bien el fondo de las verduras pero con algunos pequeños errores que convierte el fondo como si fuera de color verde.

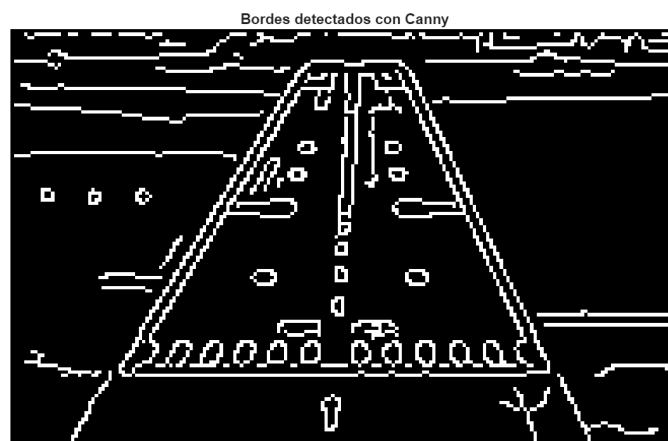
2 Ejercicio practico 2

Se tiene la siguiente imagen de una pista de aterrizaje, de la cual se quiere extraer los bordes que delimitan a la misma mediante la transformada de Hough.



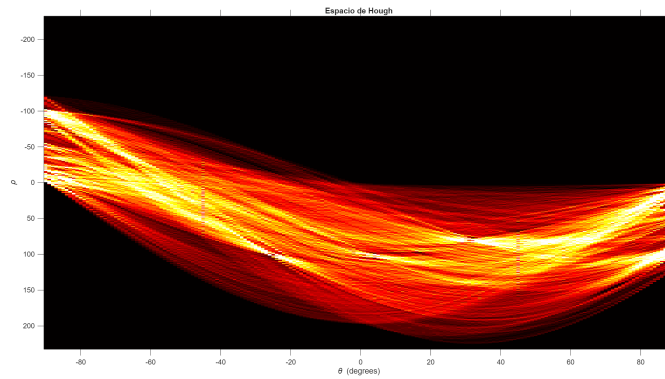
Imagen 2. Pista de aterrizaje.

Para llevar a cabo este proceso, primero fue necesario convertir la imagen en escala de blanco y negro para hacer la detección de bordes con el filtro de Canny.

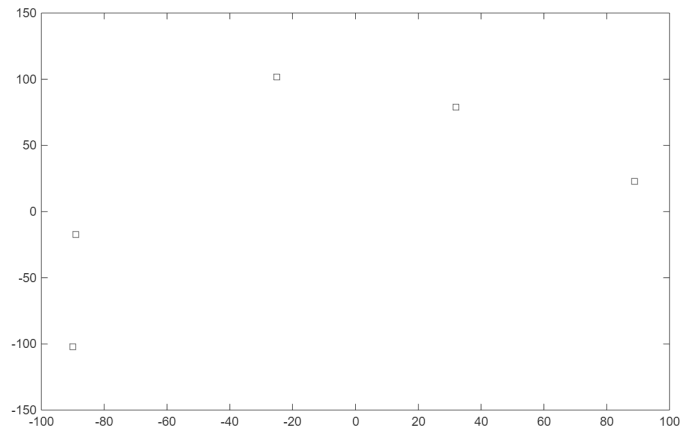


Pista con bordes detectados.

Posteriormente, se hicieron las estimaciones de los parametros de la ecuación del modelo de Hough para mandar los pixeles al espacio de Hough y detectar los picos más altos:



Espacio de Hough.



Picos detectados.

Finalmente, con los picos detectados, se proyectan sobre la imagen original para mostrar las formas encontradas que delimitan a la pista de aterrizaje.



Pista con los margenes proyectados.

3 Anexos

3.1 Implementación en matlab para la resolución de los ejercicios 1 y 2

```
1
2  %% Cargar la imagen de las verduras %%
3  I = imread("imagen1.jpg");
4  imshow(I)
5
6  %% Pedir un kmeans con k = 3 %%
7  [seg,centros] = imsegkmeans(I,3);
8  imagesc(seg);
9  title("Imagen segmentada")
10 % Agregar los colores de los centros como el colormap %
11 colormap(centros);
12
13
14 %% Cargar la imagen para el segundo ejercicio %%
15 I = imread("imagen2.jpg");
16 imshow(I)
17
18 %% Convertir la imagen en escala de grises %%
19 BI = rgb2gray(I);
20
21 %% Aplicar la transformada con canny y volver a proyectar
22 BW = edge(BI,'canny');
23 imshow(BW);
24 title('Bordes detectados con Canny');
25
26 %% 4. Realizar la Transformada de Hough y proyectar su espacio %%
27 [H, T, R] = hough(BW);
28 imshow(imadjust(rescale(H)), 'XData', T, 'YData', R, ...
29 'InitialMagnification', 'fit');
30 title('Espacio de Hough');
31 xlabel('\theta (degrees)');
32 ylabel('\rho');
33 axis on, axis normal, hold on;
34 colormap(gca, hot);
35
36
```



```

37
38 %% 5. Encontrar los picos en el espacio de Hough
39 P = houghpeaks(H, 5, 'threshold', ceil(0.6 * max(H(:)))));
40 x = T(P(:,2));
41 y = R(P(:,1));
42 plot(x, y, 's', 'Color', 'black');
43
44 %% 6. Extraer las líneas de los picos encontrados
45 lines = houghlines(BW, T, R, P, 'FillGap', 20, 'MinLength', 40);
46 imshow(I); % Muestra la imagen original
47 title('Líneas de la pista de aterrizaje detectadas');
48 hold on; % Permite dibujar sobre la imagen
49 max_len = 0; % Para encontrar la línea más larga, opcional
50 for k = 1:length(lines)
51 xy = [lines(k).point1; lines(k).point2];
52 plot(xy(:,1), xy(:,2), 'LineWidth', 2, 'Color', 'red'); % Dibuja la línea en rojo
53 % Marca los puntos inicial y final de la línea (opcional)
54 plot(xy(1,1), xy(1,2), 'x', 'LineWidth', 2, 'Color', 'yellow');
55 plot(xy(2,1), xy(2,2), 'x', 'LineWidth', 2, 'Color', 'green');
56 % Determina la línea más larga (opcional)
57 len = norm(lines(k).point1 - lines(k).point2);
58 if ( len > max_len)
59 max_len = len;
60 longest_line = xy;
61 end
62 end
63 if exist('longest_line', 'var')
64 plot(longest_line(:,1), longest_line(:,2), 'LineWidth', 2, 'Color', 'blue');
65 end
66 hold off;
67

```