



TRABAJO PRÁCTICO N° 3 - Año 2025 - 1° Semestre

Problema 1 - Detección de carril

Los videos ruta_1.mp4 y ruta_2.mp4 fueron grabados con una cámara colocada en una posición fija en el interior de un auto, apuntando al frente, mientras el auto circulaba por una ruta. En la Figura 1 se muestra un frame de dichos videos. Se debe realizar lo siguiente:

- Desarrollar un algoritmo para detectar automáticamente el carril por el cual está circulando el auto, mediante la detección de las dos líneas que lo delimitan. En la Figura 2 se muestra un ejemplo del resultado esperado.
- Generar videos donde se muestren las líneas que definen el carril resaltadas en color azul.



Figura 1 – Frame del video ruta_1.mp4.

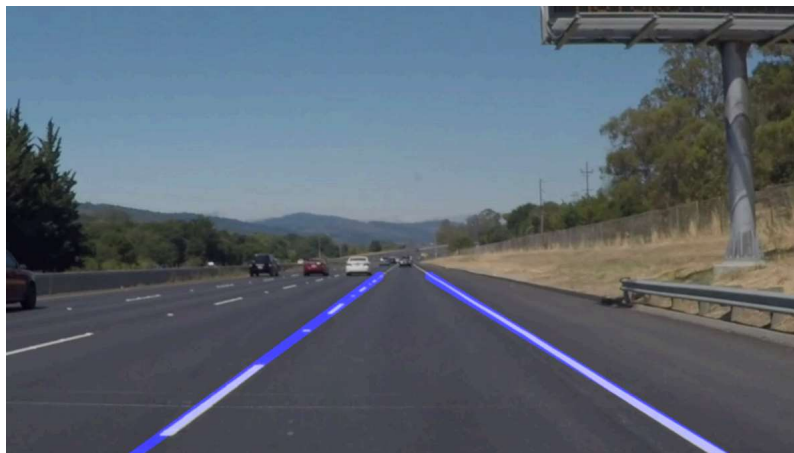


Figura 2 – Ejemplo del resultado esperado.



AYUDA:

1) Para detectar líneas, puede utilizar la función `cv2.HoughLinesP(img, rho, theta, threshold, lines, minLineLength, maxLineGap)`. Los parámetros `img`, `rho`, `theta` y `threshold` tienen el mismo significado que para `cv2.HoughLines()`, el parámetro `lines` no lo utilizaremos (le asignamos el valor `np.array([])`), `minLineLength` representa la longitud mínima de línea permitida y `maxLineGap` representa la distancia máxima permitida entre puntos de una misma línea para enlazarlos. A diferencia de `cv2.HoughLines()`, cada línea encontrada se representa mediante un vector de 4 elementos (x_1, y_1, x_2, y_2) , donde (x_1, y_1) e (x_2, y_2) son los puntos finales de cada segmento de línea detectado.

2) Se puede crea una máscara para definir una ROI de forma poligonal utilizando `cv2.fillPoly(img, points, color)`, donde `img` es la imagen donde se dibujará el polígono (por ejemplo `np.zeros((H,W),dtype=np.uint8)`), `points` son los vértices que representan al polígono y `color` es el color con el cual se rellena el polígono (por ejemplo 255).