

ELE4001 Robótica - Otoño 2024

Tarea N°2: Aprendiendo a detectar líneas y eliminar distorsión

Profesores: Ignacio Bugueño, Alfonso Ehijo

Ayudante de cátedra: Ariel Zuñiga

Fecha de entrega: Jueves 9 de Mayo

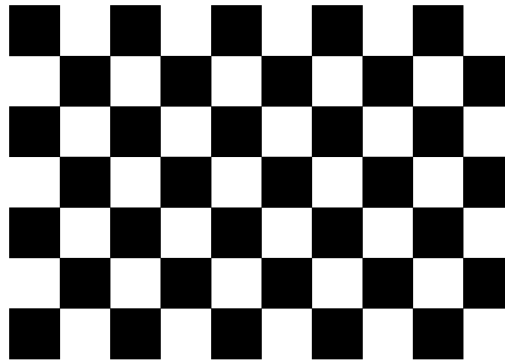
P1. Detección de líneas usando Transformada de Hough

- (a) Implemente un sistema de detección de bordes. Se recomienda usar el detector de Canny incluido en OpenCV. Dicho detector depende de dos umbrales de binarización, que suelen mantenerse en una proporción 1:3. También puede usar otros detectores vistos en clases, tales como detector de Sobel, entre otros.
- (b) Implemente un sistema de detección de líneas rectas usando la Transformada de Hough para verificar la pertenencia de los candidatos encontrados en (a).

Importante: La cuantización de los parámetros es muy relevante para el adecuado funcionamiento de la transformada. Investigue estos temas en libros o en Internet antes de hacer su implementación final.

- (c) Visualizar las líneas obtenidos por Hough sobre la imagen original.

P2. Calibración de Cámara utilizando Tablero de Ajedrez



- (a) Primero, defina las coordenadas del mundo real de los puntos 3D utilizando un tamaño conocido de patrón de tablero de ajedrez (utilice el patrón de 9x6 de la figura adjunta).
- (b) Luego, cargue las imágenes provistas en U-Campus con diferentes puntos de vista del tablero de control.
- (c) Sobre las diferentes imágenes, utilice alguno de los métodos disponibles en OpenCV para encontrar las coordenadas de píxeles (u, v) para cada punto 3D.
- (d) Finalmente, utilice alguno de los métodos disponibles en OpenCV para encontrar los parámetros intrínsecos y extrínsecos de la cámara. Reporte los resultados asociados a la Camera Matrix, coeficiente de distorsión, vector de rotación y vector de traslación.

Recuerde que Camera Matrix ayuda a transformar puntos de objetos 3D en puntos de imagen 2D y el coeficiente de distorsión devuelve la posición de la cámara en el mundo, con los valores de los vectores de rotación y traslación.

P3. ¡Detectando las vías de la ciudad!

En Gym-Duckietown, deben generar un detector de líneas de carril (deben basarse en su totalidad en técnicas de procesamiento de imágenes).

A modo general, a medida que el duckiebot recorre la ciudad de Gym-Duckietown, este debe ser capaz de detectar las líneas correspondientes al carril, es decir, debe detectar las líneas amarillas que representan la separación de carriles y las líneas blancas que representan la separación entre vías y arcén.

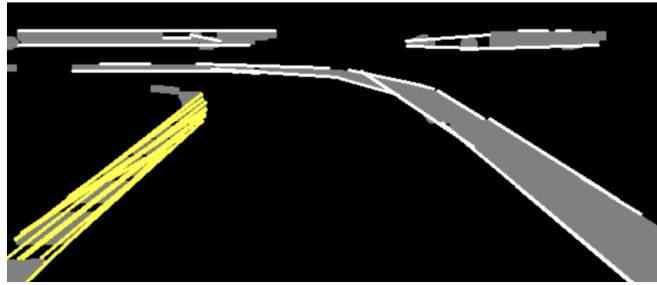


Figure 1: Detección de líneas de vías en Gym-Duckietown.

Se recomienda:

- Filtrar los colores de interés usando un espacio de color arbitrario.
- Sobre la imagen binarizada, aplicar las operaciones morfológicas respectivas (erosión y dilatación).
- Detectar los bordes de la imagen (usando algún método de detección de su preferencia).
- Detectar las líneas de las vías usando los bordes de la imagen.
- Dibujar las vías resultantes sobre la imagen original.

Consideraciones

Para la realización de esta tarea, usted debe usar las imágenes provistas en Material Docente (para la P1 y P2) y capturar imágenes representativas del simulador Gym-DuckieTown (para la P3). Por otro lado, debe enfocarse en programar los algoritmos pedidos y hacer las pruebas solicitadas.

El informe debe contener como mínimo: resumen, introducción, marco teórico, metodología (descripción de los algoritmos, partes relevantes del código), resultados (mostrando las imágenes resultantes), análisis/discusión de los resultados, y conclusiones generales. Cada elemento en el enunciado debe ser abordado en el informe. La cantidad máxima de páginas es 10, puede crear una sección Anexos para reportar resultados y códigos de menor relevancia.

Los informes y los códigos deben ser subidos a la Plataforma U-Campus hasta el día Jueves 9 de Mayo a las 23:59 hrs. Cada día de retraso (incluyendo sábados y domingos) será penalizado con 10 décimas de descuento en la nota.