

CorridorTracking

Arquitectura de altas prestaciones para visión

Máster en Ciencia de Datos e Ingeniería de Computadores

Curso 2020 - 2021

Álvaro Ayuso Martínez

Índice

- ▶ Contexto
- ▶ Estado del arte: Seguimiento de línea y “pasillo”
- ▶ Implementación: Idea inicial
- ▶ Estado del arte: Segmentación de la imagen
- ▶ Implementación: Idea final
 - ▶ Extracción de suelo
 - ▶ Cálculo de la dirección
- ▶ Resultados
- ▶ Demostración
- ▶ Trabajo futuro

Contexto



Imagen simple
¿Sencillo?



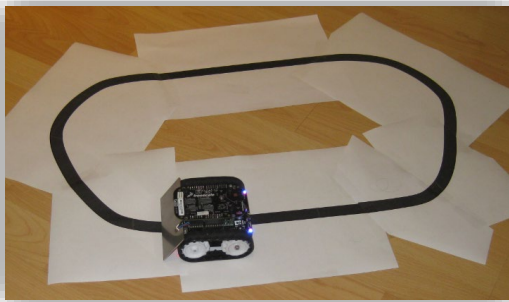
Imagen compleja
Complicado

- ▶ Tracking en robots y vehículos móviles
- ▶ TFM: Dji Tello Ryze. Posicionamiento por visión

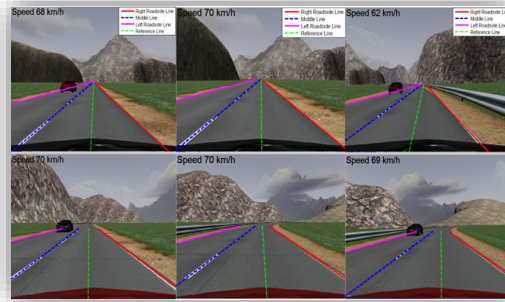


Estado del arte

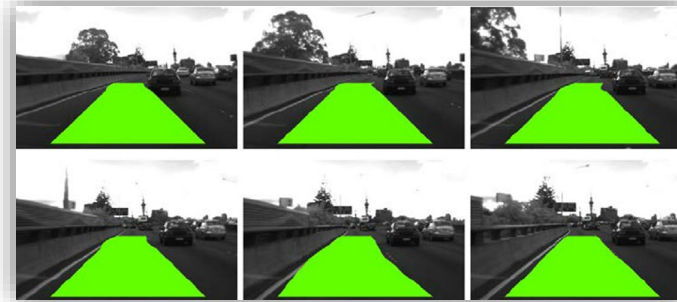
Seguimiento de línea y “pasillo” (corridor)



Zumo



Lane detection & tracking [1]



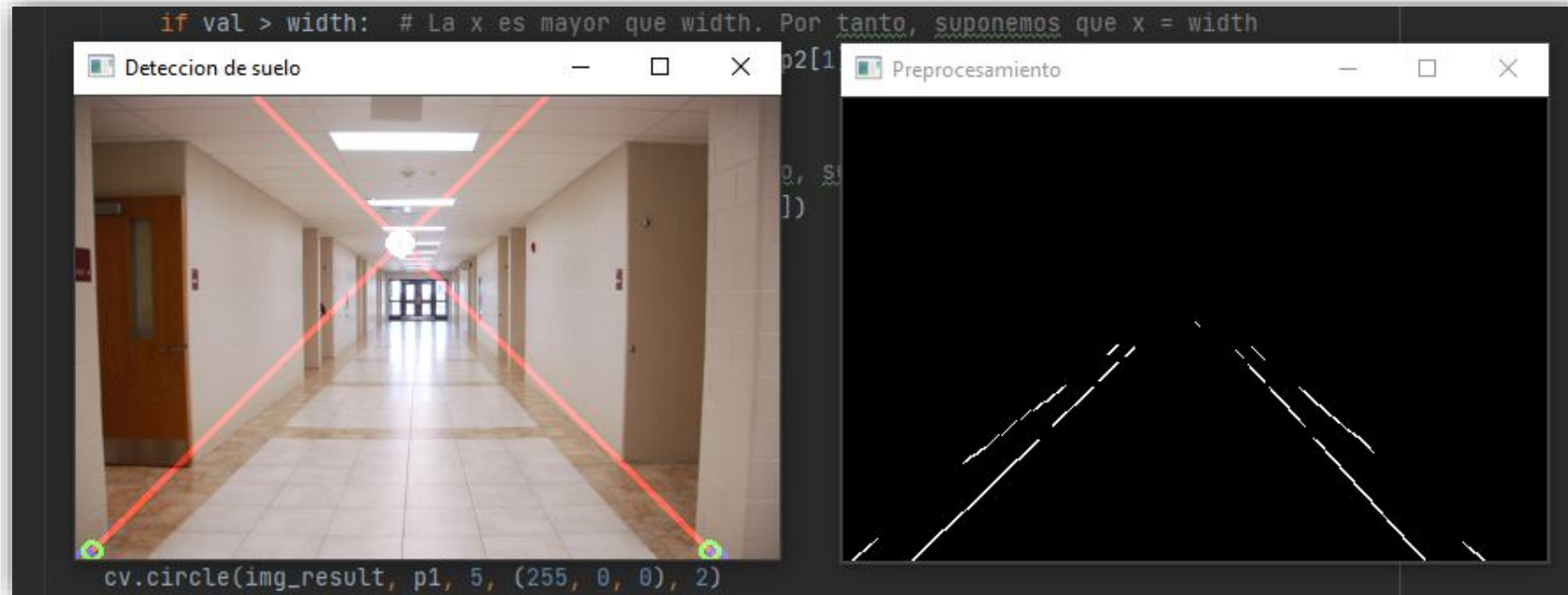
Corridor detection & tracking [2]

[1] F. Bounini, D. Gingras, V. Lapointe and H. Pollart, "Autonomous Vehicle and Real Time Road Lanes Detection and Tracking," 2015 IEEE Vehicle Power and Propulsion Conference (VPPC), 2015, pp. 1-6, doi: 10.1109/VPPC.2015.7352903.

[2] Jiang, Ruyi & Klette, Reinhard & Vaudrey, Tobi & Wang, Shigang. (2011). Corridor Detection and Tracking for Vision-Based Driver Assistance System.. IJPRAI. 25. 253-272. 10.1142/S0218001411008567.

Implementación

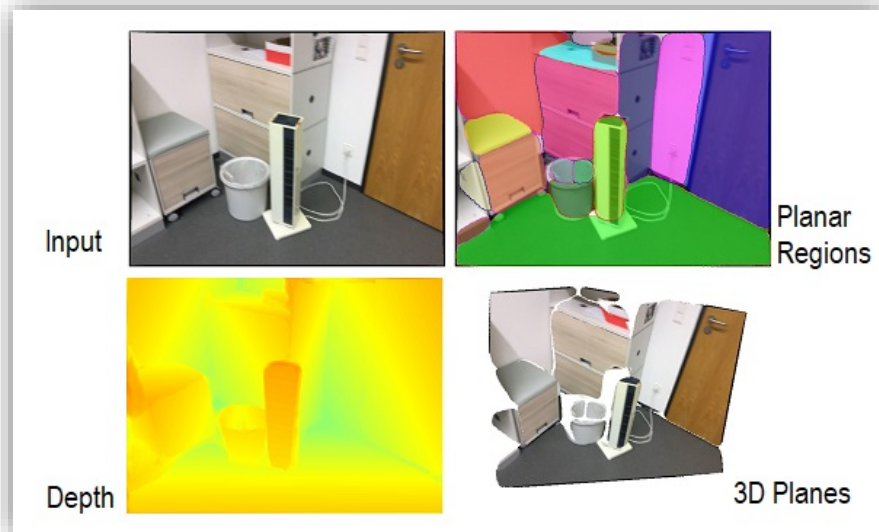
Idea inicial



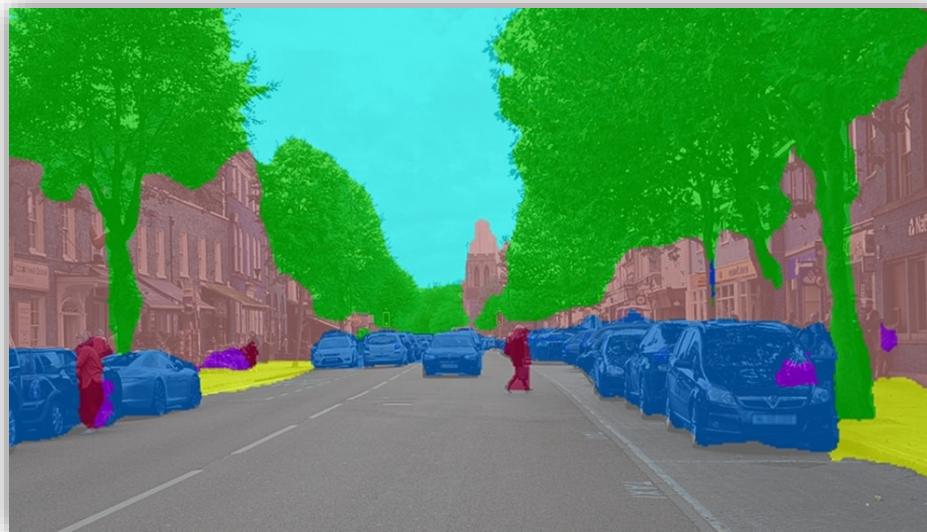
- ▶ Extensión de líneas. Cálculo de la intersección
- ▶ Problema: Resultado pobre y poco fiable
- ▶ Solución: Segmentación semántica

Estado del arte

Segmentación de la imagen



PlaneRCNN [3]



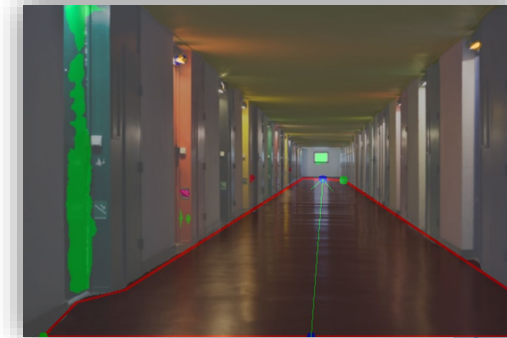
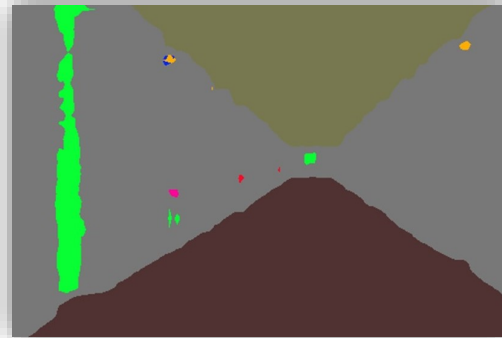
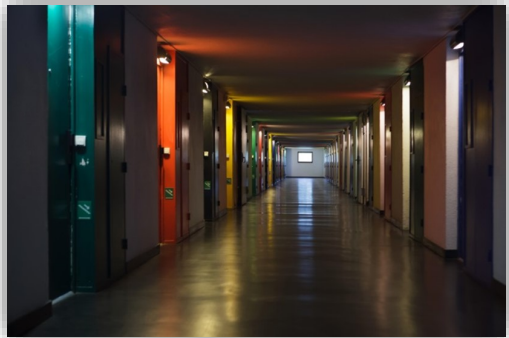
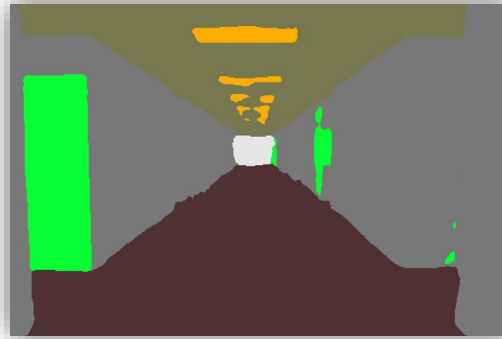
PixelLib (Ade20k) [4]

[3] <https://github.com/NVlabs/planercnn>

[4] https://pixellib.readthedocs.io/en/latest/image_ade20k.html

Implementación

Idea final



- ▶ **Ventaja:** Algoritmo mucho más robusto
- ▶ **Desventaja:** Gran incremento del coste computacional (segmentación)

Implementación

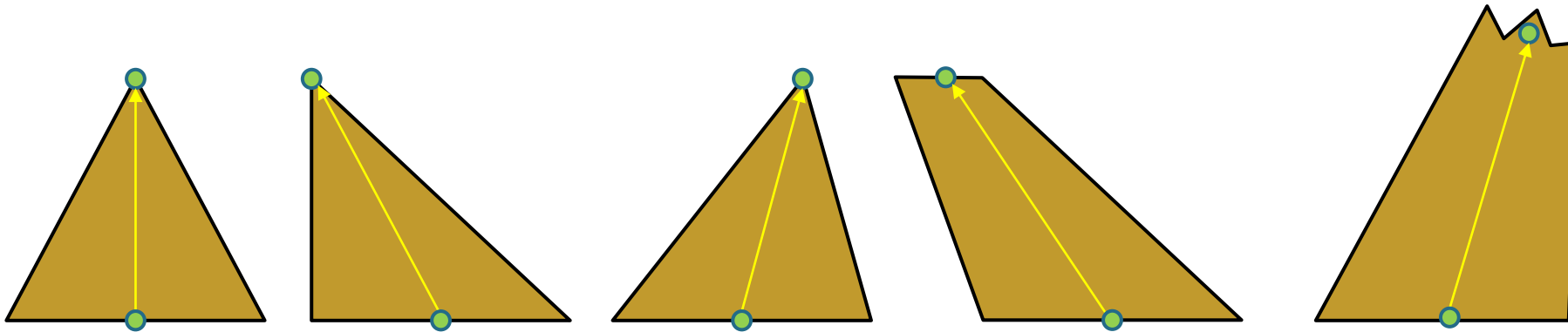
Idea final - Extracción del suelo



- ▶ Extracción por color <- Rango de colores: (45, 45, 75) hasta (55, 55, 85)
- ▶ Operaciones morfológicas de cierre y apertura para pulido
- ▶ Paso siguiente: Búsqueda de contornos y aproximación poligonal

Implementación

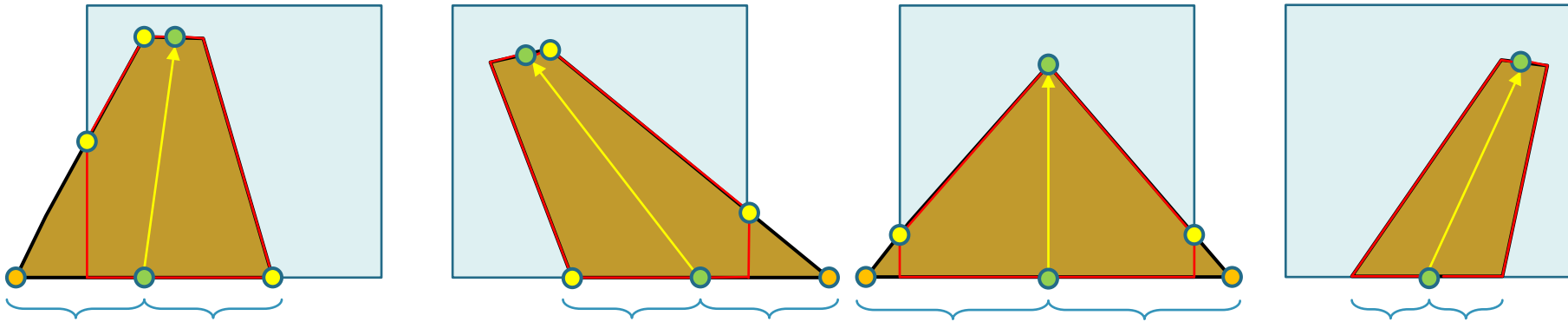
Idea final - Cálculo de la dirección



- ▶ Búsqueda inicial de puntos extremos (coordenada Y)
- ▶ Punto superior:
 - ▶ Pendientes pronunciadas -> Punto extremo superior
 - ▶ Defectos de la aproximación poligonal -> Punto medio (conjunto de puntos)
 - ▶ Otro caso: Comparación entre puntos medios (**segmento anterior y segmento siguiente**. Menor coordenada Y)

Implementación

Idea final - Cálculo de la dirección



- **Punto inferior:**

- Casuística según 4 diferentes escenarios
- Caso sin bordes laterales:
 - Proceso similar al cálculo del punto superior
 - Menos sofisticado (errores poco frecuentes)

- **Dirección final:** Media de las direcciones en n_{mem} frames anteriores.

Resultados

Segmentación sin optimizaciones (CPU)

Sistema	
Procesador	AMD Ryzen 5 3600 (3.6 GHz)
Memoria RAM	16384 MB
Gráfica	AMD Radeon RX 5600 XT

1) Segmentación en imágenes

Número de imágenes	Total (s)	Media por imagen (s)	FPS (procesamiento)
10	21.80	2.18	0.46

2) Segmentación + Post-procesamiento en cámara

Nº prueba	Primer plano (FPS)	Segundo plano (FPS)
1	0.46 – 0.47	0.44 – 0.45
2	0.53 – 0.54	0.50 – 0.53

Resultados

Segmentación sin optimizaciones (CPU)

3) Segmentación en video

Nombre	Total (s)	Media por frame (s)	FPS (procesamiento)
corridor1	786.69	1.81	0.55
corridor2	1309.30	2.05	0.49
corridor3	645.86	2.10	0.48
corridor4	1066.01	2.41	0.41
corridor5_Trim	1013.60	2.01	0.50

Nombre	Resolución	Duración (s)	FPS	Frames totales
corridor1	640x352	14.46	30.01	434
corridor2	1920x1080	12.80	50.00	640
corridor3	1920x1080	12.32	25.00	308
corridor4	3840x2160	17.68	25.00	442
corridor5_Trim	640x360	16.80	30.00	504

Resultados

Segmentación con optimizaciones (GPU)

Sistema	
Procesador	Intel Core i7-1075H (2.6 GHz)
Memoria RAM	32768 MB
Gráfica	NVIDIA GeForce RTX 2060

1) Segmentación en imágenes

Número de imágenes	Total (s)	Media por imagen (s)	FPS (procesamiento)
10	9.92	0.99	1.01

2) Segmentación + Post-procesamiento en cámara

Nº prueba	Primer plano (FPS)	Segundo plano (FPS)
1	2.92 – 3.02	2.74 – 2.95
2	2.93 – 3.02	2.74 – 2.99

Resultados

Segmentación con optimizaciones (GPU)

3) Segmentación en video

Nombre	Total (s)	Media por frame (s)	FPS (procesamiento)
corridor1	141.1	0.33	3.03
corridor2	239.2	0.37	2.70
corridor3	118.3	0.38	2.63
corridor4	236.7	0.54	1.85
corridor5_Trim	163.8	0.33	3.03

Nombre	Resolución	Duración (s)	FPS	Frames totales
corridor1	640x352	14.46	30.01	434
corridor2	1920x1080	12.80	50.00	640
corridor3	1920x1080	12.32	25.00	308
corridor4	3840x2160	17.68	25.00	442
corridor5_Trim	640x360	16.80	30.00	504

Resultados

Comparación

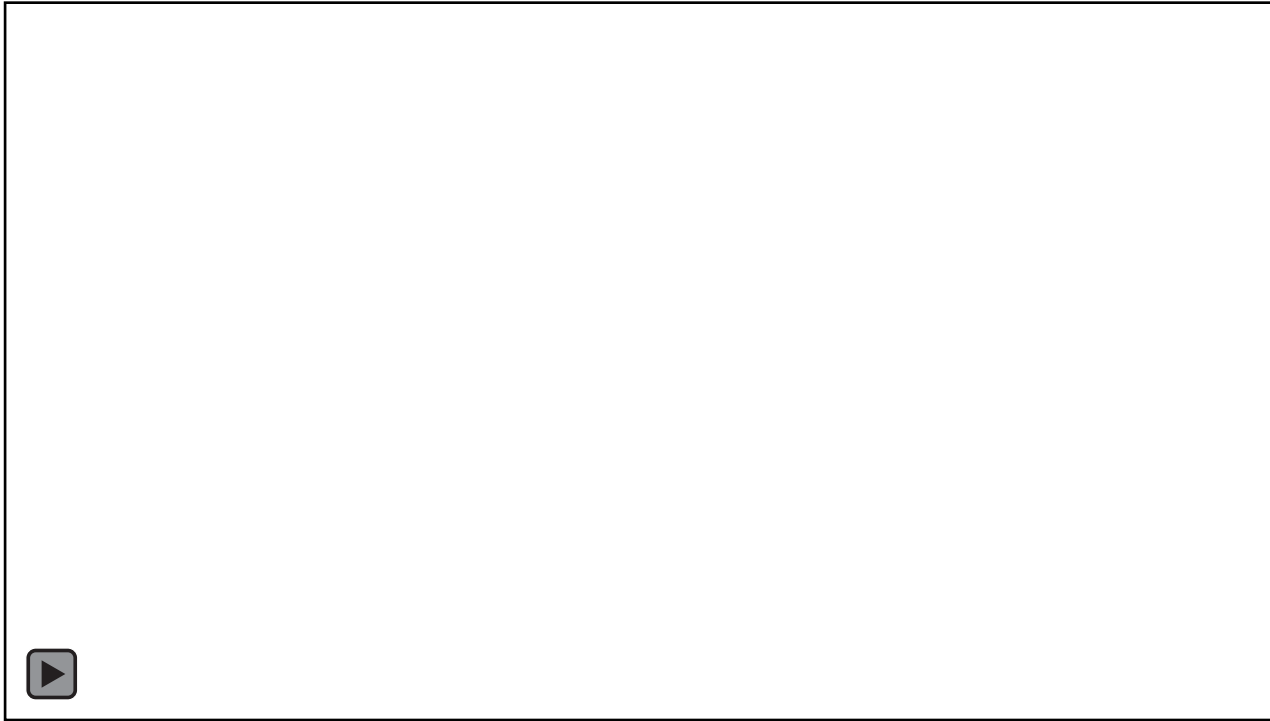
Segmentación en video

Nombre	FPS (CPU)	FPS (GPU)	Incremento
corridor1	0.55	3.03	~ 451%
corridor2	0.49	2.70	~ 451%
corridor3	0.48	2.63	~ 448%
corridor4	0.41	1.85	~ 351%
corridor5_Trim	0.50	3.03	~ 506%

Segmentación y post-procesamiento en cámara

Prueba	Media FPS (CPU)	Media FPS (GPU)	Incremento
1	0.46	2.90	~ 530%
2	0.53	2.91	~ 449%

Demostración



Trabajo futuro

- ▶ Corrección de errores y mejoras
- ▶ Optimización del código
- ▶ Implementación de memoria en extracción de suelo
- ▶ Búsqueda de modelos de segmentación menos pesados
 - ▶ Decremento del coste computacional
 - ▶ Pruebas en sistemas de baja memoria RAM (Jetson Nano)

¿Alguna pregunta?