Comportamientos para vehículos autónomos y la categoría AutoModelCar

Instructor: Dr. Marco Antonio Negrete Villanueva

Facultad de Ingeniería, UNAM

Escuela de Invierno de Robótica 2024
Nanchital, Veracruz
https://github.com/mnegretev/EIR-2024-AutonomousVehicles

Objetivos:

Objetivo General: Revisar conceptos básicos para operar un vehículo sin conductor en el contexto de las pruebas de la categoría AutoModelCar

Objetivos Específicos:

- Revisar el hardware necesario para tener un vehículo sin conductor
- Dar un panorama general del software necesario para desarrollar un vehículo sin conductor
- Revisar las herramientas disponibles para implementar los comportamientos básicos en un vehículo sin conductor
    - Detección de carriles
    - Seguimiento de carriles
    - Rebase
    - Detección de señales de tránsito

Contenido

Motivación

¿Por qué vehículos sin conductor?

- La mayor cantidad de accidentes tienen como causa un error humano
- Se podría disminuir este número con vehículos sin conductor
- Menores tiempos de traslado mediante planeación de rutas óptimas y conducción sin distracciones
- Movilidad urbana más accesible
- Disminución de emisiones debido a una conducción óptima

¿Cuándo un vehículos se considera autónomo?

La SAE propone seis niveles de autonomía:

Nivel 0 (No driving automation): El conductor es responsable de todas las tareas de conducción.

Nivel 1 (Driving assistance): El sistema de conducción automatizada toma control en situaciones muy específicas. Ejemplo: mantener cierta velocidad en carretera o frenar en automático ante un obstáculo.

Niveles de autonomía SAE

Nivel 2 (Partial driving automation): El auto puede realizar la mayor parte de tareas de conducción pero el conductor debe mantenerse atento durante toda la conducción. Ejemplo: Tesla.

Niveles de autonomía SAE

Nivel 3 (Conditional driving automation): El conductor puede desconectarse de la tarea de conducción bajo ciertos climas y carreteras. Todavía se requiere que el conductor resuelva ciertas situaciones de manejo. Ejemplo: Mercedes-Benz

Niveles de autonomía SAE

Nivel 4 (High driving automation): El vehículo es capaz de ejecutar funciones tácticas y operativas y es capaz de asegurarse automáticamente si los límites operativos se superan. El vehículo no es completamente autónomo pero el conductor no necesita responder ante fallas operativas. Ejemplo: **El TMR, pronto (esperemos)**

Nivel 5 (Full driving automation): No se requiere intervención humana y el sistema puede conducir sin restricciones de clima, carretera, horario o condición geográfica. Ejemplo: **El TMR, pronto (esperemos)**

La categoría AutoModelCar

- Se originó por el proyecto *Visiones de movilidad urbana* a través del cual se donaron vehículos a escala a varias instituciones educativas y de investigación del país

La categoría AutoModelCar

- Originalmente solo se permitían vehículos AutoNOMOS:

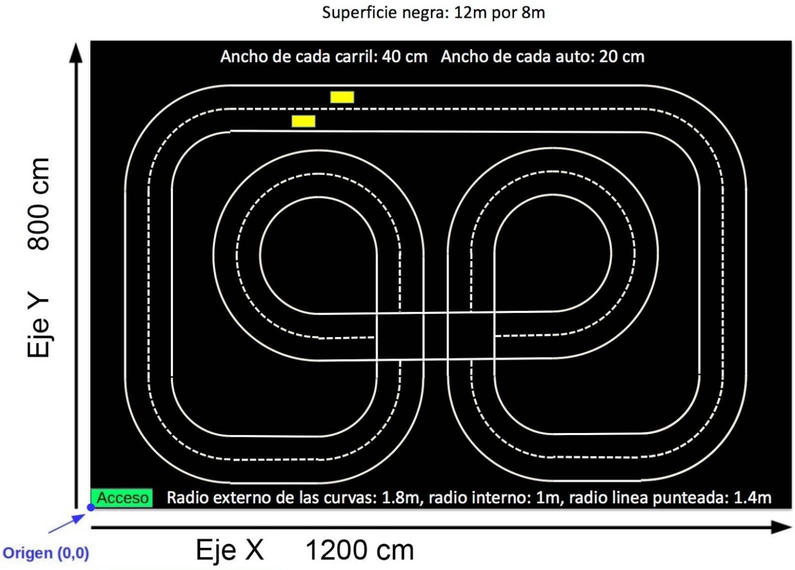
- Pero ahora se puede participar con cualquier vehículo a escala:

La categoría AutoModelCar

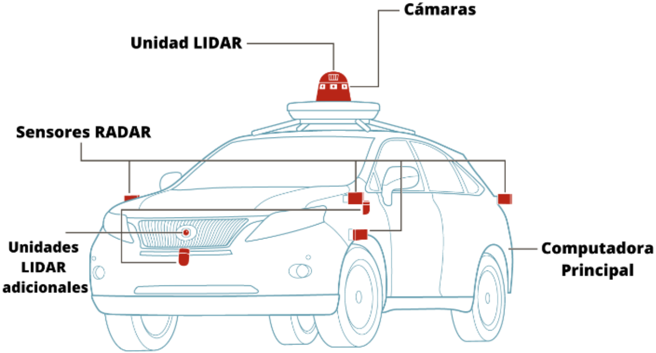
- La competencia consta de cuatro pruebas (esperemos aumentarlas este año):
    - Navegación autónoma sin obstáculos
    - Navegación con obstáculos estáticos
    - Navegación con obstáculos en movimiento
    - Estacionamiento

La categoría AutoModelCar

Se utiliza una pista con rectas, curvas y cruces:



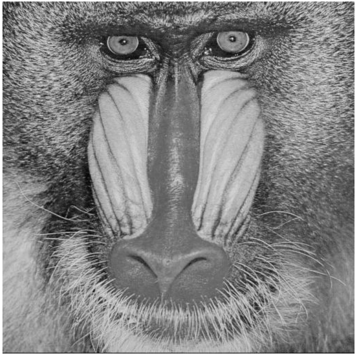
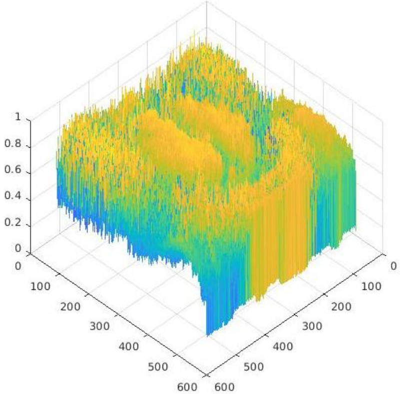
Hardware para vehículos sin conductor



El simulador Webots

Imágenes como funciones

- Una imagen (en escala de grises) es una función $I(x, y)$ donde x, y son variables discretas en coordenadas de imagen y la función I es intensidad luminosa.
- Las imágenes también pueden considerarse como arreglos bidimensionales de números entre un mínimo y un máximo (usualmente 0-255).
- Aunque formalmente una imagen es un mapeo $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, en la práctica, tanto x, y como I son varialbes discretas con valores entre un mínimo y un máximo.
- Las imágenes de color son funciones vectoriales $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ donde cada componente de la función se llama canal.

Convolución

Si se conoce la respuesta al impulso $H(x,y)$ de un sistema SLID, se puede obtener la salida $O(x,y)$ ante cualquier entrada $I(x,y)$, mediante la convolución, definida como:

$$O(x,y) = I(x,y) * H(x,y) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} I(i,j)H(x-i,y-j)$$

Ejemplos:

$$\begin{bmatrix} 3 & 1 & 4 & 1 \\ 5 & 9 & 2 & 6 \\ 5 & 3 & 5 & 8 \\ 9 & 7 & 9 & 3 \end{bmatrix} * \begin{bmatrix} 1 & -1 \end{bmatrix} = \begin{bmatrix} 3 & -2 & 3 & -3 & -1 \\ 5 & 4 & -7 & 4 & -6 \\ 5 & -2 & 2 & 3 & -8 \\ 9 & -2 & 2 & -6 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 & 4 & 1 \\ 5 & 9 & 2 & 6 \\ 5 & 3 & 5 & 8 \\ 9 & 7 & 9 & 3 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 4 & 1 \\ 2 & 8 & -2 & 5 \\ 0 & -6 & 3 & 2 \\ 4 & 4 & 4 & -5 \\ -9 & -7 & -9 & -3 \end{bmatrix}$$

El filtro Gaussiano

Gradiente

El gradiente de una imagen está definido como:

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]$$

Las derivadas parciales se puede aproximar mediante diferencias finitas:

$$\begin{aligned}
\frac{\partial I}{\partial x} &= \lim_{\Delta x \to 0} \frac{I(x+\Delta x, y) - I(x,y)}{\Delta x} \approx I_{i,j} - I_{i,j-1} \\
\frac{\partial I}{\partial y} &= \lim_{\Delta y \to 0} \frac{I(x, y+\Delta y) - I(x,y)}{\Delta y} \approx I_{i,j} - I_{i-i,j}
\end{aligned}$$

donde (i, j) representan las coordenadas de imagen renglón-columna. Estas diferencias finitas se puede obtener mediante una convolución:

$$\begin{aligned}
\frac{\partial I}{\partial x} &\approx I * \begin{bmatrix} 1 & -1 \end{bmatrix} \\
\frac{\partial I}{\partial y} &\approx I * \begin{bmatrix} 1 \\ \text{-1} \end{bmatrix}
\end{aligned}$$

Gradiente

Una mejor aproximación de la derivada es no solo tomar la diferencia entre el valor actual y el anterior $(I_{i,j} - I_{i-1,j})$, sino promediarlo con la diferencia $(I_{i+1,j} - I_{i,j})$:

$$\frac{1}{2}[(I_{i,j} - I_{i-1,j}) + (I_{i+1,j} - I_{i,j})] = \frac{1}{2}(I_{i+1,j} - I_{i-1,j})$$

Generalmente se ignora el coeficiente y se utilizan los siguientes Kernels:

$$\begin{aligned}\frac{\partial I}{\partial x} &\approx I * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \\ \frac{\partial I}{\partial y} &\approx I * \begin{bmatrix} 1 \\ 0 \\ \text{-1} \end{bmatrix}\end{aligned}$$

El filtro de Sobel

El Operador de Sobel o Filtro de Sobel consiste en un Kernel que permite obtener las derivadas parciales, aproximadas por diferencias finitas, y promediadas con un filtro Gaussiano:

$$
\overbrace{\begin{bmatrix} 10 & 12 & 20 & 28 & 30 \\ 10 & 13 & 19 & 29 & 30 \\ 10 & 11 & 20 & 27 & 29 \\ 11 & 13 & 21 & 28 & 30 \\ 9 & 12 & 19 & 28 & 31 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \longrightarrow \frac{1}{2} \begin{bmatrix} 2 & 10 & 16 & 10 & 2 \\ 3 & 9 & 16 & 11 & 1 \\ 1 & 10 & 16 & 9 & 2 \\ 2 & 10 & 15 & 9 & 2 \\ 3 & 10 & 16 & 12 & 3 \end{bmatrix}}^{\frac{\partial I}{\partial x}} * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \longrightarrow \frac{1}{8} \begin{bmatrix} 9 & 39 & 64 & 41 & 7 \\ 9 & 38 & 64 & 41 & 6 \\ 7 & 39 & 63 & 38 & 7 \\ 8 & 40 & 62 & 39 & 9 \\ 11 & 40 & 63 & 45 & 11 \end{bmatrix}
$$

Promedio ponderado

Se realiza un proceso similar para la derivada parcial en Y. Aplicando la propiedad asociativa de la convolución, se obtienen los siguientes ekernels:

$$
S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}
$$

Magnitud y Ángulo

El gradiente en cada pixel de la imagen se puede calcular mediante la approximación de las derivadas parciales:

$$\frac{\partial I}{\partial x} \approx I * Sx = G_x$$
$$\frac{\partial I}{\partial y} \approx I * Sy = G_y$$

En la mayoría de las aplicaciones es más últil expresar el gradiente en forma polar:

$$\nabla I = G_m \angle G_a$$

Donde la magnitud del gradiente y la fase, para cada pixel, se calculan como:

$$G_{m_{i,j}} = \sqrt{G_{x_{i,j}}^2 + G_{y_{i,j}}^2}$$
$$G_{a_{i,j}} = \text{atan2}(G_{y_{i,j}}, G_{y_{i,j}})$$

Detector de Bordes de Canny

El detector de bordes de Canny es un detector basado en gradiente que consta de los siguientes pasos básicos:

1. Obtención del gradiente en magnitud y ángulo, mediante operadores de Sobel
2. Supresión de puntos no máximos
3. Aplicación de un doble umbral

Aunque no es un paso en sí del Detector de Canny, generalmente se considera como primer paso la aplicación de un filtro Gaussiano para disminuir el ruido.

Obtención del gradiente

Después del filtro Gaussiano, el primer paso es obtener el gradiente de la imagen mediante el Filtro de Sobel, en la forma de magnitud y ángulo:

10	12	20	28	30
10	13	19	29	30
10	11	20	27	29
11	13	21	28	30
9	12	19	28	31

*

1	0	-1
2	0	-2
1	0	-1

=

9	39	64	41	7
9	38	64	41	6
7	39	63	38	7
8	40	62	39	9
11	40	63	45	11

10	12	20	28	30
10	13	19	29	30
10	11	20	27	29
11	13	21	28	30
9	12	19	28	31

*

1	2	1
0	0	0
-1	-2	-1

=

1	1	0	1	1
-1	-2	-2	-3	-4
3	3	3	0	-1
-2	0	0	3	7
-7	-6	-5	-1	3

$$x, y \rightarrow r\angle\theta$$

$\longrightarrow$

9.05	39.01	64.00	41.01	7.07
9.05	38.05	64.03	41.10	7.21
7.61	39.11	63.07	38.00	7.07
8.24	40.00	62.00	39.11	11.40
13.03	40.44	63.19	45.01	11.40

0.11	0.02	0	0.02	0.14
-0.11	-0.05	-0.03	-0.07	-0.58
0.40	0.07	0.04	0	-0.14
-0.24	0	0	0.07	0.66
-0.56	-0.14	-0.07	-0.02	0.26

Supresión de no máximos

Este paso consiste en comparar la magnitud de cada pixel, con los pixeles anterior y posterior en la dirección del gradiente. Aunque la fase es un ángulo en $[-\pi, \pi]$, la dirección del gradiente se debe redondear a algún valor correspondiente a la connectividad 8: *N*, *NE*, *E*, *SE*. Debido a que el pixel se compara en la dirección positiva y negativa del gradiente, no es necesario considerar las direcciones *S*, *SW*, *W*, *NW*.

9.05	39.01	64.00	41.01	7.07
9.05	38.05	64.03	41.10	7.21
7.61	39.11	63.07	38.00	7.07
8.24	40.00	62.00	39.11	11.40
13.03	40.44	63.19	45.01	11.40

$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\searrow$
$\nearrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\nearrow$
$\searrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$

0	0	64.00	0	0
0	0	64.03	0	0
0	0	63.07	0	0
0	0	62.00	0	0
0	0	63.19	0	0

Para cada pixel p_i, considere p_{i+1}, el pixel siguiente en la dirección del gradiente, y p_{i-1}, el pixel anterior, en la dirección del gradiente. El valor para cada pixel q_i en la imagen resultante es:

$$q_i = \begin{cases} p_i & \text{si} \quad p_i > p_{i+1} \quad \text{y} \quad p_i > p_{i-1} \\ 0 & \text{en otro caso} \end{cases}$$

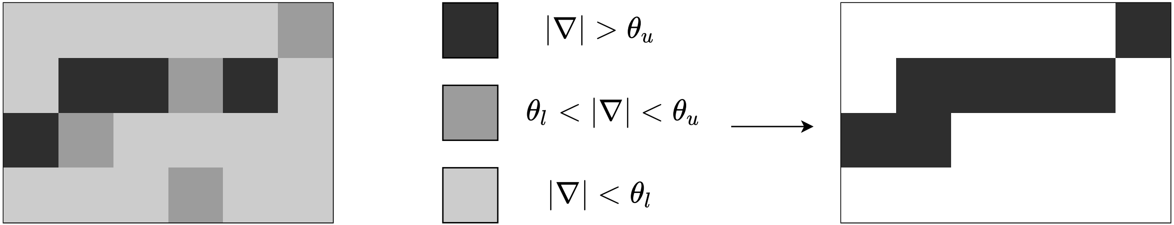
Aplicación de doble umbral

En este paso, se definen dos umbrales: superior θ_u e inferior θ_l. Los pixeles se clasifican en tres tipos:

- Fuertes: pixeles con magnitud del gradiente mayor que el umbral superior $|\nabla| > \theta_u$
- Débiles: pixeles con magnitud del gradiente entre ambos umbrales $\theta_l < |\nabla| < \theta_u$
- Suprimidos: pixeles con magnitud del gradiente menor que el umbral inferior $|\nabla| < \theta_l$

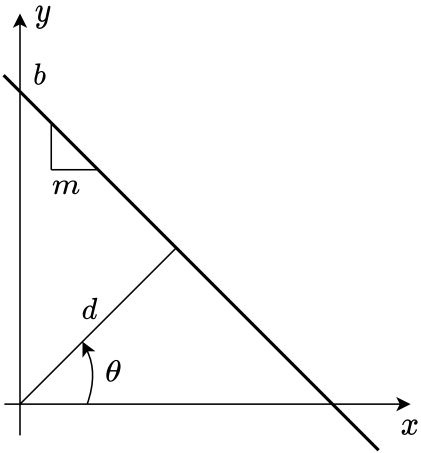
La imagen resultante se forma con las siguientes reglas:

- Todos los pixeles fuertes son parte de un borde.
- Todos los pixeles suprimidos no son bordes.
- Los pixeles débiles son parte de un borde solo si están conectados (en conectividad 8) con un pixel fuerte.



La Transformada Hough

La Transformada Hough es un método que permite encontrar líneas, círculos y otras formas geométricas que se puedan describir fácilmente mediante expresiones analíticas. En el caso de las líneas, se trata de encontrar los dos parámetros que describen la recta:

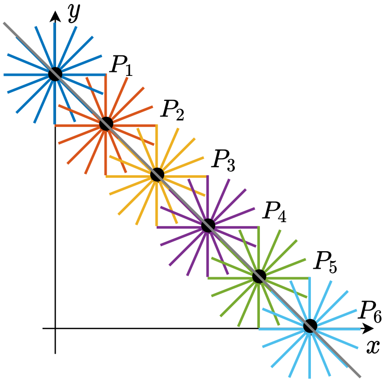


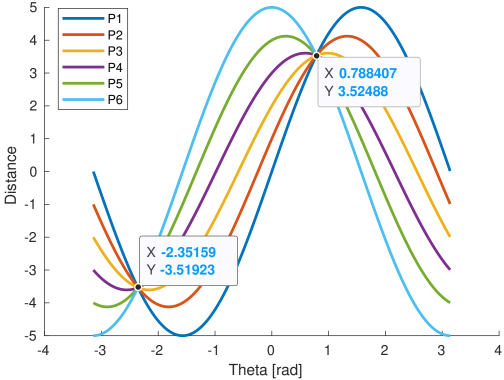
- La forma pendiente-ordenada $y = mx + b$ tiene la desventaja de no poder expresar líneas verticales.
- La forma canónica $Ax + By + C$ requiere de una normalización $A_1 x + B_1 y + 1 = 0$ para que solo sean dos parámetros.
- Una forma más conveniente, es la forma normal $d = x \cos\theta + y \sin\theta$
- Esta última forma tiene una ventaja: si la línea correponde a un borde, el ángulo θ será la dirección del gradiente.

El Espacio de Hough

El Espacio de Hough, para el caso de las líneas, es el conjunto de todos los posibles pares (θ, d).

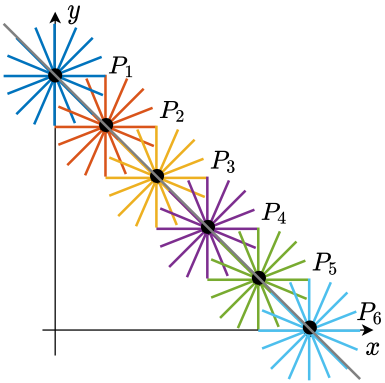
- Una recta L en el espacio cartesiano corresponde a un punto P_h en el Espacio de Hough
- Un punto P_c en el espacio cartesiano corresponde a una curva C en el Espacio de Hough. Esta curva representa los parámetros (θ_i, d_i) de todas las rectas L_i que pasan por el punto P.

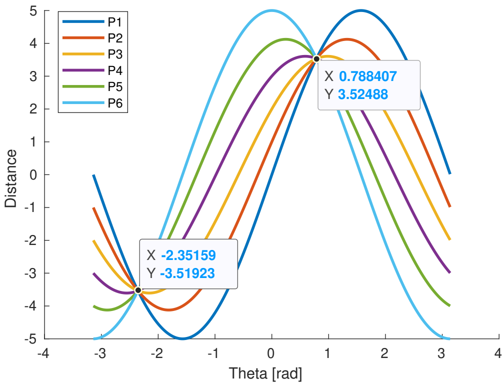




Extracción de Líneas por Transformada Hough

Este método consiste en encontrar las curvas C_i en el espacio de Hough que pasan por cada punto P_c en el espacio cartesiano. Los puntos P_h en el Espacio de Hough por donde pasen más curvas C_i corresponderán a las rectas resultantes en el espacio cartesiano.





Extracción de Líneas por Transformada Hough

Input: Imagen binaria M, umbral mínimo de votos a_{min}
Output: Líneas expresadas en la forma (d, θ)

Inicializar en 0 un conjunto A de acumuladores para cada par cuantizado (d_k, θ_k)
forall *Pixel* $M[i,j] \neq 0$ **do**
    forall *Ángulo* θ_k *cuantizdo* **do**
        $d = j\cos\theta_k + i\sin\theta_k$ $d_k =$ valor cuantizado de d Incrementar en uno el acumulador correspondiente $A[d_k, \theta_k]$
    end
end
forall $a \in A$ **do**
    if $a > a_{min}$ **then**
        Agregar la línea (d_k, θ_k) al conjunto de líneas detectadas
    end
end
Devolver el conjunto de líneas detectadas

Gracias

Contacto

Dr. Marco Negrete
Profesor Asociado C
Departamento de Procesamiento de Señales
Facultad de Ingeniería, UNAM.

mnegretev.info
marco.negrete@ingenieria.unam.edu