

# Robot Path Planning

Cristóbal Lobos - Alan Grez

Universidad Técnica Federico Santa María

*cristobal.lobos@sansano.usm.cl*

*alan.grez@sansano.usm.cl*

January 7, 2021

# Índice

- 1 Motivación y contexto
- 2 Modelamiento Matemático
  - Definición General de Robot Path Planning
  - Modelo 3D
  - Modelo 2D
- 3 Algoritmo RRT
  - Definiciones
  - Explicación Pseudocódigo
- 4 Simulaciones realizadas
  - Mapa con forma del logo de Unreal Engine 4
  - Mapa con forma de Pikachu
  - Laberinto Circular
- 5 Conclusiones, Dificultades y Trabajos futuros.

# Motivación

- Para los robots, moverse es una acción que no es para nada trivial. Uno de los mayores desafíos para los robots autónomos es tener un sentido de ubicación en un espacio dado.
- En particular, esto representa al **problema de navegación**, el cual puede resumirse en las siguientes preguntas. ¿Dónde estoy? ¿A dónde voy? ¿Cómo llego ahí?
- La última corresponde al problema de **planificación de caminos**, en inglés, este último corresponde a **path planning**. Este último consiste en encontrar el camino más corto que debe recorrer el robot desde un punto inicial a un punto final en un ambiente dado, el cual posee obstáculos.

# Problema General

- Sea  $S \subset \mathbb{R}^2$  ó  $\mathbb{R}^3$ , una superficie de movimiento,  $R \subset S$  un subconjunto que se traslada por  $S$ , el cual representa al robot, y  $O \subset S$  conjunto de obstáculos tales que la geometría, posición y orientación de  $R$  y  $O$  son conocidos a priori.
- Entonces, nuestro problema es: Dada las posiciones inicial y final pertenecientes a  $P \subset S$ , por demostrar que existe dicho  $P$  tal que para todo  $p \in P$ ,  $R(p) \cap O = \emptyset$  sujeto a minimizar cierta función de costo. Notar que esta función objetivo depende netamente de la superficie estudiada.

## Problema Específico

Introduzcamos un poco de notación:

Sea  $G : [0, 1]^2 \rightarrow \mathbb{R}$  una función  $C^1([0, 1]^2)$ , consideremos:

$$S = \{(x, y, G(x, y)) \in \mathbb{R}^3 \mid (x, y) \in [0, 1]^2\},$$

en el cual definiremos los puntos de inicio y fin respectivamente como:

$$A := (0, 0, G(0, 0)), \quad B := (1, 1, G(1, 1))$$

Y definimos  $O$ , el conjunto de obstáculos que veremos más adelante, y

$P := \Omega \setminus \Omega_O$ , el conjunto de las trayectorias factibles, donde

$$\Omega := \{\gamma : [0, 1] \rightarrow S \mid \gamma(0) = A, \gamma(1) = B, \gamma \in C^1([0, 1])\}$$

$$\Omega_O := \{\gamma \in \Omega \mid \exists t \in [0, 1], \gamma(t) \in O\}$$

# Función Objetivo

Ahora, al momento de presentar el problema general, hablamos de una función de costo a minimizar, esta función va de la mano con la motivación de nuestro problema, i. e., una función que minimice tanto la distancia recorrida como el gasto de combustible, luego, escribimos nuestro problema de la forma:

$$\min_{\gamma \in \Omega} F(\gamma) \quad \text{ó} \quad \min_{\gamma \in P} F(\gamma)$$

**sujeto a**  $\gamma \notin \Omega_0$

con

$$F(\gamma) = C_d \int_{\gamma} dl + W \int_0^1 \left( \frac{\gamma' \cdot e_3}{\sqrt{(\gamma' \cdot e_3)^2 + 1}} + \frac{C_r}{\sqrt{(\gamma' \cdot e_3)^2 + 1}} \right) dt,$$

$C_d$  constante relativa a la distancia y condiciones relativas al sólido que se desplaza y  $C_r$  relativas a la gasto de combustible.

## Modelo 2D

- Consideremos que el robot es un punto en  $\mathbb{R}^2$ . Consideremos  $M \subset \mathbb{R}^2$  que llamaremos mapa, donde el mapa consiste de una sección rectangular  $M$  del primer cuadrante del plano cartesiano, junto a un conjunto de obstáculos  $O$  que se encuentra adentro de esta sección rectangular. Definimos el conjunto de **caminos factibles**  $\Omega$  como  $\Omega = \{\gamma : [0, 1] \rightarrow M / \gamma \text{ simple, continua; ; } \nexists t \in [0, 1] \text{ t.q } \gamma(t) \in O; \gamma(0) = A, \gamma(1) = B\}$ .
- Luego nuestro problema de minimización es:

$$\min_{\gamma \in \Omega} \int_{\gamma} dl$$

- Además, relajamos las condiciones de la curva, ya que en particular, queremos encontrar estas curvas de manera gráfica un camino factible en un mapa dado.

## ¿Por qué un modelo 2D?

- Notemos que consideramos el modelo 2D como un problema relajado del 3D, esto pues la principal diferencia entre el conjunto  $\Omega$  (el conjunto de los caminos factibles) es que le hemos quitado la condición de suavidad a la curva. Esto, en efecto, se sustenta en la complejidad computacional que representa programar una curva "suave".
- Por otro lado, si tuviéramos la condición de suavidad, demostrar la existencia de mínimo se convierte en un problema altamente no trivial, esto pues habría que exigir cierta suavidad y regularidad a los obstáculos, lo cual se convertiría más en un problema de propiedades matemáticas del conjunto de obstáculos que de minimización, lo cual escapa de los objetivos del problema.



# Pseudocódigo

- Este algoritmo crea un árbol de búsqueda que agrega puntos reescalados de puntos admisibles al azar a lo largo del mapa sucesivamente hasta que si un punto se encuentra a una distancia menor a  $\tau$  del punto final, entonces el algoritmo procede a calcular la ruta más corta recorriendo el árbol en sentido contrario, es decir, partiendo desde el nodo donde se encuentra el punto final, hasta la raíz sumando en cada iteración la distancia entre puntos.

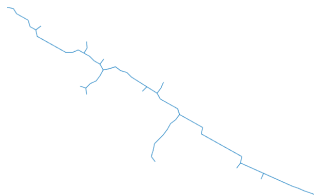


Figure: Ejemplo de árbol de algoritmo RRT

# Realización de las simulaciones

- Mediante el uso de MATLAB, utilizando el algoritmo RRT basado en el trabajo de Rahul Kala, se simularon 3 mapas.
- Se varía el paso realizado, y la tolerancia del algoritmo RRT. Además, se aumentó la resolución de las imágenes para poder realizar pasos de píxeles más largos. Esto con el fin de obtener más caminos factibles y compararlos entre sí.
- Se realizaron 4 simulaciones por resolución para el mapa con forma de Pikachu, y para el mapa con forma del logo de Unreal Engine 4, por otra parte, se realizaron 2 simulaciones por resolución para el laberinto circular.

## Punto inicial y final

- La resolución de la imagen original es de  $900 \times 900$  pixeles, mientras que la de la aumentada es de  $1500 \times 1500$ . El punto inicial y el punto final se encuentran en la siguiente figura.



Figure: Punto inicial y final en el logo de Unreal Engine 4.

## Simulación 1 - Resolución Normal

- Para un paso de 200 pixeles y una tolerancia de 50 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1046 pixeles y un tiempo de ejecución de 17,65 segundos.



**Figure:** Camino obtenido con un paso de 200 pixeles y una tolerancia de 50 pixeles para el logo de Unreal Engine 4 de resolución normal.

## Simulación 2 - Resolución Normal

- Para un paso de 100 pixeles y una tolerancia de 50 pixeles se obtuvo el siguiente camino, con una longitud de curva de 871 pixeles y un tiempo de ejecución de 16,04 segundos.



**Figure:** Camino obtenido con un paso de 100 pixeles y una tolerancia de 50 pixeles para el logo de Unreal Engine 4 de resolución normal.

## Simulación 3 - Resolución Normal

- Para un paso de 70 pixeles y una tolerancia de 35 pixeles se obtuvo el siguiente camino, con una longitud de curva de 822 pixeles y un tiempo de ejecución de 23,24 segundos.



**Figure:** Camino obtenido con un paso de 70 pixeles y una tolerancia de 35 pixeles para el logo de Unreal Engine 4 de resolución normal.

## Simulación 4 - Resolución Normal

- Para un paso de 30 pixeles y una tolerancia de 15 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1188 pixeles y un tiempo de ejecución de 38,52 segundos.



**Figure:** Camino obtenido con un paso de 30 pixeles y una tolerancia de 15 pixeles para el logo de Unreal Engine 4 de resolución normal.

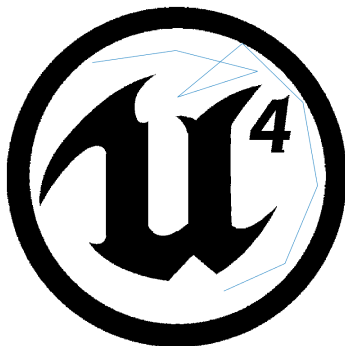
# Análisis de resultados

- Podemos notar que el algoritmo tiende a ejecutarse en un mayor tiempo si el paso decrece.
- La menor longitud de curva obtenida fue de 822 pixeles utilizando un paso de 70 pixeles, mientras que la mayor longitud de curva obtenida fue de 1188 pixeles utilizando un paso de 30 pixeles.
- Además, notemos que ambas curvas obtenidas que fueron de menor longitud, recorren el obstáculo en sentido antihorario, mientras que las de mayor longitud, la recorrieron en sentido horario, lo que nos indica una tendencia a que la curva más corta se encuentra rodeando el obstáculo en sentido antihorario. (Para este obstáculo en particular)



## Simulación 1 - Resolución Aumentada

- Para un paso de 300 pixeles y una tolerancia de 100 pixeles se obtuvo el siguiente camino, con una longitud de curva de 2339 pixeles y un tiempo de ejecución de 48,25 segundos.



**Figure:** Camino obtenido con un paso de 300 pixeles y una tolerancia de 100 pixeles para el logo de Unreal Engine 4 de resolución aumentada.

## Simulación 2 - Resolución Aumentada

- Para un paso de 250 píxeles y una tolerancia de 100 píxeles se obtuvo el siguiente camino, con una longitud de curva de 1570 píxeles y un tiempo de ejecución de 19,13 segundos.



**Figure:** Camino obtenido con un paso de 250 pixeles y una tolerancia de 100 pixeles para el logo de Unreal Engine 4 de resolución aumentada.

## Simulación 3 - Resolución Aumentada

- Para un paso de 150 pixeles y una tolerancia de 50 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1910 pixeles y un tiempo de ejecución de 26,97 segundos.



**Figure:** Camino obtenido con un paso de 150 pixeles y una tolerancia de 50 pixeles para el logo de Unreal Engine 4 de resolución aumentada.

## Simulación 4 - Resolución Aumentada

- Para un paso de 60 pixeles y una tolerancia de 30 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1493 pixeles y un tiempo de ejecución de 18,87 segundos.



**Figure:** Camino obtenido con un paso de 60 pixeles y una tolerancia de 30 pixeles para el logo de Unreal Engine 4 de resolución aumentada.

## Análisis de resultados

- Observamos que de las curvas obtenidas, solo dos son factibles, ya que la curva obtenida con un paso de 300 pixeles y una tolerancia de 100 pixeles no es simple, y la curva obtenida con un paso de 150 pixeles y una tolerancia de 50 pixeles tampoco lo es.
- Esto es debido a que al tomar una tolerancia baja en con respecto al paso, en simples palabras, el algoritmo se pasa de la meta, y se debe devolver, lo que provoca una perdida de la simplicidad del camino.
- Con respecto a las otras dos curvas, observamos que la longitud de curva menor fue obtenida mediante el uso de un paso de 60 pixeles y una tolerancia de 30 pixeles, la cual fue de 1493 pixeles, mientras que la obtenida mediante el uso de 250 pixeles y una tolerancia de 100 pixeles tuvo una longitud de 1570 pixeles.
- Además, al tener una tolerancia menor que la mitad del paso, el algoritmo tuvo un mayor tiempo de ejecución que al ejecutarlo con un paso menor igual a tolerancia de la mitad de este.

## Punto inicial y final

- La resolución de la imagen original es de  $900 \times 981$  pixeles, mientras que la de la aumentada es de  $1800 \times 1962$ . El punto inicial y el punto final se encuentran en la siguiente figura.

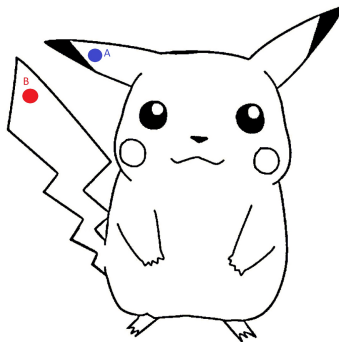


Figure: Punto inicial y final en el mapa con forma de Pikachu.

## Simulación 1 - Resolución Normal

- Para un paso de 230 píxeles y una tolerancia de 60 píxeles se obtuvo el siguiente camino, con una longitud de curva de 2493 píxeles y un tiempo de ejecución de 26,39 segundos.



**Figure:** Camino obtenido con un paso de 230 píxeles y una tolerancia de 60 píxeles para el mapa con forma de Pikachu de resolución normal.

## Simulación 2 - Resolución Normal

- Para un paso de 150 píxeles y una tolerancia de 70 píxeles se obtuvo el siguiente camino, con una longitud de curva de 2007 píxeles y un tiempo de ejecución de 25,01 segundos.



**Figure:** Camino obtenido con un paso de 150 pixeles y una tolerancia de 70 pixeles para el mapa con forma de Pikachu de resolución normal.



## Simulación 3 - Resolución Normal

- Para un paso de 90 pixeles y una tolerancia de 45 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1679 pixeles y un tiempo de ejecución de 22,68 segundos.



**Figure:** Camino obtenido con un paso de 90 pixeles y una tolerancia de 45 pixeles para el mapa con forma de Pikachu de resolución normal.

## Simulación 4 - Resolución Normal

- Para un paso de 25 pixeles y una tolerancia de 12 pixeles se obtuvo el siguiente camino, con una longitud de curva de 1617 pixeles y un tiempo de ejecución de 506,87 segundos.



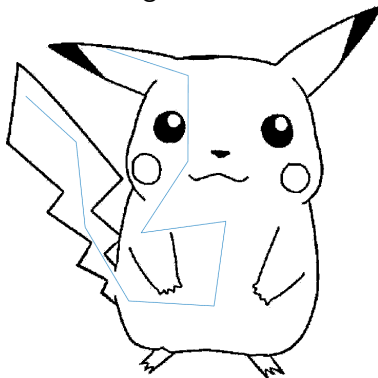
**Figure:** Camino obtenido con un paso de 25 pixeles y una tolerancia de 12 pixeles para el mapa con forma de Pikachu de resolución normal.

# Análisis de resultados

- Notemos que, como se mencionó al principio, el camino que más se acerca al borde izquierdo, que corresponde al obtenido usando un paso de 25 pixeles y una tolerancia de 12 pixeles, tuvo la menor longitud, la cual es de 1617 pixeles. Cabe destacar que este algoritmo tuvo un tiempo de ejecución 506,87 segundos, el cual es notablemente mayor que los demás.
- En base a lo último mencionado, se observa una tendencia a tener mayor tiempo de ejecución a medida que el paso utilizado es menor.
- Además, para este mapa, se observa que a medida que aumenta el paso de los pixeles, aumenta la longitud de la curva, y como se puede observar, la curva se aleja del borde a medida que el paso utilizado es mayor.

## Simulación 1 - Resolución Aumentada

- Para un paso de 400 pixeles y una tolerancia de 100 pixeles se obtuvo el siguiente camino, con una longitud de curva de 3283 pixeles y un tiempo de ejecución de 5.52 segundos.



**Figure:** Camino obtenido con un paso de 400 pixeles y una tolerancia de 100 pixeles para el mapa con forma de Pikachu de resolución normal.

## Simulación 2 - Resolución Aumentada

- Para un paso de 300 pixeles y una tolerancia de 100 pixeles se obtuvo el siguiente camino, con una longitud de curva de 3290 pixeles y un tiempo de ejecución de 20,21 segundos.



**Figure:** Camino obtenido con un paso de 300 pixeles y una tolerancia de 100 pixeles para el mapa con forma de Pikachu de resolución normal.

## Simulación 3 - Resolución Aumentada

- Para un paso de 150 pixeles y una tolerancia de 75 pixeles se obtuvo el siguiente camino, con una longitud de curva de 3664 pixeles y un tiempo de ejecución de 22,61 segundos.



**Figure:** Camino obtenido con un paso de 150 pixeles y una tolerancia de 50 pixeles para el mapa con forma de Pikachu de resolución normal.

## Simulación 4 - Resolución Aumentada

- Para un paso de 70 pixeles y una tolerancia de 35 pixeles se obtuvo el siguiente camino, con una longitud de curva de 2919 pixeles y un tiempo de ejecución de 32,57 segundos.



**Figure:** Camino obtenido con un paso de 70 pixeles y una tolerancia de 35 pixeles para el mapa con forma de Pikachu de resolución normal.

# Análisis de resultados

- Tal como se ha visto en casos anteriores, la simulación realizada con el mayor paso tuvo el menor tiempo de ejecución, esta corresponde a la realizada con un paso de 400 pixeles y tolerancia de 100 pixeles, donde se obtuvo un tiempo de ejecución de 5,52 segundos.
- La menor longitud de curva, al igual que en la resolución normal de la figura, se obtuvo la menor longitud de curva en el camino más apegado al borde izquierdo, esta corresponde a la curva obtenida con un paso de 70 pixeles y 35 pixeles de tolerancia, con una longitud de 2919 pixeles.



## Punto inicial y final

- La resolución de la imagen original es de  $1258 \times 1251$  pixeles, mientras que la de la aumentada es de  $2516 \times 2502$ . El punto inicial y el punto final se encuentran en la siguiente figura.

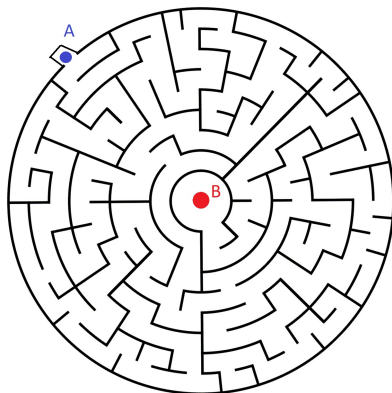
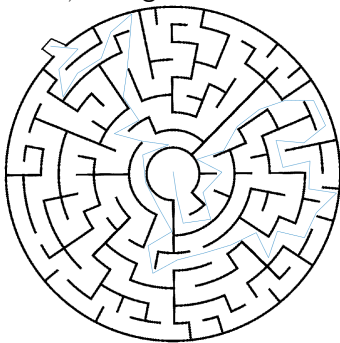


Figure: Punto inicial y final en el laberinto circular.

## Simulación 1 - Resolución Normal

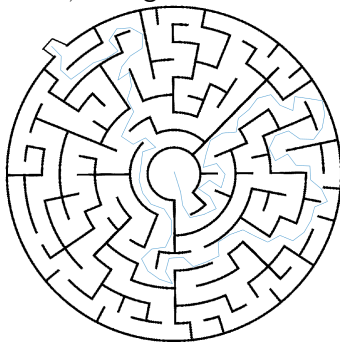
- Para un paso de 100 pixeles y una tolerancia de 50 pixeles se obtuvo el siguiente camino, con una longitud de curva de 4690 pixeles y un tiempo de ejecución de 61,44 segundos.



**Figure:** Camino obtenido con un paso de 100 pixeles y una tolerancia de 50 pixeles para el mapa con forma de laberinto de forma circular resolución normal.

## Simulación 2 - Resolución Normal

- Para un paso de 50 pixeles y una tolerancia de 25 pixeles se obtuvo el siguiente camino, con una longitud de curva de 4722 pixeles y un tiempo de ejecución de 60,47 segundos.



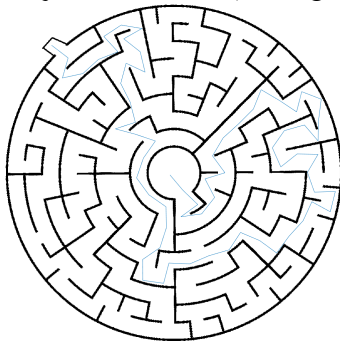
**Figure:** Camino obtenido con un paso de 50 pixeles y una tolerancia de 25 pixeles para el mapa con forma de laberinto de forma circular resolución normal.

# Análisis de resultados

- Notemos que ambos caminos encontrados siguen el camino dado por la solución del laberinto, sin embargo, el camino obtenido por el paso 100 pixeles con una tolerancia de 50 pixeles, no realiza curvas tan zig-zageantes, esto se puede notar en el lado derecho del laberinto. De esta manera, tenemos que la longitud de esta curva es de 4690 pixeles, mientras que el otro camino obtenido posee una longitud de curva de 4722 pixeles, donde esta curva zig-zagea más al realizar movimientos más "finos".

## Simulación 1 - Resolución Aumentada

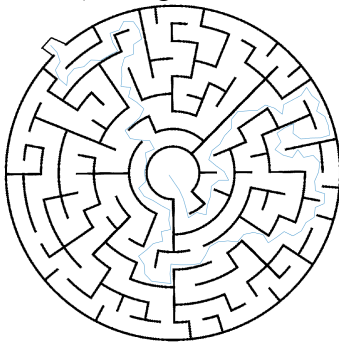
- Para un paso de 150 píxeles y una distancia 50 de igualdad de píxeles se obtuvo el siguiente camino, con una longitud de curva de 9892 píxeles y un tiempo de ejecución de 109,38 segundos.



**Figure:** Camino obtenido con un paso de 150 píxeles y una tolerancia de 50 píxeles para el mapa con forma de laberinto de forma circular resolución aumentada.

## Simulación 2 - Resolución Aumentada

- Para un paso de 70 píxeles y una tolerancia de 35 píxeles se obtuvo el siguiente camino, con una longitud de curva de 9443 píxeles y un tiempo de ejecución de 258,36 segundos.



**Figure:** Camino obtenido con un paso de 70 píxeles y una tolerancia de 35 píxeles para el mapa con forma de laberinto de forma circular de resolución aumentada.

## Análisis de resultados

- Observando los caminos obtenidos, notamos que el camino obtenido con un paso 70 pixeles y una tolerancia de 35 pixeles tiene una longitud de curva menor que la del otro camino obtenido, esta corresponde a una longitud de 9443 pixeles, mientras que la otra corresponde a una de 9892 pixeles. Esto es debido a que el camino mencionado rodea más el borde del laberinto que el otro, lo cual produce una menor longitud de curva, además, como se puede observar en su respectiva figura, el otro camino encontrado tiende a alejarse del borde al girar, lo cual produce una mayor longitud de curva de 9892 pixeles.

## Conclusiones y discusión

- Mediante el uso del algoritmo RRT, si bien no se obtienen caminos que sean el óptimo de nuestro problema de minimización, nos entrega caminos factibles para que el robot recorra en los ambientes simulados.
- De manera general, mientras mayor sea el paso realizado, con una tolerancia no tan menor con respecto al paso realizado, el algoritmo no toma mucho tiempo en encontrar un camino, y en contraparte, si el paso realizado es pequeño, el algoritmo toma un tiempo no menor en encontrar un camino.
- Un problema de optimización que podría surgir a partir de este problema, sería escoger el paso óptimo para el algoritmo RRT de tal manera de que nos entregue la curva con la menor longitud de curva para una configuración de obstáculos dada.



Gracias por su atención.

# Robot Path Planning

Cristóbal Lobos - Alan Grez

Universidad Técnica Federico Santa María

*cristobal.lobos@sansano.usm.cl*

*alan.grez@sansano.usm.cl*

January 7, 2021