

# Examen Final

Adriana Basurto Vázquez A01323251

Patricia Cañete Leyva A01323992

Luis Alfredo León Villapún A01322275

Diego Osorio Sánchez A01328920

ITESM

Puebla, México

**Resumen**— En este documento se presentan los resultados del examen final del curso de Visión para Robots, el cual consiste en utilizar un algoritmo de planeación de trayectorias para detectar posibles caminos en una imagen, así como la generación de distintos caminos entre un punto a otro con el algoritmo A\*.

**Índice de términos:** *Espacio de configuración, Probabilistic Road Maps, A\*.*

## I. INTRODUCCIÓN

Lo más importante dentro del procesamiento de imágenes, así como de la ciencia en general, es la aplicación que se le da al conocimiento o a las herramientas que genera esta misma. Por ende, una parte vital dentro del procesamiento de imágenes corresponde a lo que se puede hacer con esta. Una de esas aplicaciones es el buscar caminos óptimos a partir de una imagen, de donde robots o drones se pueden basar para circular sin generar colisiones y siguiendo un camino óptimo. Para esto se usan diversos algoritmos, entre los cuales se encuentra Probabilistic Road Maps (PRM), y A\*, los cuales fueron ocupados en esta práctica para cumplir los objetivos trazados. Finalmente, se utilizó una imagen satelital del Tecnológico de Monterrey, Campus Puebla, para encontrar caminos interesantes entre diversos lugares del campus.

## II. MARCO TEÓRICO

Es común, en el marco de la robótica, que un robot tenga que moverse de manera autónoma, sin embargo, es un reto darle la información necesaria al agente para que pueda trasladarse en su entorno sin generar colisiones. Es por eso que se han ideado diversos algoritmos que le proporcionan al robot las herramientas necesarias para cumplir con este fin.

### Espacio de Configuración

Se define como “la transformación de un mundo regular a un espacio considerando todas las posibles configuraciones de un robot” (Carballo, 2017).

Considerando al espacio-C como un conjunto de “posiciones” o “estados” que puede tomar un robot, en referencia a un plano cartesiano, o un mundo en 2D, podemos dividir este espacio en dos grandes subconjuntos, que ayudan en gran medida a definir el espacio que puede ocupar el robot, el cual es un punto en el mapa. Primero que nada,  $C_{free}$ , o espacio libre, es el subconjunto donde el agente puede movilizarse. En

segundo lugar, está  $C_{obs}$ , o espacio ocupado, el cual representa todos los obstáculos en el mundo 2D.

Existen dos propuestas para discretizar este espacio, dado que es continuo. La primera opción es la llamada planeación combinatorial, donde simplemente se toma el  $C_{free}$  como un grafo, y se busca el camino entre un punto A y un punto B utilizando este grafo. La segunda opción es la planeación basada en muestreo. Aquí, el grafo generado se obtiene después de probar que las conexiones entre las aristas no generen ninguna colisión en algún elemento de  $C_{obs}$ .

### Probabilistic Road Maps

En este algoritmo de planeación basada en muestreo, se toman de manera aleatoria diversos puntos del  $C_{free}$ . De manera local se conectan los vértices, y se verifica que no generen colisión alguna. Si se genera alguna colisión, esta conexión del grafo se descarta.

Para generar la conexión, se puede utilizar un algoritmo de K-nn vecinos más cercanos, o se puede optar por una solución más sencilla que es delimitar un radio y hacer la operación para todos los nodos que se encuentren dentro de esa vecindad.

Un problema con PRM es que en ocasiones no genera un grafo completamente conexo, y se aísla. Esto resulta en que el robot no podrá reconocer caminos entre ciertos puntos del espacio-C. Sin embargo, generar una vecindad lo suficientemente grande normalmente produce un grafo conexo.

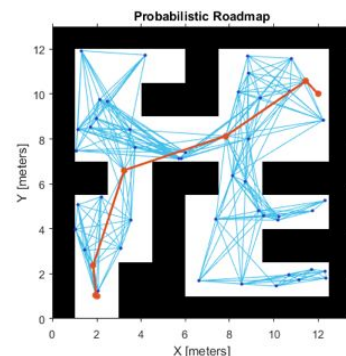


Figura 1. Ejemplo visual de un PRM. [1]

### Algoritmo A\*

Una vez generado el grafo que el agente utilizará para moverse a través de  $C_{free}$ , es necesario encontrar un camino óptimo entre todas las posibles combinaciones del grafo. Para estos fines, normalmente se recurren a diversos algoritmos, como la búsqueda en anchura (BFS), el algoritmo de Dijkstra, o el A\*.

A\* es un algoritmo de búsqueda que es muy útil para encontrar un solo camino entre dos nodos de un grafo, a diferencia del de Dijkstra, el cual encuentra varios. Es muy útil en ramas como la visión a computadora o en los videojuegos, donde los agentes inteligentes tienen que moverse de manera autónoma.

Su característica principal es el uso de una función heurística para ir armando el camino entre nodos. Esta función heurística puede ser definida de diferentes maneras, sin embargo, la más común es:

$$h = f + g$$

Donde la función  $f$  representa la distancia euclidiana entre el nodo actual y el nodo objetivo, y la función  $g$  representa la distancia Manhattan (contada entre bloques) entre el nodo actual y el nodo objetivo.

Siguiendo esta norma, realizando una búsqueda en anchura, se analizan los nodos candidatos, y se selecciona el que menor costo cause.

### III. DESARROLLO

El desarrollo de esta práctica se dividió en dos secciones. Primero, el algoritmo PRM, y posteriormente, el algoritmo A\* para los caminos interesantes.

#### PRM

Se decidió hacer un algoritmo de Probabilistic Road Maps usando radios para generar las vecindades. En general, el procedimiento fue el siguiente.

En primer lugar, dividir la imagen o “muestrear” generando un *grid* de líneas horizontales y verticales con distancias no uniformes. Esto es, para seleccionar la posición donde debía caer una línea del *grid*, se generaba un número aleatorio no mayor a las dimensiones de la imagen, y se colocaba en esa coordenada, de manera que la distribución de las líneas fuera completamente al azar. Para agregar incertidumbre a la posición de la línea y hacer que su posición aleatoria fuera más fidedigna, a la coordenada se le agregaba un número aleatorio de complemento, cuyo valor no representaba más del 3% de las dimensiones de la imagen. Así, la función para colocar las filas y las columnas del *grid* quedó de la siguiente manera:

$$posV = random(0, width) + random(0, 3\%(width))$$

$$posH = random(0, height) + random(0, 3\%(height))$$

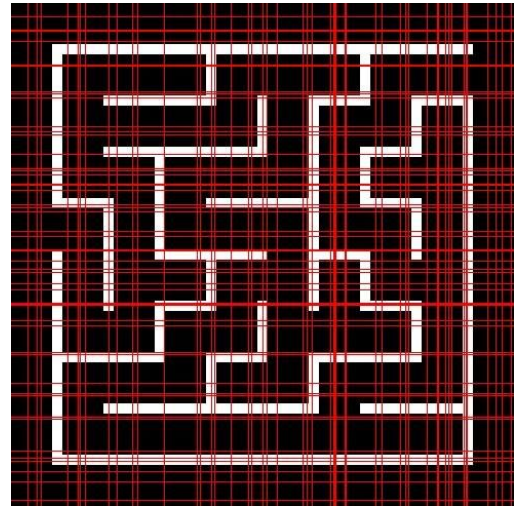


Figura 2. Grid generado. Se puede apreciar la distancia no uniforme entre filas y columnas.

El siguiente paso fue identificar los cruces que caían en el espacio de  $C_{free}$  de la imagen, con el fin de descartar los cruces en obstáculos. Así los cruces restantes serían considerados como nodos al iniciar la generación del grafo.

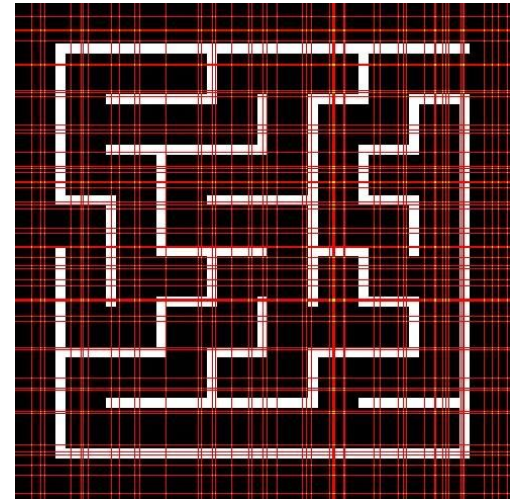


Figura 3. Se pueden apreciar los nodos seleccionados en amarillo, posteriormente se convertirán en los nodos del grafo.

Una vez se tiene el conjunto de nodos generado, se puede proceder a crear el grafo. En esta práctica se utilizó un radio de 70 píxeles como parámetro de conexión entre nodos. Después de experimentación a prueba y error, se observó que un radio de 70 píxeles genera un grafo lo suficientemente conexo en la mayoría de los casos. Además, en este paso se verificaba, naturalmente, que el arista entre los dos nodos no tocara ningún punto de  $C_{obs}$ .

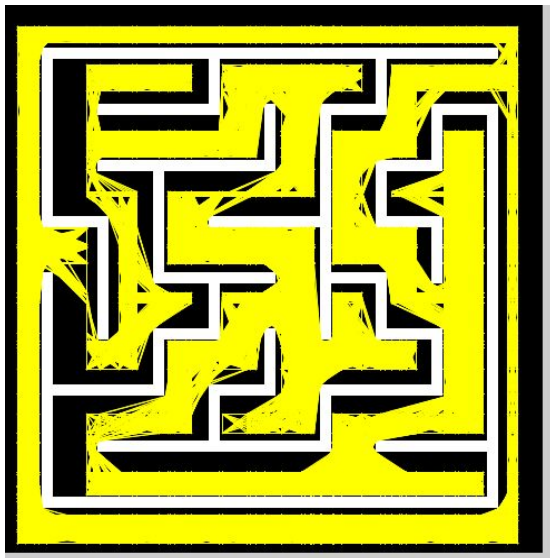


Figura 4. En amarillo, el grafo ya conexo. Se puede apreciar que en este caso existe un camino posible para todos los puntos de la imagen.

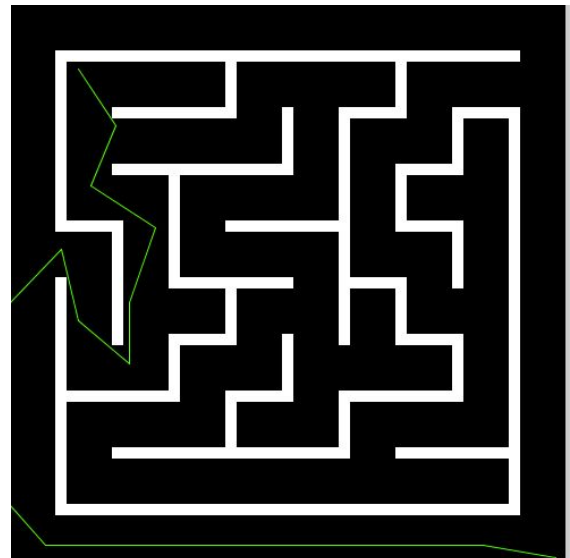


Figura 5. Camino elegido por el algoritmo A\*. El nodo de inicio es en la parte superior izquierda, el nodo final es en la parte inferior derecha.

#### A\*

Tener el grafo ya armado nos da las herramientas necesarias para buscar un camino entre dos puntos de la imagen. Para esto, se desarrolló un algoritmo de A\*.

Seleccionando dos nodos del grafo, uno de inicio y uno de fin, se procede a realizar el algoritmo.

El primer ejercicio fue, usando los criterios de Nodo inicio y Nodo fin, realizar una búsqueda en anchura que le informara a cada nodo del grafo cuál era su Nodo padre, esto mediante un atributo llamado *cameFrom*.

Posteriormente, se realiza propiamente el algoritmo A\* con una profundidad de 1, puesto que solo se analizaron como candidatos los vecinos inmediatos del nodo estudiado. Por cada nodo se calcula su heurística, y se selecciona el que causa menor costo. Se agrega ese nodo al camino y se itera hasta llegar al objetivo.

#### IV. RESULTADOS

A continuación se presentan los resultados de aplicar el algoritmo a una imagen satelital del Tecnológico de Monterrey Campus Puebla, ya segmentada.



Figura 6. Grafo resultante tras aplicar PRM a la imagen.

Como se puede observar, el grafo que resulta tras aplicar el algoritmo sobre la imagen satelital es lo suficientemente conexo como para averiguar caminos entre diversos lugares del tecnológico.



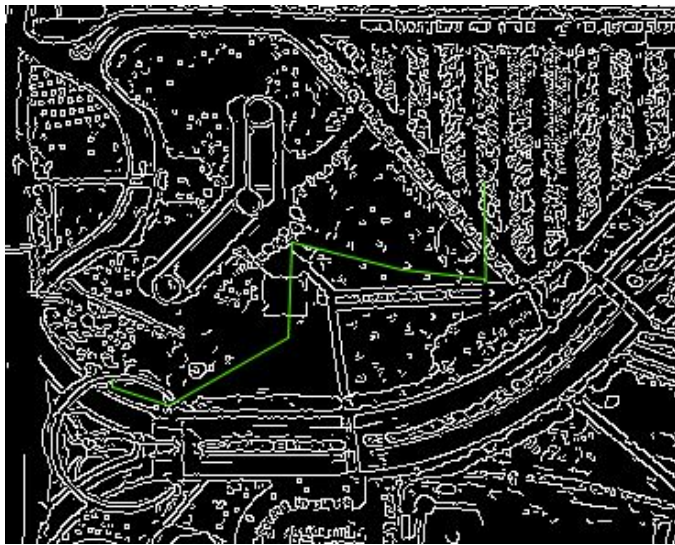


Figura 7. De la entrada del campus al estacionamiento de preparatoria.

Como se observa en la Figura 7, el algoritmo encontró un muy buen camino para llegar al estacionamiento de preparatoria desde la entrada del campus. Esto es muy útil para, por ejemplo, un alumno que salió a fumar un cigarro a la entrada y desea saber un camino para llegar hasta su coche en el estacionamiento 3. Como se puede notar, se corta por los jardines aledaños a la preparatoria y finalmente llega a su destino.

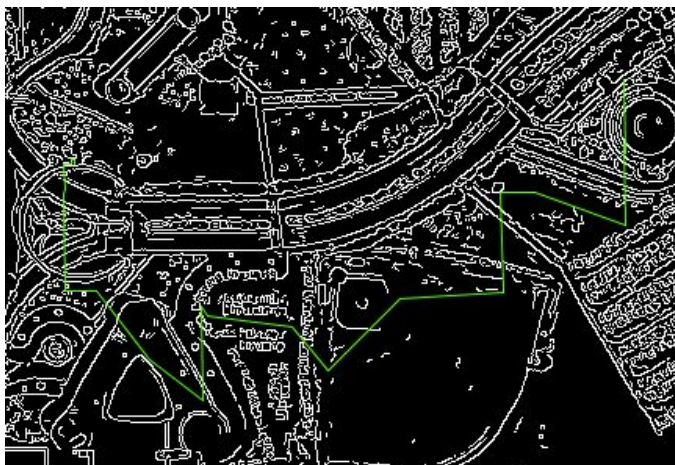


Figura 2. De la entrada del campus a la cafetería.

Aquí se puede apreciar un área de oportunidad del algoritmo, ya que aunque los nodos logran evadir ciertos obstáculos en el camino, hay algunos puntos donde sí cruzan líneas blancas. Esto se debe a que el chequeo de colisión es muy básico y solo considera pixel contra pixel al momento de generar el grafo, y la función que dibuja líneas de Processing (software utilizado para esta práctica) las dibuja con difuminado en 2 pixeles aledaños a la línea, por lo cual en la imagen puede aparentar que impacta, sin embargo, es un hecho que hay partes del

grafo que sí cruza, en una imagen con mucho ruido como esta, y donde el espacio de obstáculos tiene un grueso en promedio de un solo pixel. Ya que como se pudo notar en el desarrollo, en una imagen más sencilla no ocurre.

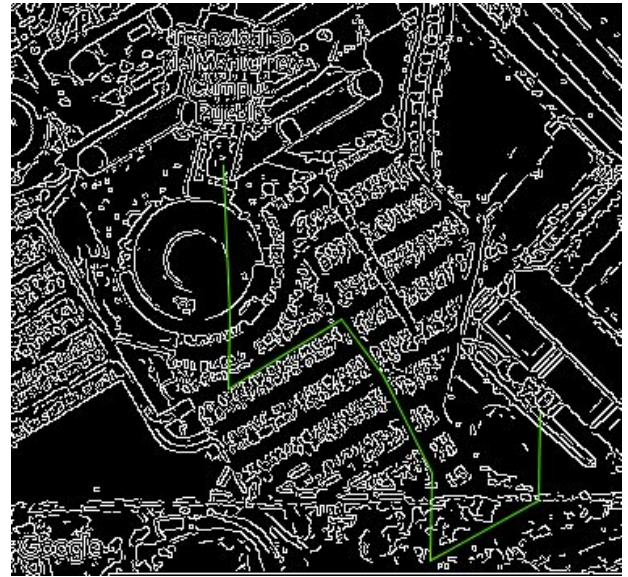


Figura 8. De Aulas 1 y Biblioteca al gimnasio 1.

Aquí se puede apreciar muy claramente cómo el algoritmo pasa por los caminos delimitados por el estacionamiento de atrás de biblioteca para llegar al gimnasio 1. En la vida real, este camino no sería válido porque como se puede notar el algoritmo pasa por los terreno que ya están fuera del Instituto. Sin embargo, esto no lo sabe nuestro algoritmo.

## V. CONCLUSIÓN

Por medio de esta práctica se pudo comprender mejor los temas relacionados a la generación de espacios de configuración para robots, así como la generación de caminos posibles en un grafo de búsqueda. Con la ayuda de algoritmos como el PRM, y A\*, se pudo simplificar un problema que es complejo. Esto ayuda ya que provee a los agentes la información necesaria para trasladarse en su espacio. También es cierto que mejorar el grafo al generar más nodos perfeccionaría el trabajo, sin embargo tendría un costo computacional muy alto, y en robots específicamente hay que encontrar una justa medida entre velocidad y conocimiento del medio. De esta manera, se puede concluir que este tipo de algoritmos dan una respuesta buena al problema, pero son los parámetros que se eligen los cuales delimitan la calidad del mismo.

## VI. REFERENCIAS

- [1] Carballo, W. (2017). *Planeación y seguimiento de caminos para robots móviles en dos dimensiones*.

28/Dic/2017, de ITESM Campus Puebla. Presentación de Power Point.

[2] N.A. (2014). *Introduction to A\**. 28/Dic/2017, de Red Blob Games Sitio web: <https://www.redblobgames.com/pathfinding/a-star/introduction.html>