

TSRMI: Assignment #9

Luis Alberto Ballado Aradias

luis.ballado@cinvestav.mx

CINVESTAV UNIDAD TAMAULIPAS — 4 de julio de 2023

Potencialmente, el número de aristas que pueden aparecer en un grafo de visibilidad es n^2 , con n el número total de vértices en el medio ambiente considerado. Para reducir el número de aristas (y reducir el espacio en memoria requerido para el almacenamiento del grafo), sin perder la solución óptima, se propuso una construcción alternativa denominada grafo de cotangentes o grafo-T.

Documentar algún desarrollo científico que emplee esta representación.

Un artículo científico que utiliza el algoritmo TangentBug en robótica móvil es **Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles** por Yun Hui Liu y Suguru Arimoto. El artículo propone la aplicación del algoritmo TangentBug para la navegación autónoma de robots móviles en entornos desconocidos. Se describe cómo el algoritmo utiliza sensores de distancia para seguir el perfil de un obstáculo y encontrar una ruta alrededor de él.

- ¿Cuál es el principio de construcción de esta representación?

El principio de construcción de la representación utilizando el algoritmo TangentBug se puede describir de la siguiente manera:

1. Inicialización: El robot se coloca en una posición inicial y se establecen los parámetros iniciales del algoritmo, como la velocidad de movimiento y la distancia de seguimiento al contorno del obstáculo.
2. Detección de obstáculos: El robot utiliza sus sensores de distancia para detectar los obstáculos presentes en su entorno. Estos sensores pueden ser láseres, ultrasonidos u otros dispositivos que miden la distancia desde el robot hasta los objetos cercanos.
3. Seguimiento del contorno: Una vez que se detecta un obstáculo, el robot utiliza el algoritmo TangentBug para seguir el contorno del obstáculo. El algoritmo calcula la dirección tangente al contorno del obstáculo en función de las mediciones de distancia y la posición relativa del robot con respecto al obstáculo.
4. Evitación de obstáculos: Si el robot se acerca demasiado al obstáculo, el algoritmo TangentBug realiza maniobras para evitar colisiones. Esto puede implicar cambiar la dirección de movimiento, disminuir la velocidad o detenerse temporalmente hasta que se aleje lo suficiente del obstáculo.
5. Exploración y mapeo: A medida que el robot sigue el contorno de los obstáculos, también realiza una exploración del entorno y crea un mapa del área. Esto se logra registrando la posición y orientación del robot a medida que se mueve y utilizando esta información para construir una representación del entorno.

El algoritmo TangentBug permite que el robot navegue de manera autónoma en entornos desconocidos, evitando obstáculos y siguiendo el contorno de los mismos. Esto facilita la exploración y el mapeo de entornos complejos sin la necesidad de tener un conocimiento previo del entorno.

- ¿Cuál es el algoritmo empleado para su construcción?

El algoritmo TangentBug es una combinación de movimientos de seguimiento de contorno y movimientos de evasión de obstáculos, utiliza la información de los sensores de distancia para detectar

y seguir el contorno de los obstáculos, ajustando la dirección y velocidad de movimiento del robot en función de las mediciones y la posición relativa al obstáculo. Además, incorpora maniobras de evasión para evitar colisiones cuando la distancia al obstáculo es menor que un umbral establecido.

1. Inicialización

- Establecer la posición y orientación inicial del robot.
- Establecer la velocidad de movimiento y la distancia de seguimiento al contorno del obstáculo.
- Obtener las mediciones iniciales de los sensores de distancia

2. Mientras no se alcance la posición objetivo

a) Detección de obstáculos

- Obtener las mediciones actuales de los sensores de distancia.
- Determinar si hay obstáculos presentes y su ubicación relativa al robot.

b) Seguimiento del contorno

- Calcular la dirección tangente al contorno del obstáculo utilizando las mediciones de distancia y la posición relativa del robot.
- Ajustar la velocidad y la dirección de movimiento del robot para seguir el contorno.

c) Evitación de obstáculos

- Si la distancia al obstáculo es menor que un umbral predefinido, realizar una maniobra de evasión.
- Cambiar la dirección de movimiento o detenerse temporalmente hasta que se aleje lo suficiente del obstáculo.

d) Actualizar la posición y orientación del robot en función de la velocidad y dirección de movimiento.

3. Repetir los pasos en el punto 2 hasta alcanzar la posición objetivo