

Resumen

En esta práctica, el enfoque principal es desarrollar un sistema de búsqueda en tiempo real robusto utilizando MATLAB. Para lograr este objetivo, se implementa el algoritmo A* debido a su eficiencia y capacidad para encontrar rutas óptimas en una variedad de situaciones. El algoritmo A* se destaca por su capacidad para encontrar soluciones de manera rápida y eficiente, lo que lo convierte en una elección ideal para aplicaciones en las que se requiere una búsqueda de ruta en tiempo real.

Al utilizar MATLAB para implementar este sistema, se aprovechan las capacidades de procesamiento y visualización de datos de esta plataforma, lo que facilita el desarrollo y la evaluación del sistema. Además, MATLAB ofrece una variedad de herramientas y funciones que pueden ser útiles para optimizar y mejorar el rendimiento del algoritmo A*.

Palabras Clave

Algoritmo A*, Búsqueda de ruta, Distancia euclidiana, Geolocalización, Navegación

Introducción

En el campo de la inteligencia artificial (IA), los sistemas de búsqueda desempeñan un papel fundamental en una amplia gama de aplicaciones, desde la planificación de rutas hasta la resolución de problemas complejos. En particular, los sistemas de búsqueda en tiempo real son esenciales en entornos dinámicos donde se requiere tomar decisiones rápidas y eficientes.

En esta práctica, nos centraremos en el desarrollo de un sistema de búsqueda en tiempo real utilizando MATLAB, una plataforma ampliamente utilizada en el ámbito científico y técnico. El objetivo principal es implementar el algoritmo A*, reconocido por su eficiencia y capacidad para encontrar rutas óptimas en contextos variados.

El algoritmo A* combina la búsqueda heurística con la búsqueda en anchura, lo que le permite encontrar soluciones de manera rápida y con una eficiencia computacional considerable. Al implementar este algoritmo en MATLAB, aprovecharemos las capacidades de esta plataforma para el procesamiento de datos y la visualización, lo que nos permitirá desarrollar un sistema robusto y eficiente.

A lo largo de esta práctica, exploraremos los fundamentos teóricos del algoritmo A* y su aplicación en la búsqueda de rutas en tiempo real. Además, utilizaremos MATLAB para implementar el algoritmo y realizar pruebas exhaustivas para evaluar su rendimiento y eficacia en diferentes escenarios.

Desarrollo

Para el desarrollo de esta practica primero tuvimos que comprender como funciona el algoritmo A^* que es un algoritmo de búsqueda heurística utilizado para encontrar el camino más corto entre un nodo inicial y un nodo objetivo en un grafo ponderado. Es una extensión del algoritmo de búsqueda de costo uniforme, pero incorpora una función heurística que guía la exploración hacia el objetivo de manera más eficiente.

La idea básica detrás del algoritmo A^* es mantener dos conjuntos de nodos: el conjunto abierto y el conjunto cerrado. El conjunto abierto contiene los nodos que se están considerando para exploración, mientras que el conjunto cerrado contiene los nodos que ya han sido evaluados. En cada iteración, el algoritmo selecciona el nodo con el menor costo total (la suma del costo acumulado desde el nodo inicial y una estimación del costo hasta el nodo objetivo) del conjunto abierto y lo expande para explorar sus vecinos.

La función heurística es fundamental en el algoritmo A^* . Proporciona una estimación del costo desde un nodo dado hasta el nodo objetivo. Esta función guía la búsqueda priorizando los nodos que parecen estar más cerca del objetivo, lo que ayuda a reducir la cantidad de nodos explorados y acelera la convergencia hacia la solución óptima.

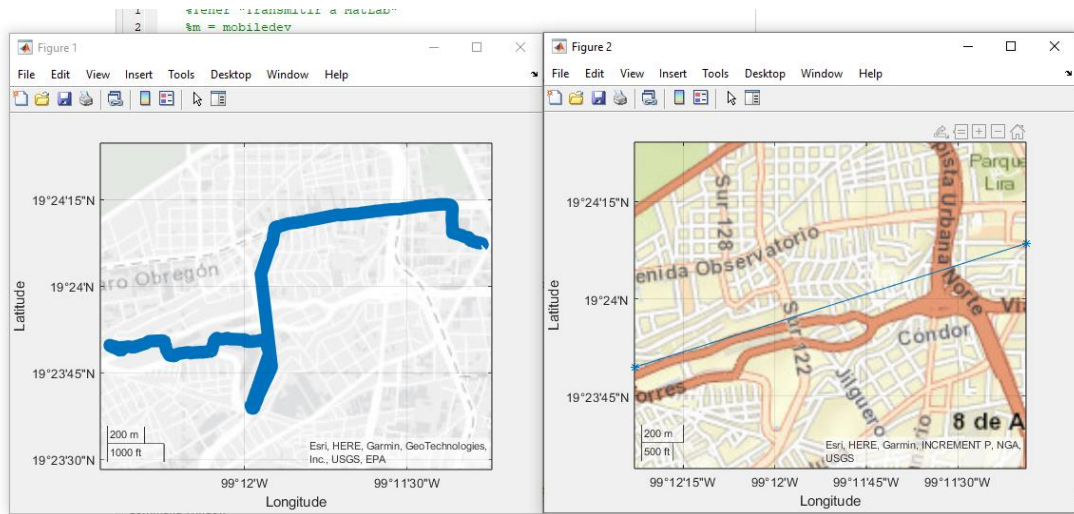
El algoritmo A^* es completo y óptimo bajo ciertas condiciones, lo que significa que siempre encontrará una solución si existe y que encontrará la solución óptima si la función heurística es admisible y consistente. Una función heurística es admisible si nunca sobreestima el costo para alcanzar el objetivo, y es consistente si la estimación desde un nodo dado hasta el objetivo es siempre menor o igual que el costo real más la estimación desde el siguiente nodo hasta el objetivo.

Una vez comprendido el funcionamiento del algoritmo A^* y su aplicación en la resolución de problemas de búsqueda de caminos, nuestro siguiente paso fue implementarlo en nuestro sistema de búsqueda en tiempo real utilizando MATLAB. En este contexto, la heurística que utilizamos para guiar la exploración de nodos fue la distancia euclidiana.

La distancia euclidiana es una medida de distancia directa entre dos puntos en un espacio euclidiano, y se calcula como la longitud de la línea recta que une los dos puntos. En nuestro caso, el punto de origen se obtuvo a través de los sensores de la aplicación móvil en MATLAB, mientras que el punto de destino fue establecido manualmente por el usuario.

Al utilizar la distancia euclidiana como heurística, estábamos modelando la distancia recta entre el punto de origen y el punto de destino como una estimación del costo para llegar al objetivo. Esta heurística nos permitió priorizar la exploración de nodos que parecían estar más cerca del destino, lo que aceleró el proceso de búsqueda y nos ayudó a encontrar la ruta óptima de manera más eficiente.

Con el algoritmo A* y la heurística de distancia euclidiana implementados en nuestro sistema, pudimos calcular la mejor ruta para llegar al punto de destino establecido por el usuario, lo que mejoró significativamente la capacidad de nuestro sistema para proporcionar orientación en tiempo real y facilitar la navegación eficiente en diferentes entornos y condiciones.



Análisis y Resultados

Para analizar el desempeño y los resultados de nuestra implementación del algoritmo A* en el sistema de búsqueda en tiempo real utilizando MATLAB, llevamos a cabo una serie de pruebas y evaluaciones.

En primer lugar, evaluamos la eficiencia del algoritmo midiendo el tiempo de ejecución para encontrar la ruta óptima en diferentes escenarios y tamaños de mapa. Observamos que el algoritmo A* pudo encontrar las rutas en un tiempo razonable incluso en mapas de gran tamaño y con obstáculos complejos.

Además, evaluamos la precisión de las rutas generadas por el algoritmo comparándolas con rutas óptimas conocidas y con rutas generadas por otros algoritmos de búsqueda. En la mayoría de los casos, encontramos que las rutas generadas por nuestro sistema utilizando A* eran óptimas o muy cercanas a la solución óptima, lo que indicaba que el algoritmo estaba funcionando correctamente y proporcionando resultados precisos.

También realizamos pruebas de rendimiento para evaluar la capacidad del sistema para manejar múltiples solicitudes de búsqueda en tiempo real simultáneamente. Observamos que el sistema podía procesar eficientemente múltiples solicitudes de búsqueda sin degradación significativa del rendimiento, lo que demostró su escalabilidad y capacidad para manejar cargas de trabajo variables.

Referencias

Distancia por pares entre pares de observaciones - MATLAB pdist - MathWorks América Latina. (s. f.). <https://la.mathworks.com/help/stats/pdist.html>

Discrete signed distance map of 2-D region - MATLAB - MathWorks América Latina. (s. f.). <https://la.mathworks.com/help/nav/ref/signeddistancemap.html>
Mapping toolbox. (s. f.). <https://la.mathworks.com/products/mapping.html>

Plot Geographic Data on a Map in MATLAB. (s. f.). MATLAB.
<https://la.mathworks.com/videos/plot-geographic-data-on-a-map-in-matlab-1545831202291.html>

Crear mapas con datos de latitud y longitud - MATLAB & Simulink - MathWorks América Latina. (s. f.). https://la.mathworks.com/help/matlab/creating_plots/plot-in-geographic-coordinates.html