Módulo de enrutamiento y guiado para un sistema integral de guiado y localización de objetivos en interiores para discapacitados visuales basados en dispositivos móviles libres.

Autora: Rocío Rueda Villanúa

Abstract- La persona y el espacio se relacionan e interactúan permanentemente. En esta relación persona-espacio, la capacidad de desplazamiento se puede reducir por insuficiencia de la persona para acceder y comunicarse, o por impedimentos del espacio para posibilitar el acceso a la comunicación. Un avance en la accesibilidad para personas con limitaciones sensoriales como es la visión, además de la necesidad de que los elementos arquitectónicos estén libres de obstáculos, es asistirle en la realización de la ruta que tiene que realizar la persona. Un sistema de guiado y enrutamiento ha de realizarse del modo que sea lo más sencillo al usuario y se adapte siempre a sus necesidades. Se propone un sistema de navegación: simple, asequible y no intrusivo para disminuidos visuales que les permita llevar a cabo una actividad cotidiana en recintos interiores (supermercados, centro comercial etc.) de forma autónoma e independiente. Este proyecto estudia diferentes alternativas para el cálculo de rutas, seleccionando la más óptima, en función de un análisis de las preferencias de este tipo de usuario y también desarrolla la estructura empleada en la etapa de guiado al usuario.

INTRODUCCIÓN

Acciones como ir de compras a un supermercado, buscar un objeto en un almacén, o en cualquier entorno desconocido para el usuario pero controlado, no deberían significar un inconveniente. Pretendemos plantear un sistema para guiar y dar indicaciones a estos usuarios, y, en la medida de lo posible, a encontrar lo que necesitan de forma independiente, como un paso más hacia la eliminación de las barreras en la accesibilidad. Con este proyecto pretendemos ayudar a las personas con discapacidad visual a desenvolverse en un entorno cerrado y desconocido.

Las personas que disponen del sentido de la vista no son realmente conscientes de las limitaciones a las que se ven sujetos aquellos que sufren algún tipo de discapacidad visual, un factor que disminuye enormemente las posibilidades de moverse libremente y de forma independiente. Tareas cotidianas como desenvolverse en un edificio desconocido (hospital biblioteca, etc.) o simplemente ir de compras se antojan complejas o imposibles sin asistencia de terceros, impidiéndoles por tanto una vida libre autónoma y plena. Esta libertad se convierte en el objetivo diario de estas personas muchas veces aisladas del exterior y dependientes de otras personas. Ir de comprar es una actividad casi diaria aporta a los usuarios autonomía para el consumo libre sin la dependencia de otras personas.

El interés de este proyecto radica en la posibilidad de ofrecer un sistema de navegación acorde con: las preferencias de las personas con capacidades visuales disminuidas, asequible a sus

economías y mínimamente intrusivo con el entorno donde se implante, que dé un paso más hacia la integración de las personas con incapacidad visual.

En los artículos publicados hasta ahora en referencia a este tema, realizan caso omiso a la situación en la que se encuentran estas personas, contando con un cálculo de rutas basado en la minimización de la distancia recorrida por el usuario.

Es el caso de Proyectos como el Proyecto Enabled [1], y otros proyectos como [2] y [5] en el que se desarrollan un prototipos de navegación tanto para exterior como para interior para personas con discapacidad visual, en el que se habla del cálculo de rutas empleando mapas inteligentes, pero en el mejor de los casos se centra en la obtención de la ruta más corta, en otros casos no mencionan para nada cual es el algoritmo empleado.

Pero no siempre la minimización de la distancia se trata de la solución óptima para este tipo de usuarios, dado que existen otros factores que resultan más importantes. Dentro de los problemas de movilidad que provoca la pérdida total o parcial de visión se encuentra un factor fundamental denominado desorientación. En muchas ocasiones las personas de deficiencias visuales suelen encontrarse en situación de desorientación muchas veces derivadas de la realización de giros y cambios de sentidos. Es por esto un factor muy importante en el cálculo de ruta para estos usuarios el número de cambios de sentidos que se realicen al establecer la ruta, por lo que consideramos una grave deficiencia que no se tenga en cuenta en proyectos anteriores esta situación.

Dentro de este proyecto de sistema de navegación se identifica el cálculo de ruta como un módulo particular en el que el sistema identificará cual es la ruta óptima en cada caso para el usuario. Definiremos ruta como el conjunto de paso/movimientos que se deben seguir para desde un punto inicial alcanzar un objetivo final.

En el caso de las personas con discapacidad visual, debemos tener en consideración que no cuentan la misma percepción de los sentidos que el resto de personas. Estudios demuestran que cuentan con un mayor desarrollo de los otros sentidos. Por esto es fundamental entender que las preferencias en cuanto a realizar movimientos, suelen ser diferentes que en el caso de personas que cuenten con todos los sentidos por igual.

Por todo esto, el presente proyecto busca realizar, dentro de un sistema de navegación de interiores orientado a personas con discapacidad visual, un estudio sobre el enrutamiento óptimo que no sólo cuente con factores como la distancia, sino que también tenga en consideración otras preferencias de este tipo de usuario, observando y comparando cuales son los factores que influirán en la obtención de rutas óptimas.

Dentro del presente proyecto, estudiaremos en primer lugar cuales son las necesidades y preferencia en el cálculo de ruta del perfil de discapacitados visuales. Conociendo este tipo de preferencias realizaremos un estudio de cuáles son las propuesta que encontramos en la actualidad, y mediante una comparativa observaremos en base a las preferencias de los usuarios cuál sería la solución más óptima.

PRELIMINARES

1. Definición del Problema

El problema que se debe resolver en el presente artículo es la selección de un algoritmo para el enrutamiento de interiores, que obtenga el camino óptimo que debe recorrer la persona con discapacidad visual. A pesar de que el objetivo principal es encontrar la ruta de un lugar a otro en un recinto cerrado, no aceptaremos cualquier camino que cumpla con esta condición, ya que existen una serie de requisitos que deberán ser optimizados para la ruta dado el perfil de los usuarios.

La problemática se agrava teniendo en cuenta que la funcionalidad incluye un guiado en un determinado entorno para alcanzar diferentes hitos, lo que será en este caso definido como guiado del usuario. En nuestro caso, nuestro sistema está orientado al movimiento en entornos lineales, similar al mapa que podemos encontrarnos en mercados, supermercado y centros comerciales. Resulta interesante resaltar este aspecto para su posterior adecuación, ya que el entorno en este tipo de proyectos y para este tipo de usuarios es de vital importancia. El mapa con el contaremos se dividirá en una serie de casillas o celdas de un tamaño específico, en nuestro caso seleccionaremos 50 cm, donde cada una de ella tenga diferentes estados posibles: ocupado, accesible y no accesible, las posiciones ocupadas serán definidas como estante que pueden contener posibles hitos. A continuación se observa un posible mapa genérico de un prototipo de un supermercado donde se encuentran una serie de pasillos con estantes.

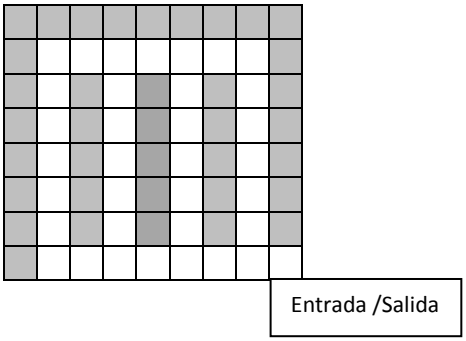


Imagen muestra de un mapa genérico

Otro apartado del proyecto trata sobre el guiado y es la parte de nuestro sistema que permitirá indicar al usuario cual debe ser su recorrido a través del mapa para con un punto inicial y un punto final, alcance entre medios todos los hitos que le sean atribuidos.

2. Algoritmos de Enrutamiento existentes

En el cálculo de rutas a través del mínimo número de giros descrito en el paper [4], se realizan diferentes alternativas al algoritmo enrutamiento para minimizar tanto la distancia como el número de giros, definiendo algunos teoremas interesantes para el presente proyecto. Este proyecto nos sirve como base para los algoritmos que vienen a continuación.

Podemos encontramos diferentes tipos de algoritmos que son susceptibles de ser tenidos en cuenta como solución fiable para nuestro sistema. Estos algoritmos se agrupan en 2 tipos fundamentales, los determinísticos y no determinísticos.

Los algoritmos no determinísticos son aquellos en los que un algoritmo ofrece varios resultados para una misma entrada, y es interesante saber que no se conocerá previamente cual será el resultado. Como algoritmo no determinístico encontramos un algoritmo genético, que significa puede devolver diferentes soluciones a un mismo problema y usa técnicas inspiradas en la biología evolucionaria como herencia, mutación, selección y reproducción.

Como algoritmo genético nos encontramos el proyecto descrito en el paper [3], a partir de ahora GA. En él se muestra una descripción del algoritmo genético que da solución a un enrutamiento en este caso de robots. El caso de los robots guarda cierta similitudes porque como veremos a continuación los algoritmos de enrutamiento de los mismo intentan siempre minimizar el número de giros dado que algunos robots para realizar giros debe pararse en seco y con posterioridad reanudar su marcha, lo cual supone una pérdida de tiempo considerable.

En el caso de algoritmos determinísticos encontraremos una confrontación entre el algoritmo de Dijsktra y el algoritmo A* que se ven definidos ambos a través del paper de un sistema de navegación en interiores orientado a personas discapacitadas visuales [6]. Un algoritmo determinístico es aquel que dada siempre la misma entrada obtiene la misma salida.

PROPUESTA DE SOLUCIÓN

1. Requisitos del usuario

Dentro de la propuesta del presente proyecto tenemos que tener en cuenta que cobra vital importancia conocer cuáles son las necesidades especiales de los usuarios a los que se destina el sistema. Por lo tanto en primer lugar pasaremos a realizar un estudio de preferencias a nivel de movilidad por parte de personas discapacitadas visuales.

Muchas de las preferencias de las personas con discapacidad visual pronunciada dependen en mucha medida de los gustos individuales, sin hacer corresponder esta circunstancia con un patrón preestablecido. Entre estas preferencias encontramos algunas sobre el tipo de terreno. Las personas con discapacidad visual se encuentran más cómodas, en terrenos lisos, sin escalones, sin pendiente y no resbaladizos.

También resulta muy interesante para este tipo de usuario el espacio de maniobra con el que puede contar, siendo deseable espacios amplios, libres de obstáculos. Tomar en consideración estas preferencias, no está al alcance del sistema sino que debe tomarse como un estándar necesario para la adecuación de un entorno susceptible de trabajar con este sistema. Aunque estas preferencias resultan imposibles de controlar, deben especificarse en la etapa de instalación como condiciones de cumplimiento para un óptimo desarrollo de la actividad del sistema.

Tras la consideración de diversos factores en la actualidad se identifica 3 apartados que diferencian el nivel de sencillez que alcanza una ruta para una persona invidente, y que tratan factores definibles en el cálculo de rutas.

En primer lugar cabe destacar que este tipo de personas cuentan con la particularidad de que suelen preferir caminar **cerca de las paredes**, lo cual es una preferencia muy destacable en lo que cálculo de rutas se refiere. Si vamos a trabajar con tiendas y pequeños comercios, cabe destacar que en la gran mayoría podemos localizar una distribución basada en pasillos que facilita la presencia de una superficie vertical cercana a lo largo de todo el recorrido. Aun así en nuestro sistema establecerá un criterio de forma que cuando se vaya a salir a una zona diáfana, la recorra de forma que encuentre una pared de apoyo a menos de 50 cm. Esta distancia permitirá al usuario en una situación de inseguridad o alarma, apoyarse en la pared o superficie vertical más cercana.

Otro de los asuntos fundamentales a la hora de trabajar con personas invidentes es que se debe simplificar la ruta de forma que sea capaz de seguirlas sin perder el rumbo. Para ello, uno

de los requisitos con el que tenemos que contar es que el número de **cambios de sentido** hasta alcanzar nuestros objetivos debe ser mínimo, en todo momento, para de esta forma facilitar la ruta. Con los cambios de sentido se corre el riesgo de provocar a los usuarios desorientación e inestabilidad.

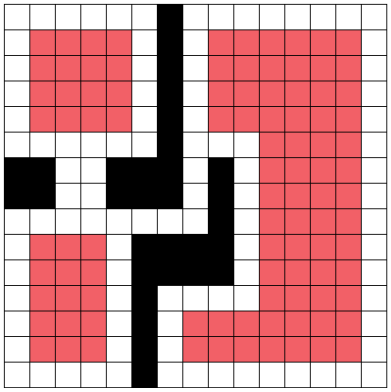
Pero todos estos requisitos no pueden aumentar de forma exagerada la distancia recorrida de la ruta. Por lo que aunque en menor medida dentro de las rutas que reúnan igualdad de condiciones optimización se seleccionará aquella que **minimice la distancia** recorrida

2. Tratamiento de los requisitos del usuario

Dentro de los requisitos definidos en el apartado anterior es el de la **cercanía a las paredes** el que ofrece mayor controversia, no existen antecedentes del tratamiento de este requisito, y en nuestro proyecto al observar que el entorno interior cuenta con una serie de lineales de estanterías que pueden valer de apoyo, puede parecer que carezca de importancia. De todas formas para que nuestra aplicación pueda ser destinada a grandes almacenes, vamos a estudiar como solventar estas circunstancias.

En primer lugar debemos tener claro que nuestro sistema contara con la opción de que el usuario elija si desea activar la opción de cercanía a las paredes o no, ya que puede ocurrir ocasiones en las que el sistema no sea capaz de alcanzar un hito al encontrase en una zona aislada sin pareces ni estantes para su acceso, en ese caso será el usuario el que decidirá si desea alcanzarlo de cualquier o no.

Sabemos que al inicio contamos con un mapa en el cual localizamos las celdas que cuentan con casillas definidas por una distancia de 50 cm, por ello podemos definir que aquellas celdas que no encuentra a un estante o una pared al lado, sea definidas como inaccesibles para que no puedan ser recorridas por el usuario, y contaran para el sistema como celdas ocupadas, a no ser que el usuario indique lo contrario. De esta forma solucionamos el posible conflicto que puede acarrear, de forma que no interfiera en el algoritmo de enrutamiento y que de forma óptima siempre se acceda por zonas que cuenten con un apoyo al usuario. Dado un mapa cualquiera serán definidas como celdas inaccesibles aquellas que en su entorno no encuentren una pared o un estante como muestra las celdas sombreadas en rojo de la siguiente imagen.



Mapa conceptual con indicativo de accesibilidad de las celdas.

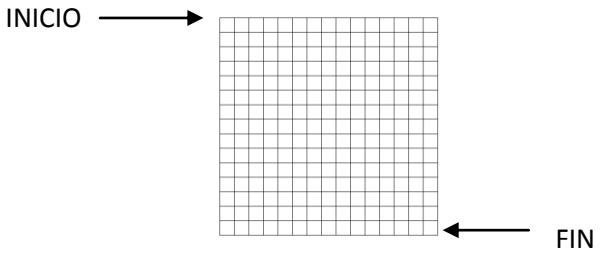
Por lo tanto el recinto contará con 2 mapas almacenados y será el usuario el que indique cual de los 2 va a emplear en el sistema, o bien de forma inicial, indicando si se quiere usar este requisito, o bien de forma dinámica, durante la ejecución del sistema donde se le mostrará un mensaje cuando un hito sea inaccesible sin el incumplimiento de este requisito, donde será el usuario en ese momento quien decidirá que opción tomará.

En nuestro caso, vamos a continuar sin contemplar más la existencia de esta opción ya que no interfiere en los resultados de idoneidad del algoritmo de enrutamiento, y porque en nuestro caso tenemos que analizar recorridos con algoritmos en los que no se ha tenido en cuenta esta opción.

Los otros requisitos de mínimo número de cambios de sentidos y mínima distancia, serán tratados de forma paralela mediante los algoritmos de enrutamiento, aunque es interesante dar a conocer cuál va a ser la prioridad y la importancia que se le va a atribuir a cada uno de los requisitos. En nuestro caso para un sistema de las conocidas características queda demostrado que el problema que puede ocasionar un caso de desorientación en el usuario puede ser un impedimento bastante grande para alcanzar los objetivos que se pretende. Por ello como prioridad fundamental trataremos la minimización de los cambios de sentido, y en segundo lugar buscaremos minimizar la distancia total del camino, para un equivalente número de giros.

3. Funcionamiento de los algoritmos de enrutamiento

Vamos a trabajar con mapas de 15 X 15 casillas lo que hace un total 7,5m X 7,5m de área, para los que trataremos de simplificar el problema creando rutas desde esquina superior izquierda, a la esquina inferior derecha.



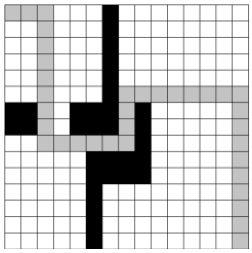
Con color gris señalaremos el camino que se propone por parte del algoritmo, y en negro se visualizan las casillas que son consideradas como inaccesibles por el usuario, en función de su preferencia inicial, en este caso sólo aparecerán los estantes o columnas.

a. Genético

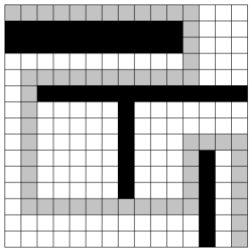
En primer lugar, vamos a ver el funcionamiento del algoritmo **genético** propuesto en el paper [3] (GA) que en este caso busca no sólo el camino óptimo en longitud sino también en giros. Para analizar los resultados de un algoritmo no determinista, como son los genéticos, debemos tener en consideración un que se trata de algoritmos cuyo resultados pueden cambiar en diferentes ejecuciones para una entrada estable. Es por ello que debemos tener en cuenta que los resultados obtenidos se representan en función del tanto por ciento de éxito tras un total de 15 ejecuciones

Si observamos los resultados obtenidos el algoritmo genérico distan bastante de una solución óptima para nuestro sistema dado que aunque intenta reducir el número de giros, ya que así lo indica, se encuentra lejos de minimizarlo por completo, y existen ejecuciones en las que no alcanzan el éxito buscado.

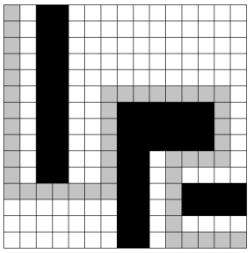
Entorno	**Porcentaje de Éxito**
1	100
2	86,7
3	73,7



Entorno 1



Entorno 2



Entorno 3

b. Dijkstra

El **algoritmo de Dijkstra**, también conocido como algoritmo de caminos mínimos es un algoritmo para la determinación de la ruta mínima de un origen a un destino. Este algoritmo está diseñado para trabajar con grafos y encuentra el camino más corto entre un punto (vértice) al resto de puntos en un grafo dirigido donde el peso de cada arista representa el coste de recorrerla.

Pero para trabajar con este algoritmo debemos realizar una modificación de este ya que nuestro entorno está definido por una matriz. Para ello deberemos transformar el mapa en un grafo y así poder trabajar con Dijkstra. Obtendremos de la ejecución un camino en el grafo que deberá ser convertido a una ruta en el mapa.

Para esta transformación nos ayudamos almacenando la orientación y el número de cambios de sentidos en cada posición del mapa, así construimos un grafo donde hay 4 vértices por cada celda del mapa, uno por cada punto cardinal. Un vértice o nodo tendrá por vecino a los nodos a los que se puede llegar a través de algún movimiento. Por lo tanto un nodo vendrá definido por 2 coordenadas y la orientación, por ejemplo [x0, y1, E].

Por ejemplo: Los 3 vecinos posibles de [x0, y0, E]

[x0, y0, E] → es vecino de → [x0, y0+1, E]

[x0, y0, E]→ es vecino de → [x0, y0, N]

[x0, y0, E] →es vecino de→ [x0, y0, S]

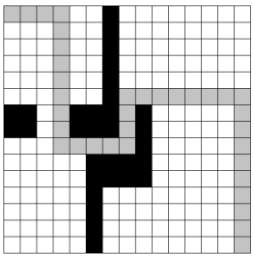
La idea de esta transformación es la división en capas del grafo de forma que cada nodo tiene como vecino un nodo con la misma orientación, así como nodos de otra capa con diferente orientación pero las mismas coordenadas.

Dentro de nuestro mapa tendremos también posiciones ocupadas donde no se permitirá el transito, para ello se eliminaran las aristas que llegan a estos nodos inaccesibles, asegurando así que el algoritmo no encuentre caminos no realizables, excluyéndolos de cualquier posible ruta. Teniendo ya el grafo a través del mapa, ejecutamos Dijkstra para saber cuáles son los caminos mínimos. El análisis de los resultados obtenidos es bastante sencillo dado que la ruta viene definida por los vértices seleccionados del grafo, por lo que el concepto de mapear o representar el grafo en el mapa, es bastante simple.

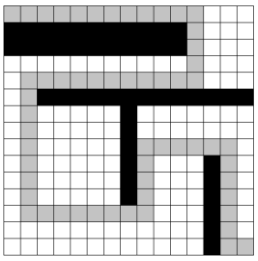
Los resultados pueden ser analizados en función del éxito obtenido, donde simplemente contaremos con la opción de que para un determinado mapa funcionan o no, sin detenernos

en hacer más ejecuciones dado que siempre obtendrán el mismo resultado. En el caso de Dijkstra el éxito es absoluto, pero para poder realizar comparaciones posteriores vamos a centrarnos en el tiempo de ejecución ya que en estos casos suele ser un factor fundamental, siempre y cuando el entorno, en este caso el sistema donde sean ejecutados sea el mismo. Dado que este algoritmo realiza multitud de operaciones, el tiempo de retardo suele ser fundamental en estas situaciones, porque aunque las áreas de trabajo suelen ser bastante mayores, nos vale para hacernos una idea proporcionalmente.

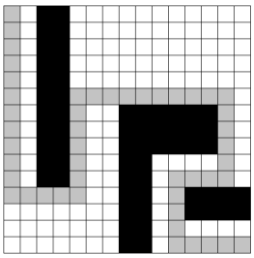
Entorno	Tiempo Ejecución
1	200 ms
2	198 ms
3	199 ms



Entorno 1



Entorno 2



Entorno 3

c. A*

Este algoritmo, denominado también A estrella, se define también como un algoritmo de búsqueda en grafos, y es básicamente una optimización heurísticas del anterior. Se trata de un algoritmo mejorado que encuentra, siempre y cuando se cumplan unas determinadas condiciones, el camino de menor coste entre el inicio y el fin.

Este algoritmo A* deriva de una función de evaluación:

$$f(n) = g(n) + h(n)$$

h(n): representa el valor heurístico a evaluar

g(n): el costo real de la ruta recorrida para llegar a un nodo o vértice.

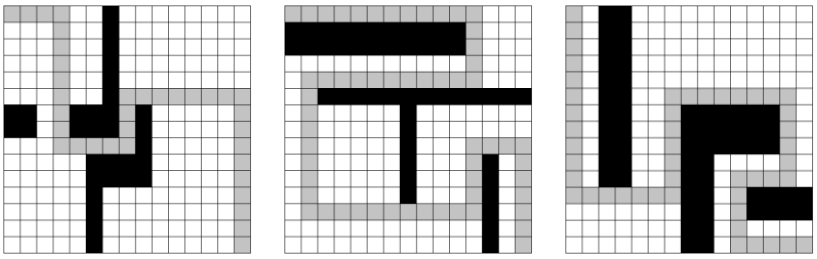
A* cuenta con 2 estructuras de datos: por un lado los abiertos, implementando así una cola de prioridad, cuyo orden va en función del valor de f(n) en el nodo, y por otro lado los cerrados donde se guarda que nodos han sido visitados. En cada paso del proceso, se expande el nodo que esté primero en abierto, y si no es el hito, calcula la f(n) de los hijos, y así los inserta como abiertos y pasa él ha estar cerrado.

Tal y como se observa este algoritmo es una combinación de diferentes métodos de búsqueda como el de búsqueda en anchura BFS, hacia el que tiende el coste real (g(n)), y el de búsqueda en profundidad hacia el que tiende el valor heurístico ((h(n)). Pero sin duda A* tiene por definición que si el problema tiene solución, él la encuentra.

En el caso particular de este proyecto se sigue la misma estrategia de conversión a grafo que en el apartado anterior, por lo que se arma el grafo a partir del mapa y hacemos correr la ejecución del programa en él. Posteriormente debemos realizar la operación inversa de representar el camino elegido, en el mapa.

Al algoritmo A* le tenemos que indicar una función heurística h(n) para mejorar el desempeño del mismo en el momento de buscar un camino que resuelva el grafo. Lo que indica la función h(n) es el nivel de preferencia del nodo n que se le ingresa. Para este algoritmo h(n) devuelve la distancia del nodo n al nodo final. Esa distancia se calcula de acuerdo a la cantidad de celdas horizontales más verticales que distancian la posición que representa el nodo n con la celda que representa el nodo final. Finalmente es sencillo observar el camino que recorre por el grafo y convertirlo a la matriz del mapa. Los resultados son similares al apartado anterior.

Entorno	Tiempo Ejecución
1	143 ms
2	208 ms
3	153 ms



Entorno 1 Entorno 2 Entorno 3

Las rutas obtenidas son diferentes a las encontradas en el apartado anterior debido a la función heurística h que elije los nodos a visitar según cercanía con el punto final. El algoritmo obtiene al igual que Dijkstra resultados óptimos tanto de número de giros como de distancias, pero con una mejora mínima en tiempos de ejecución en algunos casos.

4. Comparativas de Algoritmos

En este apartado vamos a realizar una comparativa sobre los 3 algoritmos que describimos con anterioridad. Lo que vamos a hacer es definir unos factores y analizarlos en la ejecución de cada uno de los algoritmos:

a. **Previsible:** Para que un algoritmo bien definido debe obtener la misma salida para una misma entrada, esto no ocurre en los algoritmos genéticos por lo que lógicamente se encuentra en clara desventaja frente a los 2 últimos. Un algoritmo debe ser previsible y estar bien definido para que aporte estabilidad a un sistema, en este caso un algoritmo cambiante no aporta ninguna ventaja, y puede ocasionar algunos problemas, al no controlar todas las alternativas.

b. **Preciso:** La precisión de un algoritmo viene dada por el éxito que es capaz de alcanzar, en este caso, el tanto por ciento de éxito del algoritmo de Dijkstra y de A* es de un 100%, mientras que en el caso del algoritmo genético no siempre es del 100%.

c. **Tiempo:** En este caso es interesante para un sistema autónomo y dinámico que su tiempo de reacción sea lo más corto posible. En el caso del algoritmo genético no se ha contemplado ya que durante las 15 ejecuciones este ha sido completamente inestable, llegando incluso en duplicar y triplicar su tiempo de una ejecución a otra. Para los algoritmos A* y Dijkstra cuenta con tiempo similares con una clara tendencia a que en la mayor parte de entorno A* es más rápido ya que incluye el extra de ir buscando el punto final.

5. Discursión de Algoritmos

Tras la comparativa llegamos a la conclusión de que un algoritmo genético haría inviable nuestro sistema, por lo que debemos centrarnos en el de Dijsktra y el de A*. En este caso la elección parece también simple viendo la comparativa anterior, ya que a la larga A* ofrecerá mejores resultados en cuanto a tiempo de ejecución. Por lo tanto tras el análisis anterior no es de descartar que el algoritmo utilizado va a ser A* o A estrella.

6. Guiado

Para el guiado del usuario describiremos las acciones que debe contemplar el sistema desde que el usuario entra en el recinto interior hasta que sale de él. En primer lugar, el individuo entra nuestro entorno por un punto determinado que será a su vez punto de salida. El proceso de guiado es un simple algoritmo que permite dividir en diferentes acciones, que se repetirán de forma iterativa:

- Origen. El origen en la primera parte corresponderá al punto de entrada, mientras que en las sucesivas etapas verificaremos como origen, su posicionamiento actual, aportado por el módulo de posicionamiento.

- Elegir Hito. En la etapa de elección de hitos nos basaremos en diversos factores para una selección del hito óptima, siempre conociendo cual es nuestro punto de origen.

- Ruta hacia el Hito. Una vez que hemos seleccionamos el hito, nos basamos en su ruta óptima para mediante indicaciones alcanzarlo. Una vez llegamos al hito comentaremos al módulo de localización que hemos llegado frente al objetivo.

Estos módulos se repetirán por cada uno de los hitos. Una vez alcanzado el último hito, el sistema debe calcular su origen e indicar la ruta hacia la salida, en realidad lo que se le indica al sistema es que la salida, debe ser el último hito alcanzable.

Para la implantación de este sistema debemos definir los siguientes puntos donde localizamos lo que se define a continuación:

Punto 1 Definición del MAPA: En primer lugar debemos definir el mapa, que es el primer paso en la instalación del sistema en el deberá realizarse un análisis de cuáles son las celdas, y definir así cuales estarán disponibles y cuáles no serán accesible por el usuario. Este proceso pertenece a la parte de instalación del sistema, pero entra en juego cada vez que un usuario accede al recinto, dado que deberá seleccionar si desea o no, contar con la asistencia de proximidad a las paredes.

Punto 2 Localización de HITOS en el Mapa: Es un punto fundamental en el inicio del sistema, en este caso, una vez definido el mapa, pasamos a seleccionar la localización de cada uno de los hitos o productos, que se encuentra en el entorno, indicando su situación en la estantería.

Punto 3 Selección de Hitos: Una vez localizados los hitos o productos en el mapa, entra en acción el usuario que realizará 2 elecciones. Por un lado describirá los hitos que quiera

alcanzar, y por otro lado indicará si desea contar con el asistente que evite zonas aisladas o sin paredes o estantes cercanos.

Punto 4.1 Selección del Próximo hito: Una vez conozcamos todos los hitos pasamos a recalcular cual será el próximo hito, mediante el cálculo de la ruta, tal y como se estudia en el presente artículo, el hito no alcanzado cuya ruta cuente con el menor número de giros será definido como el próximo hito. Posteriormente, cargamos en nuestro sistema la ruta óptima para alcanzar el mismo. Teniendo en cuenta que en el caso de que nos encontremos ya sin hitos que alcanzar, debemos enrutarnos hasta la salida.

Punto 4.2 Guiado en la ruta para alcanzar el hito: Tras el cálculo de la ruta, el sistema comienza el guiado mediante órdenes auditivas, indicando cual debe ser el siguiente paso a dar. Estas órdenes deben darse de forma clara y concisa siempre teniendo en consideración el posicionamiento del individuo, y la ruta a realizar. En el momento que el sistema identifique que el usuario ha salido de la ruta a trazar, debemos proceder a comenzar con el recalculo que se realiza en el punto anterior. Es por esto que el conocimiento del posicionamiento en tiempo real cobra vital importancia para un guiado y enrutado correcto. Una vez que alcanzamos el hito, será el módulo de localización el que cogerá el mando para alcanzar finalmente el hito y así será este módulo el que se encargará de definir el hito como alcanzado.

Con todo esto, queda definido cual va a ser el plan de actuación del sistema para llevar a cabo la tarea establecida.

7. Conclusiones

En este proyecto hemos realizado el diseño inicial de un sistema de enrutamiento y guiado para un dispositivo móvil en un entorno interior. Se ha tenido en cuenta la necesidad de adaptar estos sistemas a las personas con discapacidad visual para incrementar su accesibilidad, y se ha contado con que la simplicidad de la ruta influirá en el éxito del uso de este sistema.

Para el diseño de la ruta se han estudiado diferentes algoritmos de enrutamiento, con la conclusión de que los algoritmos determinista, tanto Djisktra como A*, son los más óptimos dado la definición de los requisitos impuestos.

Se ha definido también cual va a ser el plan de acción del sistema, a partir de la división del problema. Incidiendo en los pasos que van a llevar a cabo el usuario a lo largo del proceso de guiado.

8. Futuros Trabajos

Una vez realizado el diseño de los módulos de posicionamiento, guiado, e identificación, realizaremos un estudio de mercado, que nos permitirá hacer un balance de la accesibilidad y usabilidad de nuestro sistema.

La aplicación final se desarrollará en principio para el sistema operativo Android como software libre, lo que facilitará el acceso a los usuarios en mayor número, y además las

librerías necesarias para el desarrollo de los módulos del proyecto son software libre, con lo que podremos emplearlas o modificarlas según los requerimientos del sistema.
La distribución a los usuarios sería simple, ya que el programa podría ser descargado gratuitamente como una aplicación desde el Android Market.

Para finalizar, indicar que este sistema no sólo tiene como objetivo a personas con discapacidad visual, sino que debemos tener en cuenta, que con posterioridad puede existir una adecuación del sistema para el empleo de éste por parte de diferentes usuarios como personas de la tercera edad, que también necesitan de asistencia a la hora de realizar este tipo de acciones rutinarias.

9. Bibliografía

[1] Hua Wu, Alan Marshall, Wai Yu "**Path Planning and Following Algorithms in an Indoor Navigation Model for Visually Impaired**" School of Electronics, Electrical Engineering and Computer Science, Queen's University of Belfast, Belfast, Northern Ireland, United Kingdom, BT9 5HN

[2]Ben Harrison, Hua Wu, Alan Marshall, Wai Yu "**The ENABLED Indoor / Outdoor Navigati on Systems for the Blind and Visually Impaired** " Virtual Engineering Centre, Queens University Belfast, Cloreen Park, Malone Road, Belfast, Northern Ireland

[3] T. H. Riehle, Member, IEEE, P. Lichter, Member, IEEE, and N. A. Giudice **"An Indoor Navigation System to Support the Visually Impaired "**30th Annual International IEEE EMBS Conference Vancouver, British Columbia, Canada, August 20-24, 2008

[4] Robert J. Szczerba, Danny Z. Chen , Kevin S. Menkt **"Minimum Turns / Shortest Path Problems: A Framed-Subspace Approach" 1997**

[5]KamranH.Sedighi,KavehAshenayi,TheodoreW.Manikas,RogerL**.Wainwright,and Hengming Tai.** Autonomous local path planning for a mobile robot using a genetic algorithm.