Performance evaluation of path planning algorithms for blind people

Paula Mosquera-Ortega[1], Andrés Díaz-Toro[2], Anyela Villamizar-Carrillo[3], Sixto Campaña-Bastidas[1]
[1]School of Basic Sciences, Technology and Engineering, Universidad Nacional Abierta y a Distancia, Pasto, Colombia
[2]Department of Electrical, Electronics, Systems and Telecommunications, Faculty of Engineering and Architecture, University of Pamplona, Pamplona, Colombia
[3]Department of Arts, Faculty of Arts and Humanities, University of Pamplona, Pamplona, Colombia

Article Info

Article history:

Received Mar 24, 2025
Revised Oct 31, 2025
Accepted Nov 16, 2025

Keywords:

Artificial intelligence
Assistive technology
Embedded systems
Occupancy grids
Purposeful navigation
Search algorithms

ABSTRACT

Blind people face difficulties in identifying objects of interest and moving to them safely and efficiently in unfamiliar environments. Thanks to high-performance computers, high-quality sensors and artificial intelligence algorithms, it is possible to perform real-time tasks such as locating the user, generating occupancy grids that represent the environment and identifying objects of interest. From this information, paths can be generated that allow the user to reach a point of interest in an optimal way. This paper presents the performance evaluation of four path planning algorithms that were implemented in MATLAB and tested with synthetically generated occupancy grids, varying their size and occupancy percentage. The evaluation criteria include time to reach the goal, number of expanded cells and number of cells in the path. In addition, a single indicator that integrates all performance criteria is proposed to identify the best algorithm. The results show that the A* algorithm presents the best performance in static environments, under certain hardware requirements for data processing and restrictions on grid size for real-time applications. These findings expand the fields of application of path planning algorithms, quantify their performance under different conditions of the environment, and make them attractive for implementation in embedded systems.

Corresponding Author:

Andrés Díaz-Toro
Department of Electrical, Electronics, Systems and Telecommunications
Faculty of Engineering and Architecture, University of Pamplona
Kilometer one, Bucaramanga Road, Pamplona, Colombia
Email: andres.diazto@unipamplona.edu.co

1. INTRODUCTION

Approximately 80% of human learning is mediated through vision. Its absence significantly impacts various activities such as spatial orientation, independent mobility, and access to visual information, including signage, gestures, and facial expressions [1], [2]. It hampers reading, performing everyday tasks such as cooking, cleaning, or shopping, and using technologies not designed for accessibility. It also limits educational and employment opportunities and can contribute to social isolation.

At least 2.2 billion people worldwide have some form of near or distance vision impairment. In almost a billion of these cases, visual impairment could have been avoided if some preventive treatment was applied [3]. In Colombia, the 2,018 National Population and Housing Census revealed that 1,948,332 people have visual impairment, which represents 62.17% of the population with disabilities in the country and is equivalent to 7.1% of the total population [4].

Currently, blind people have tools such as white canes and guide dogs that facilitate mobility but these methods do not offer detailed information about objects in the environment or the best path to reach a destination. Technological advancement has allowed the development of tools that improve the quality of life of people with visual impairments. These innovations can facilitate reading and improve accessibility to various daily activities [5]. Nevertheless, there are few developments focused on assisting blind people in their mobility, a key challenge for their independence and safety in unknown environments.

Electronic travel aids (ETAs) are assistive tools that help blind people navigate unfamiliar spaces safely and independently. Several assistants focus on obstacle detection and avoidance, while others include additional functions such as environment description, object detection, fall detection, and water detection, but without path planning. Below are some outstanding systems of this type, based on smart canes, wearables, and guidance robots, which include various sensors such as ultrasound, light detection and ranging (LiDAR), red, green, and blue (RGB) cameras, and depth cameras.

Electronic canes are assistive tools that have gained great popularity, thanks to their ability to detect objects even without physical contact and to include other features such as a panic button [6], power supply with solar panels and water detection [7]. Their main advantage over conventional canes is their ability to detect hanging obstacles, as analyzed in [8]. Ali [9] compared two obstacle detection devices based on ultrasound sensors are: a smart cane and a belt, identifying advantages of these over the classic cane in terms of usability and accuracy.

The system [10], in addition to detecting obstacles and moving objects, identifies if the user has fallen and sends an alarm to the guards when this occurs, indicating the exact position. It uses ultrasonic sensors, a passive infrared (PIR) motion sensor, an accelerometer, a microcontroller, a cell phone app, and a transmitter. The system is lightweight and compact, allowing it to be worn on the shin. User feedback is provided through voice instructions. Another type of navigation assistant is the one based on shoes that integrate ultrasound sensors for obstacle detection. Wu *et al.* [11] use accelerometers are included that allow falls to be detected and when this happens, the system calls an emergency contact.

In addition to ultrasound sensors, [12] includes a milli-LiDAR and inertial sensors, along with a fuzzy classifier and a navigation algorithm. The system is implemented on a Raspberry Pi 4 running robot operating system (ROS) 2 to integrate all the modules into nodes and provide feedback to the user with voice and haptic commands. The mini-LiDAR is housed in a hand-band unit, while the ultrasound sensors are housed in a head-band unit.

To detect objects of interest in real time, embedded computers such as the Jetson Nano and Raspberry Pi 3 B+ are used. The processing speed, latency, and accuracy of the inferences made by these algorithms are very important. For this reason, Joshi *et al.* [13] compare their performance considering the requirements of ETAs. Boussihmed *et al.* [14] propose real-time obstacle detection on sidewalks to assist blind and low vision people in mobility using lightweight deep learning models and internet of things (IoT) devices with memory and processing limitations. In addition to detecting obstacles, blind people require a description of the environment. For this purpose, deep learning models such as SSDLite MobileNetV2 have been used in [15]. A camera, a Raspberry Pi 4B, and auditory feedback are used. The device is integrated into a cap so that it is not noticed.

Path planning algorithms are widely used in robotics but in recent years, the most advanced assistance systems include them to generate efficient and safe trajectories. The calculated route must be the shortest or fastest, and the directions for following that route must be clear so as not to overload the user with too much information. Furthermore, the directions given to the user must not interfere with auditory information about the environment, so haptic feedback is preferred. Among the most widely used path planning algorithms for navigation assistance for the blind are Dijkstra's and A* [16]. The most notable systems that integrate path planning are presented below.

A greedy path planning algorithm called OPTIPATH is used in [17] to estimate a route along the corridors of a supermarket, thus offering blind people an autonomous shopping experience, avoiding collisions with obstacles and minimizing the number of turns along with the distance traveled. The algorithm can be used with any obstacle detection assistant such as canes with ultrasonic sensors.

The wearable system [18] employs an RGBD (red-green-blue + depth) camera integrated into glasses, a smartphone user interface, a visual odometry-based localization algorithm, and the D* lite algorithm to calculate the shortest path to the target and guide a person in indoor environments through audio and haptic commands. The environment is represented in the form of 3D voxels that define spaces to walk safely. The wearable system called intelligent situation awareness and navigation aid (ISANA) [19] creates a map with high-level semantic information and calculates an optimal route in indoor environments using the A* search algorithm. It includes a module for obstacle detection and avoidance based on data from an RGBD camera. The feedback is auditory with a priority-based mechanism to avoid overloading the user. The wearable system [20] uses a priori information about the structure of the environment, inertial sensors, an RGB-D camera, global positioning system (GPS), and a Jetson Orin Nx processor that allows it to run different algorithms in real time to improve the quality of depth maps, create a map with semantic information, and estimate the position of the user navigating in outdoor environments. A route to objects of interest is generated using D* lite, and the user is guided using a bone conduction headset.

Wang *et al*. [21] introduced a smartphone-based system. The user initially explores an unknown environment and the system creates a representation of it (2D grid), computing the localization using a stereo camera and a visual simultaneous localization and mapping (SLAM) algorithm. The user then chooses an object of interest, and the system calculates the optimal route and guides the user safely around obstacles through audio and haptic instructions. The algorithm employed to estimate optimal routes is A*. To simplify the route and thus make it easier for the user to follow the instructions, the Douglas-Peuker algorithm [22] is implemented. The ANSVIP navigation assistant for the blind [23] uses haptic (haptic glove) and auditory feedback to guide the user to a goal. It runs a SLAM algorithm on an ARCore-compatible smartphone to estimate motion and for situational awareness along with the mapping. It employs an adaptive artificial potential field algorithm that generates smooth and safe routes. The smartphone-based Corridor-Walker system [24] assists blind people in recognizing obstacles and identifying intersections in corridors of buildings such as apartments, offices, and hospitals. The system constructs a 2D occupancy grid from a LiDAR sensor and guides the user using audio and vibrations. The path planning algorithm is A*, while you only look once (YOLO) v3 is used to identify intersections.

CCNY [25] is a robotic cane that assists blind people with indoor mobility, integrating a SLAM localization and mapping system, an IR depth camera (based on Google Tango), and the A* path planning algorithm with Manhattan distance-based heuristics. The device generates real-time navigation instructions through haptic feedback. The autonomous suitcase-shaped robot, called carry-on-robot (CaBot) [26], guides blind people to a destination while avoiding obstacles along the route and detecting pedestrians. It has a ZED stereo camera and a LiDAR to obtain information about the environment. Information is provided through auditory and haptic feedback. For route estimation, it uses the ROS navigation packages: Navfn global path planner2 and DWA local path planner3 [27]. Tests conducted on the vibro-tactile handle, walking speed, and ease of handling demonstrated high confidence in the robot.

This paper compares the performance of four path planning algorithms: Spanning tree (ST), Breadth First Search (BFS), Dijkstra (DIJ), and A start (A*), considering three criteria: execution time, number of expanded cells, and number of cells in the path. This analysis allows us to identify the characteristics of each algorithm and its performance, important information for their implementation on portable embedded computers, which, despite significant progress in recent years, suffer from processing and memory limitations. This analysis is conducted considering different environmental conditions such as size and complexity, and within the context of assisting blind people. The main contribution of this work are: i) the overall performance indicator based on the three aforementioned criteria, which allows identification of the A* algorithm as the most appropriate option for path planning in navigation assistance systems for blind people in static indoor environments, due to its efficiency in terms of execution time and number of expanded nodes and ii) the analysis of computational requirements in the implementation of these algorithms for a real-time assistance application that impose restrictions on the type of embedded hardware and on the size of the gird that allows representing an adequate area without compromising the viability of the system.

This paper is organized as follows. Section 2 describes the methodology to determine the performance of the path planning algorithms. Section 3 analyzes the proposed overall performance indicator, based on three performance criteria (execution time, number of expanded cells, and number of cells in the path). A comparison with other systems is made, and practical implications for its implementation in embedded computers are defined. Section 4 presents the main conclusions of this work. Finally, funding information, author contributions, conflicts of interest, data availability, references, and bibliographies of the authors are presented.

2. METHOD

ETAs for assisting blind people in purposeful navigation are focused on guiding a person to a goal. These systems work in unknown environments where the user needs to reach a point of interest such as a computer, a table, a chair, a door, or even a person. The starting point is the user's current location and the end point is the location of the goal. The most appropriate representation of the environment is through occupancy grids, since it allows the system to identify free and occupied spaces using intermediate storage and processing resources, for environments such as offices, rooms in educational institutions, airports, and shopping centers. Given the user's location, the location of the object of interest, and the representation of the environment in the form of an occupancy grid, the problem is reduced to find the optimal path between two nodes. The problem of finding the optimal path is presented graphically in Figure 1.

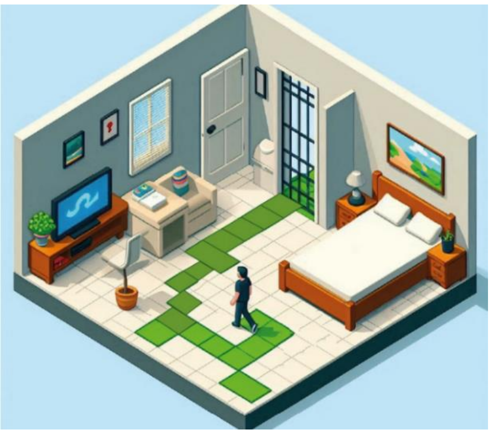


Figure 1. Description of the problem of finding an optimal path to a goal, applied to blind people

The diagram with the proposed methodology to evaluate the performance of search algorithms is presented in Figure 2. The first step is to define the requirements of the application to assist blind people in navigation. This application requires processing at least 5 images per second, in order to represent changes in the environment given the user's speed of 0.26 – 1.11 m/s [28]. This means that the occupancy grid, the objects of interest and the user's location must be updated at least 5 times per second, so the path planning algorithm must be efficient to reduce the consumption of computing resources. On the other hand, small and light processing equipment is required, which is easy for the user to carry. For processing, portable computers carried on the user's back or high-performance development kits such as the Jetson Nano, Jetson Orin Nx and Jetson TX2 have been used. This portability entails the use of computers that have limitations in computing capacity and memory, so the path planning algorithms must also be efficient in memory consumption. Considering performance requirements, portable computing equipment and limitations in processing and memory resources, a performance comparison, is very important for its implementation in real applications.

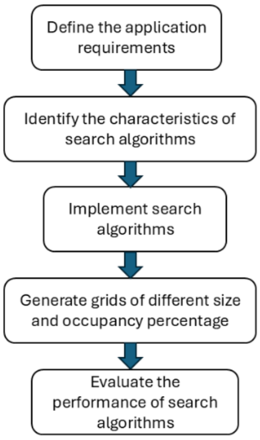


Figure 2. Block diagram of the methodology for evaluating the performance or search algorithms

If the representation of the environment were continuous, the number of possible routes to a target would be infinite. For this reason, a graph representation is used, reducing it to a search problem in a discrete space. We work with uniform grids where each cell is a node of the graph and each edge is associated with a movement action that leads from one node to another. This representation allows us to evaluate the cost of the paths and avoid collisions with obstacles. Small cells have the advantage of a more exact representation of obstacles but require more memory and processing. The search algorithm determines a sequence of decisions about which edge to follow in such a way that it connects the initial node with the target node. The main characteristics of the search algorithms are the following:

a) Optimality: the algorithm calculates the best path, regardless of whether there is more than one solution. In this application, we work with a uniform grid (with the same transition cost between neighboring nodes) so the best path is the one with the lowest number of nodes (the shortest path).
b) Feasible: the calculated paths satisfy the user's movement constraints. For this application the user can move diagonally such that each node has eight neighbors.
c) Admisible: the calculated paths do not pass through obstacles, that is, the nodes that make up these paths are empty nodes.
d) Completeness: the algorithm will find a solution if one exists. Otherwise, it will report failure.

The search algorithms are used mainly in robotics for reaching the destination in an optimal way, that is, finding the path with the least cost. In uniform grids, it means the path with the least number of nodes, which represents the least effort for the robot, or in this case, for the blind person.

ST: it uses a tree where the node being expanded is the parent and the expanded nodes are the children. Open and close lists are used, which allow the next node to be chosen randomly, but with two conditions: that the node is not an obstacle and that it has not been evaluated before. This process can take into account the nodes that have already been visited, avoiding cyclical paths. Each node can have only one parent so that when the goal node is found, the route is generated in reverse order, following the parents starting from the parent of the goal node.

BFS: it explores the nodes closest to the origin before moving to more distant nodes. It is used in graphs where the cost of moving between nodes is uniform. This form of expansion is called the wavefront optimality principle. The open list is now a first in, first out (FIFO) queue, so nodes are removed from the front of the FIFO queue and placed at the back.

DIJ: it finds the shortest route in graphs with non-uniform movement costs. It does not use a heuristic or the direction of the objective, so the computational cost is high. Instead of a FIFO queue, it uses a priority queue that is organized by cost-to-come (cost to get from the initial node to the current node) such that the removed node is the one with the lowest cost.

A*: it uses a heuristic to guide the search towards the objective efficiently. Its cost function considers the cost from the initial node to the current one (cost-to-come) and the cost defined by the heuristic (cost-to-go). The first cost is exact and known while the second cost is an estimate. The priority queue is organized based on the sum of the two previous costs, it means, the cost of going from the start node to the goal node. In this application, the Manhattan heuristic is used.

The four algorithms described above are feasible, admissible, complete but only the last three are optimal. The algorithms were implemented in MATLAB, starting with the most basic algorithm. From this, small adjustments are made that lead to the other algorithms. This methodology is similar to that presented in [29].

The general pseudocode of the search algorithms implemented in this work is presented in Figure 3. In lines 1 and 2, the current node is defined as the start node and is inserted into the open list (O) containing the nodes to be explored. In line 3, the termination condition is found, which indicates that the search algorithm is executed as long as the open list is not empty. In line 4, a node is extracted from the open list and is defined as the current node. In line 5, the current node is inserted into the close list (C), which contains the nodes already explored. In line 6, it is verified whether the goal node was reached. In that case, the search is successful and finishes. Otherwise, the expandNode() function is executed, as shown in line 9. This function is responsible for generating the reachable neighbor nodes from the current node and adding them to the open list, according to the rules of the search algorithm. Depending on how the open list and the expansion function expandNode() are handled, different algorithms are obtained.

Data: Occupancy grid ***G***, start node ***Xs*** and goal node ***Xg***
Result: Path connecting the start node to the target node

```
00  Search algorithm for path planning (G, Xs, Xg)
01  Current node X ← Start node Xs
02  O.insert(Current node X)
03  while (O is not empty) do
04          Current node X ← O.remove()
05          C.insert(X)
06          If(X==Xg)
07                  Return success
08          else
09                  expandNode()
10          end
11  end
```

Figure 3. General pseudocode for search algorithms

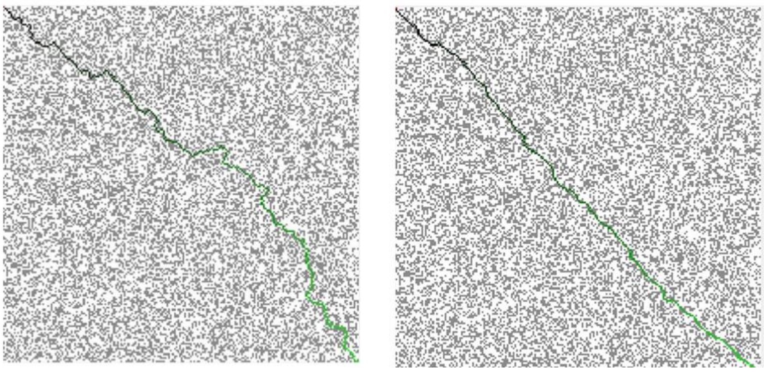
To calculate the performance of the algorithms, occupancy grids were created that represent different sizes and complexity of the environment. Three grid sizes were used: 100x100, 200x200, and 400x400 nodes. If each node is assumed to be 0.1 mx0.1 m in size, square spaces of 10 mx10 m, 20 mx20 m and 40 mx40 m would be covered, respectively. On the other hand, three occupancy percentages are defined, which are related to the complexity of the environment, since the greater the occupancy, the more extensive the search must be performed by the algorithm. These percentages were: 20%, 40%, and 50%. These grids were created randomly in compliance with the occupancy percentages. To determine the performance, the following criteria were used:

a) Execution time: this parameter refers to the duration, in time units, that the algorithm took to calculate the path until reaching the objective.
b) Expanded cells: this parameter considers the number of nodes that were expanded. The greater the number of expanded nodes, the higher the computational cost.
c) Number of nodes in the path: this is the number of nodes that make up the calculated path. The algorithms search for a solution that minimizes this parameter.

3. RESULTS AND DISCUSSION

The tests were carried out for different grid sizes and for different occupancy percentages. The initial point was the node located in the upper left corner of the grid and the goal point was the node in the lower right corner. Figure 4 shows the occupancy grid and the calculated path for each algorithm, with a grid of 200x200 nodes and for an occupancy percentage of 40%. The path is drawn in green, with a lighter tone as the iterations increase. The notation used to identify each algorithm is as follows. ST, BFS, DIJ, and A*.

Note that the path shown in Figure 4(a) is longer than the one shown in Figure 4(b). This is because the ST algorithm uses an exploration-based search structure, which tends to generate suboptimal paths. In contrast, the BFS, DIJ, and A* algorithms implement more advanced search strategies that allow obtaining optimal paths. These last three algorithms differ from each other in their exploration approaches, which is reflected in the number of expanded nodes and the execution time, as analyzed below.



(a) (b)

Figure 4. Path calculated by (a) ST and (b) BFS, DIJ and A*. Grid: 200x200 and 40% of occupancy

The time taken by each algorithm to find a path to the goal node was compared, considering different occupancy percentages, for a constant grid size of 200x200 nodes, as shown in Figure 5. It is observed that there is a small increase in the execution time due to the increase in the complexity of the environment, observing in all cases that the algorithm with the lowest execution time was A*, followed by ST, BFS, and DIJ.

The time taken by each algorithm to carry out the calculations was also calculated, varying the size of the grid and leaving the occupancy percentage constant at 40%, as shown in Figure 6. It can be observed how the time increases considerably for a grid of 400x400 nodes. This puts a restriction on the size of the grid and on the size of each node, given the limited resources of an embedded system and the real-time requirements of the application. The algorithms ST, BFS, and DIJ presented a low performance compared to A*. This last algorithm did not grow in the same proportion as the others since it used a heuristic that guided the search efficiently.

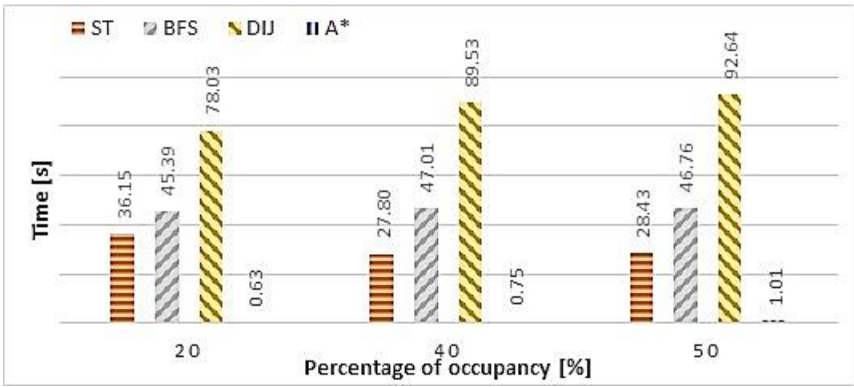


Figure 5. Time to calculate the path to the target node considering changes in the occupancy percentage. The grid has a size of 200x200 nodes

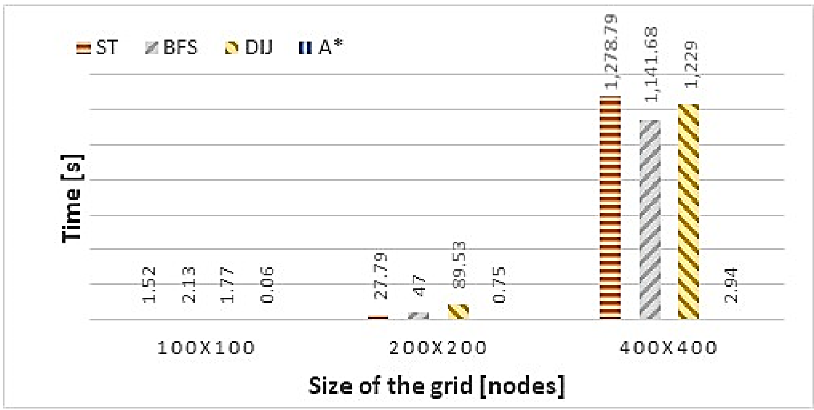


Figure 6. Time to calculate the path to the target node considering changes in the grid size. The grid has 40% of occupancy

For the last case, where the occupancy percentage is kept constant (40%) and the grid size is varied, two more criteria are evaluated. Figure 7 shows the number of nodes expanded by each algorithm to calculate a path. Note that the A* algorithm expands a significantly smaller number of nodes than the other three algorithms, which have a very similar behavior among themselves with respect to this variable. This is due to the greater search efficiency obtained by using a heuristic.

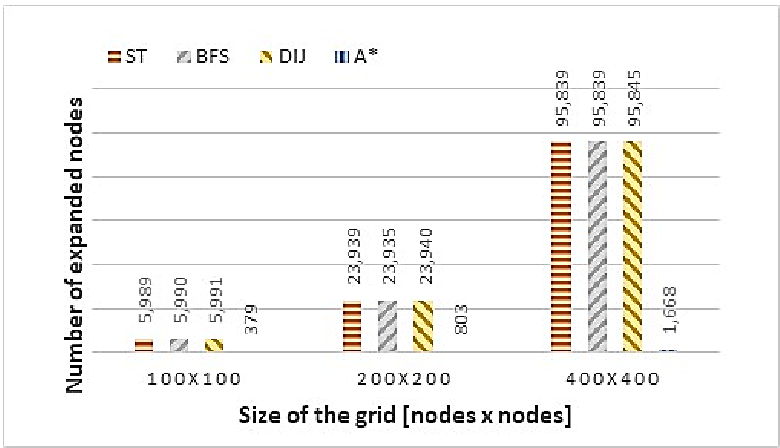


Figure 7. Number of nodes expanded by each search algorithm in a 40% occupancy grid

The number of nodes that make up the path calculated by each search algorithm is presented in Figure 8. Note that ST is the algorithm that generates the longest path while BFS, DIJ, and A* generate optimal routes, that is, with the smallest number of nodes.

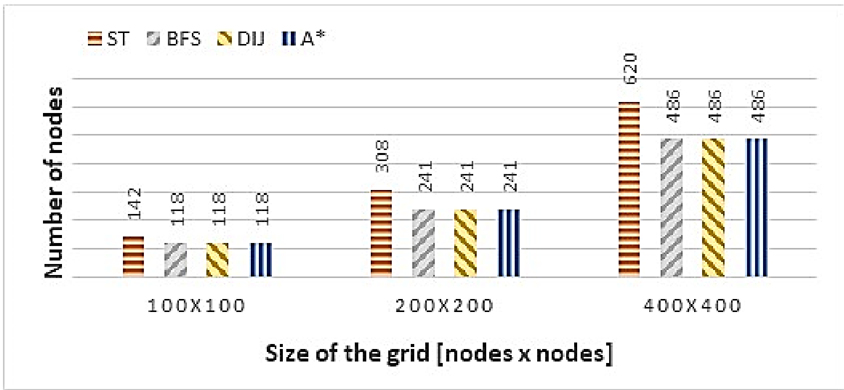


Figure 8. Number of nodes that make up the path in a grid with 40% occupancy

Given these three indicators (time, expanded nodes, and nodes that make up the obtained path), a general indicator was proposed that brings together the information from all three. The higher these indicators are, the lower the performance is, so we worked with the normalization with inverse relationship presented in (1) which assigns 1.00 to the lowest indicator and 0.00 to the highest indicator.

$$I_n = \frac{I_{max} - I}{I_{max} - I_{min}} \tag{1}$$

Where I_{max} is the indicator with maximum value, I_{min} is the indicator with minimum value, I is the indicator to be normalized and I_n is the normalized indicator. Therefore, the closer the general indicator is to 1.00, the better its performance. Table 1 presents the result of normalization. Note that A* had the best results while ST had the worst.

Table 1. Normalized indicators of time, expanded nodes and nodes on the path

Criteria/algorithm	ST	BFS	DIJ	A*
Time	0.33	0.20	0.07	1.00
Expanded nodes	0.00	0.00	0.00	1.00
Nodes on the path	0.00	1.00	1.00	1.00

Finally, these values were averaged for each algorithm, obtaining the overall indicator presented in Figure 9. It can be observed that A* had the best overall performance (1.00) while ST had the worst performance (0.11). The BFS and DIJ algorithms had a similar overall performance since they share many characteristics in the path search. However, the DIJ algorithm carries out more operations to deal with non-uniform grids (variable transition costs between nodes), although in this case the grid is uniform, so the execution time was longer, affecting its overall indicator.

This overall performance indicator tells us that the most suitable algorithm for implementation in embedded computers, considering large, static indoor environments, is A*. This statement is consistent with the one presented in [16], which states that A* is the most widely used path planning algorithm in navigation aids for blind people. Among these navigation assistants are [19], [21], [24]-[26].

The results demonstrate the need to carefully consider the computational requirements for implementing path planning algorithms in ETAs for blind people. The choice of hardware is a critical factor, since processing occupancy grids and executing algorithms in real time requires devices with advanced computing capabilities. In this sense, platforms such as the Nvidia Jetson Orin Nx [20] or high-performance smartphones [21], [24], are presented as viable options to support the computational demands of the system.

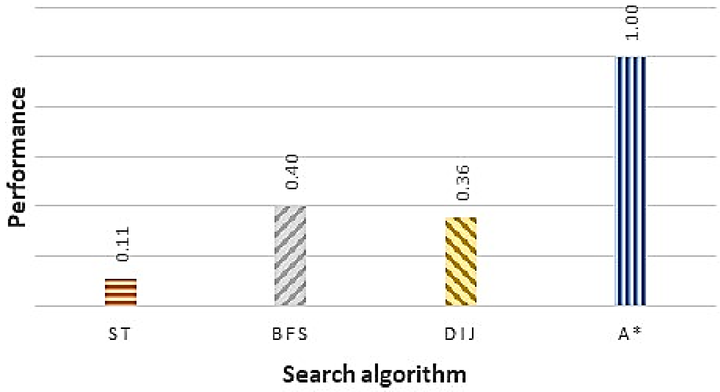


Figure 9. Overall performance indicator for each search algorithm

Another relevant aspect identified in the study is the relationship between grid size and computational cost. It was observed that an increase in the number of cells significantly impacts the algorithm's execution time, which could compromise the viability of its implementation on portable hardware. To mitigate this effect, it is proposed to restrict the grid size to 100×100 cells, which would allow representing an area of 10 m×10 m, considering a 0.1 m×0.1 m cell. This size is suitable for indoor environments, where conditions are relatively static and large-area coverage is not required.

For larger environments, a viable strategy is to maintain the same number of cells and increase the size of each cell, which reduces the resolution of obstacle representation but decreases the algorithm's execution time. This approach allows scaling coverage without significantly affecting the system's computational performance, although at the cost of lower path planning accuracy. Kapadia *et al.* [30] mention some grid cell sizes ranging from 0.4 mx0.4 m to 1.5 mx1.5 m. For the latter cell size, a 100x100-cell grid could cover an area of 150 mx150 m, which is sufficient for large environments.

In dynamic scenarios, where obstacle locations change over time, a constant update of the path planning is required. In this case, the BFS algorithm is suggested, reinitializing the search each time a change in the grid is detected. Since this algorithm is highly parallelizable, implementing it on a CPU or even a GPU can significantly improve response times, as is demonstrated in [31], allowing the user to receive real-time guidance while navigating through the environment.

Finally, a crucial aspect to consider in the implementation of these systems is the usability and user comprehension of the instructions. It has been identified that repeated changes in the user's trajectory, which involve constant modifications in the direction of movement, can generate instruction overload and make it difficult to follow the planned route. To mitigate this problem, the use of route simplification algorithms, such as the Douglas-Peucker algorithm [22], is recommended. This algorithm smooths the trajectory while maintaining its efficiency, reducing the number of direction changes, and facilitating the user's interpretation of instructions.

4. CONCLUSION

In this work, the performance of four path planning algorithms was evaluated in synthetic environments represented by occupancy grids. The algorithms were implemented and tested in MATLAB, varying the grid size and occupancy percentage, in order to quantify their performance under different conditions of the environment. The algorithm A* presented the best overall performance, achieving the highest value on the proposed global indicator (1.00). In contrast, ST showed the worst performance (0.11), making it less suitable for application in navigation systems for the blind with real-time requirements. On the other hand, the BFS and DIJ algorithms presented similar performance due to their shared search strategies. However, it was observed that DIJ requires more operations to manage grids with non-uniform transition costs, which increased its execution time in the evaluated case of uniform grids, negatively affecting its overall performance.

The number of grid nodes directly affect the computational load, while the density of obstacles has a minor effect on the algorithm's performance. For this reason, it is recommended to set a limit on the grid size for applications on portable hardware, allowing for efficient execution without compromising system viability. For a 100x100-cell grid, the execution time with A* was 60 ms, which would allow real time performance.

These findings highlight the importance of selecting the appropriate path planning algorithm based on the environmental conditions and the system's computational requirements. In particular, the superiority of A* in static environments positions it as a viable option for implementation in embedded devices intended for navigation assistance for blind people. As future work, we suggest evaluating these algorithms in the Nvidia Jetson Orin Nx.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Paula Mosquera-Ortega		✓	✓	✓	✓	✓		✓		✓	✓		✓	
Andrés Díaz-Toro	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	
Anyela Villamizar-Carrillo				✓		✓	✓		✓		✓			
Sixto Campaña-Bastidas		✓			✓		✓			✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author ADT on request.

REFERENCES

[1] K. M. Hoogsteen and S. Szpiro, “A holistic understanding of challenges faced by people with low vision,” *Research in Developmental Disabilities*, vol. 138, p. 104517, Jul. 2023, doi: 10.1016/J.RIDD.2023.104517.

[2] D. Boey, T. Tse, Y. hui Lim, M. L. Chan, K. Fitzmaurice, and L. Carey, “The impact of low vision on activities, participation, and goals among older adults: a scoping review,” *Disabil Rehabil*, vol. 44, no. 19, pp. 5683–5707, 2022, doi: 10.1080/09638288.2021.1937340.

[3] World Healt Organization, “Blindness and vision impairment,” Accessed: Mar. 18, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[4] National Administrative Department of Statistics, “DANE - Censo Nacional de Población y Vivienda 2018.” Accessed: Mar. 18, 2025. [Online]. Available: https://www.dane.gov.co/index.php/estadisticas-por-tema/demografia-y-poblacion/censo-nacional-de-poblacion-y-vivenda-2018

[5] S. S. Senjam, S. Manna, and C. Bascaran, “Smartphones-based assistive technology: accessibility features and apps for people with visual impairment, and its usage, challenges, and usability testing,” *Clin Optom (Auckl)*, vol. 13, pp. 311–322, 2021, doi: 10.2147/OPTO.S336361.

[6] K. Singh, H. Yadav, K. Aggarwal, and G. Sharma, “Smart walking stick: an aid to visually impaired,” In *Proceedings of the Third International Conference on Information Management and Machine Intelligence: ICIMMI*, pp. 127–140, 2023, doi: 10.1007/978-981-19-2065-316.

[7] M. W. Apprey, K. T. Agbevanu, G. K. Gasper, and P. O. Akoi, “Design and implementation of a solar powered navigation technology for the visually impaired,” *Sensors International*, vol. 3, p. 100181, Jan. 2022, doi: 10.1016/J.SINTL.2022.100181.

[8] A. D. P. dos Santos, F. O. Medola, M. J. Cinelli, A. R. Garcia Ramirez, and F. E. Sandnes, “Are electronic white canes better than traditional canes? A comparative study with blind and blindfolded participants,” *Universal Access in the Information Society*, vol. 20, no. 1, pp. 93–103, Mar. 2021, doi: 10.1007/S10209-020-00712-Z/FIGURES/9.

[9] Z. A. Ali, "Design and evaluation of two obstacle detection devices for visually impaired people," *Journal of Engineering Research*, vol. 11, no. 3, pp. 100–105, Sep. 2023, doi: 10.1016/J.JER.2023.100132.
[10] M. M. Rahman, M. M. Islam, S. Ahmmed, and S. A. Khan, "Obstacle and fall detection to guide the visually impaired people with real time monitoring," *SN Comput Sci*, vol. 1, no. 4, pp. 1–10, Jul. 2020, doi: 10.1007/S42979-020-00231-X/METRICS.
[11] W. Wu, N. Lei, and J. Tang, "Smart shoes for obstacle detection," *Advances in Intelligent Systems and Computing*, vol. 1274 AISC, pp. 1319–1326, 2021, doi: 10.1007/978-981-15-8462-6_151.
[12] Y. Bouteraa, "Smart real time wearable navigation support system for BVIP," *Alexandria Engineering Journal*, vol. 62, pp. 223–235, Jan. 2023, doi: 10.1016/J.AEJ.2022.06.060.
[13] R. Joshi, M. Tripathi, A. Kumar, and M. S. Gaur, "Comparative study of object recognition algorithms for effective electronic travel aids," *Lecture Notes in Networks and Systems*, vol. 175, pp. 307–316, 2021, doi: 10.1007/978-3-030-67187-7_32.
[14] A. Boussihmed, K. El Makkaoui, I. Ouahbi, Y. Maleh, and A. Chetouani, "A TinyML model for sidewalk obstacle detection: aiding the blind and visually impaired people," *Multimed Tools Appl*, pp. 1–28, Sep. 2024, doi: 10.1007/S11042-024-20070-9/METRICS.
[15] R. B. Islam, S. Akhter, F. Iqbal, M. S. U. Rahman, and R. Khan, "Deep learning based object detection and surrounding environment description for visually impaired people," *Heliyon*, vol. 9, no. 6, Jun. 2023, doi: 10.1016/j.heliyon.2023.e16924.
[16] D. Khan, Z. Cheng, H. Uchiyama, S. Ali, M. Asshad, and K. Kiyokawa, "Recent advances in vision-based indoor navigation: a systematic literature review," *Comput Graph*, vol. 104, pp. 24–45, May 2022, doi: 10.1016/J.CAG.2022.03.005.
[17] D. Nandini and K. R. Seeja, "A novel path planning algorithm for visually impaired people," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 3, pp. 385–391, Jul. 2019, doi: 10.1016/j.jksuci.2017.03.005.
[18] Y. H. Lee and G. Medioni, "Wearable RGBD indoor navigation system for the blind," *Lecture Notes in Computer Science*, vol. 8927, pp. 493–508, 2015, doi: 10.1007/978-3-319-16199-0_35.
[19] B. Li, J. Pablo Muñoz, X. Rong, J. Xiao, Y. Tian, and A. Arditi, "ISANA: wearable context-aware indoor assistive navigation with obstacle avoidance for the blind," *Lecture Notes in Computer Science*, vol. 9914, pp. 448–462, 2016, doi: 10.1007/978-3-319-48881-3_31.
[20] H. Son, A. Vora, G. Pandey, and J. Weiland, "Infrastructure enabled guided navigation for visually impaired," *IEEE Transactions on Intelligent Transportation Systems*, 2025, doi: 10.1109/TITS.2025.3538317.
[21] M. Wang, D. Wu, C. Song, and J. Cheng, "A smartphone-based affordable navigation system using visual SLAM for blind and visually impaired people," *Res Sq*, Sep. 2024, doi: 10.21203/rs.3.rs-4976304/v1.
[22] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, Oct. 2006, doi: 10.3138/FM57-6770-U75U-7727.
[23] X. Zhang, X. Yao, Y. Zhu, and F. Hu, "An ARCore based user centric assistive navigation system for visually impaired people," *Applied Sciences 2019, Vol. 9, Page 989*, vol. 9, no. 5, p. 989, Mar. 2019, doi: 10.3390/APP9050989.
[24] M. Kuribayashi *et al.*, "Corridor-walker: mobile indoor walking assistance for blind people to avoid obstacles and recognize intersections," *Proc ACM Hum Comput Interact*, vol. 6, no. MHCI, p. 179, Sep. 2022, doi: 10.1145/3546714.
[25] Q. Chen *et al.*, "CCNY smart cane," *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER 2017*, pp. 1246–1251, Aug. 2018, doi: 10.1109/cyber.2017.8446303.
[26] J. Guerreiro, D. Sato, S. Asakawa, H. Dong, K. M. Kitani, and C. Asakawa, "Cabot: designing and evaluating an autonomous navigation robot for blind people," *ASSETS 2019 - 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 68–82, Oct. 2019, doi: 10.1145/3308561.3353771.
[27] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot Autom Mag*, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: 10.1109/100.580977.
[28] Z. R. Kahaki, A. R. Safarpour, and H. Daneshmandi, "The spatiotemporal gait parameters among people with visual impairment: a literature review study," *Oman J Ophthalmol*, vol. 16, no. 3, pp. 427–433, Sep. 2023, doi: 10.4103/ojo.ojo_24_23.
[29] A. Kelly, *Mobile robotics: mathematics, models, and methods*. 2013. Accessed: Mar. 18, 2025. [Online]. Available: https://books.google.com/books?hl=es&lr=&id=laxZAQAAQBAJ&oi=fnd&pg=PR13&dq=Mobile+Robotics:+Mathematics,+Models,+and+Methods&ots=MmNoeL7Yif&sig=19mjgy3svxlqNR6JmJAp6cW515g
[30] N. Fernando, D. A. McMeekin, and I. Murray, "Route planning methods in indoor navigation tools for vision impaired persons: a systematic review," *Disabil Rehabil Assist Technol*, vol. 18, no. 6, pp. 763–782, 2023, doi: 10.1080/17483107.2021.1922522.
[31] M. Kapadia, F. Garcia, C. D. Boatright, and N. I. Badler, "Dynamic search on the GPU," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3332–3337, 2013, doi: 10.1109/IROS.2013.6696830.

BIOGRAPHIES OF AUTHORS

Paula Mosquera-Ortega received her master of management of information technology in Universidad Nacional Abierta y a Distancia UNAD in 2024. She completed her B.Sc. degree in mathematics, from Universidad de Nariño, in 2001. Moreover, she obtained the degree as specialist in pedagogy of creativity in the same University, in 2014. Currently, she is working in the project "Methods for evaluating the performance of path planning algorithms focused to assist blind people". Since 2014, she has been working as assistant professor in mathematical science in several universities in Colombia, such as Universidad de Nariño, CESMAG, ESAP, and UNAD. Currently, she is working as assistant professor in mathematical science in UNAD and as a researcher in the Davinci group. She can be contacted at email: paula.mosquera@unad.edu.co.

Andrés Diaz-Toro received his B.Sc. degree in electronic engineering from Universidad de Nariño, Pasto, Colombia, in 2009, the M.Sc. degree in automation and the Ph.D. degree in electronic engineering from Universidad del Valle, Cali, Colombia, in 2012 and 2018, respectively. He has been assistant professor of undergraduate and graduate courses in Universidad del Valle, and in Universidad Nacional Abierta y a Distancia UNAD, in charge of a postdoctoral project for assisting blind people in purposeful navigation. Currently, he is Professor at Universidad de Pamplona, Norte de Santander, Colombia. He is part of the GISM research group at this university. He was awarded with Google Reseach Awards for Latin America 2018 with a project related to assistive tools for blind people. His research interests include mobile robotics, computer vision, artificial intelligence, process control, optimization techniques, and applications at real time with embedded computers. He can be contacted at email: andres.diazto@unipamplona.edu.co.

Anyela Villamizar-Carrillo received his master's degree in marketing from Universidad EAFIT in 2018. She graduated in multimedia engineering from Universidad Militar Nueva Granada in 2012 and obtained her specialist in multimedia project management degree in 2014. She is currently working on research projects in the area of user experience design, visual arts and multimedia. She has worked as a professor in systems engineering, multimedia engineering and in the master of user experience design. She served as a teacher, program leader and zone leader of the School of Basic Sciences, Technology and Engineering of the area of Bogotá-Cundinamarca of the Universidad Nacional Abierta y a Distancia - UNAD. She is currently professor at the University of Pamplona and belongs to the Conquiro research group. She can be contacted at email: anyela.villamizar@unipamplona.edu.co.

Sixto Campaña-Bastidas has a B.Sc. degree in systems engineering from Universidad Mariana, Pasto, Colombia, a M.Sc. degree in telematic networks and services, from Universidad del Cauca, Popayan, Colombia, a M.Sc. degree in free software from Universidad Autónoma de Bucaramanga, Colombia, and a Ph.D. degree in telecommunications from Universidad Pontificia Bolivariana, Medellín, Colombia. He has several postdoctoral stays in Spain and Sweden and is an International Intern for the European Union. Currently, he is working as associated professor and researcher in Universidad Nacional Abierta y a Distancia UNAD. In 2018 he was awarded with Google Reseach Awards for Latin America with a project related to Mothernet Project. His research interests include internet of things, machine learning, process control, optimization techniques and applications at real time with embedded computers. He can be contacted at email: sixto.campana@unad.edu.co.