TECHNICAL SCIENCE RESEARCH IN UZBEKISTAN
ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

BUILDING A TRAFFIC ROUTE TAKING INTO ACCOUNT OBSTACLES BASED ON THE A-STAR ALGORITHM USING THE PYTHON LANGUAGE

Vladyslav Yevsieiev[1], Amer Abu-Jassar[2], Svitlana Maksymova[1]

[1]Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2]Faculty of Information Technology, Department of Computer Science, Ajloun National University, Ajloun, Jordan

Abstract:

This paper explores the use of the A-Star algorithm to construct an optimal route for a mobile robot, taking into account obstacles using Python. The developed program allows you to generate a map with an arbitrary location of obstacles and automatically find the optimal path from the start to the end point, taking into account the complexity of the route. The paper presents the results of implementing the algorithm under various scenarios for the location of obstacles, which allows us to evaluate the effectiveness and reliability of the algorithm in various conditions. Experiments have shown that the A-Star algorithm provides fast and accurate route generation even in complex environments with many obstacles. The results of the study can be used to improve the autonomous navigation of mobile robots in real working conditions.

Key words: Industry 5.0, Mobile Robots, Work zone, Route Planning, Manufacturing Innovation, Industrial Innovation

INTRODUCTION

Mobile robots are now playing an increasingly important role in a human life, including industrial, medical, transportation and household tasks [1]-[12]. Effective route planning for mobile robots is a key element of their operation, especially in environments where obstacles such as walls, furniture, people and other objects are present [13]-[23]. Various methods and approaches can be used here [24]-[32].

The A-Star (A*) algorithm is one of the most widely used and effective algorithms for finding the optimal path in a space with obstacles. Its use allows mobile robots to find the shortest path from the start to the end point, taking into account the location of obstacles and other conditions along the way. The implementation of the A* algorithm in Python provides simplicity and flexibility in

TECHNICAL SCIENCE RESEARCH IN UZBEKISTAN
ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

developing software for controlling mobile robots. This is especially true in the rapidly evolving field of robotics and automation, where effective path planning plays a critical role in improving the productivity and safety of mobile robots. Thus, the development and implementation of the A* algorithm in Python for constructing travel routes for mobile robots taking into account obstacles is an urgent task with practical significance in the field of robotics and autonomous systems.

Related works

The A-Star algorithm is very popular for constructing routes in various fields. Many scientists are exploring the possibilities of its application, as well as its advantages and disadvantages; and trying to improve it in order to achieve defined goals. Let's look at some recent works on this topic.

Erke, S., and co-authors in [33] propose a novel variable-step based A-Star algorithm that allows them to reduce the computing time. They also developed the heuristic function to overcome the shortcoming of traditional A-Star algorithm.

Authors in [34] also note that the path planned by the A-Star algorithm is not the shortest path under certain conditions. And they developed an improved A-Star algorithm to solve path planning under certain conditions, which can find a shorter path in contrast of other related methods.

The paper [35] proposes an improved A-Star algorithm improves the search direction to 12 search directions. Such algorithm expands the traditional A-Star algorithm, because square A-Star algorithm has 8 search directions, regular hexagon A-Star algorithm has 6 search directions.

Scientists in [36] work with another problem of A-Star algorithm. They try to overcome the limitation of poor processing times for long-distance off-road path planning, Compared with the conventional A-Star algorithm, the path planning efficiency of their improved A-Star algorithm was accelerated by at least 4.6 times, and the maximum acceleration reached was 550 times for long-distance off-road path planning.

Tang, G., & et al. in [37] researched A-Star algorithm using for an Automated Guided Vehicle in order to avoid the problems of many nodes, long-distance and large turning angle, and these problems usually exist in the sawtooth and cross paths produced by the traditional A-Star algorithm. The simulation results of the path planning confirmed the effectiveness of the geometric A-Star algorithm.

The study [38] note that compared with other path planning algorithms, A-Star occupies a large memory space. To solve this problem, this paper proposes three new concepts such as the bidirectional search, a guide line and a list of key points.

The authors [39] propose A-Star algorithm to solve the path planning problem of the amphibious hull cleaning robot, which can clean the bottom of the ship and store the bottom condition for professional inspection.

A-Star algorithm, namely Dynamic Anti-collision A-Star (DAA-Star) algorithm, is proposed for the complex multi-ship encounter scenarios in [40].

Researchers in [41] focuse on finding the optimal route to tourism places in West Java Province. The A-STAR algorithm is implemented using Python so that the optimal route to tourism places in West Java Province is obtained.

The article [42] analyzes of A-Star, ThetA-Star, and Lazy ThetA-Star path planning strategies is presented in a 3D environment.

Thus, we see that research in the field of application of the A-Star algorithm for path determination. Later in this article we will look at using this algorithm to plan a robot's trajectory.

A-Star algorithm implementation for the route planning

The A* (A-Star) algorithm is an effective method for finding the optimal route from a starting point to an ending point on a map, taking into account obstacles. Its operating principle is based on a combination of heuristic and greedy approaches.

In its simplest form, the work of the A* algorithm can be represented as follows. Let S be the initial vertex, then $g(S)=0$ is the distance from the initial vertex S to itself is 0. Let's place the vertex S in an open list O containing vertices that have not yet been checked.

Let's search for the optimal path, let the open list O to be not empty, perform the following actions:

– select the vertex n with the smallest value of the evaluation function $f(n)$

$$f(n)=g(n)+h(n), \qquad ((1)$$

$g(n)$ – path length from the starting vertex S to the vertex n,

$h(s)$ – heuristic estimation of the distance from the vertex n to the final vertex G.

At the next step, we check, if n is the final vertex G, then the algorithm ends, otherwise for each neighbor m of the vertex n we check whether m belongs to an already open or closed list of vertices. If m is already in the public list, we check if its current estimation $f(m)$ can be improved, and if it is neither the public nor private list, we add it to the public list and calculate the values $g(m)$ and $f(m)$ for it.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

The heuristic estimate (h) is usually calculated as the distance from the current cell to the target cell, but can be adapted to take into account additional factors such as obstacles.

The algorithm selects the cell with the smallest sum g and h for further research, which allows it to go to the goal in the most efficient way.

Obstacles on the map prevent the algorithm from moving through the corresponding cells, causing it to take an alternative path around the obstacles.

The efficiency of the algorithm is ensured by the use of a priority queue (heap) to store and select the next cell to explore, which allows it to quickly find the optimal path.

After reaching the end point, the algorithm reconstructs the optimal route by following the parent links back from the goal to the starting point.

Thus, the A* algorithm ensures that the optimal route is found taking into account obstacles on the map, which makes it widely used in robotics, games, path planning and other fields.

It is important to note that the effectiveness of the algorithm may depend on the choice of heuristic function and the characteristics of the terrain map.

With the right choice of parameters and heuristic function, the A* algorithm provides high accuracy and speed in finding the optimal route, making it the preferred method for navigation in difficult conditions.

Software implementation and experiments

To check the correctness of the reasoning, we will develop a program in Python in the development environment PyCharm 2022.2.3 (Professional Edition). Let us give an example of software implementation of the above described mathematical expressions.

```
class Cell:
    def __init__(self, x, y, obstacle=False):
        self.x = x
        self.y = y
        self.obstacle = obstacle
        self.parent = None
        self.g = float('inf')
        self.h = float('inf')
        self.f = float('inf')
```

This code snippet defines a Cell class, which represents a cell on an area map.

Class attributes: x and y: cell coordinates on the map; obstacle: Indicates whether the cell is an obstacle. Default is False; parent: reference to the parent cell

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

in the path search tree. Used to restore the path; g, h, f: values of the cost functions for the path finding algorithm. By default they are set to infinity.

The __init__ method initializes the object's attributes when it is created. The x and y parameters specify the cell's coordinates on the map, and the obstacle parameter specifies whether the cell is an obstacle. The attributes g, h, f are initialized to infinity (float('inf')).

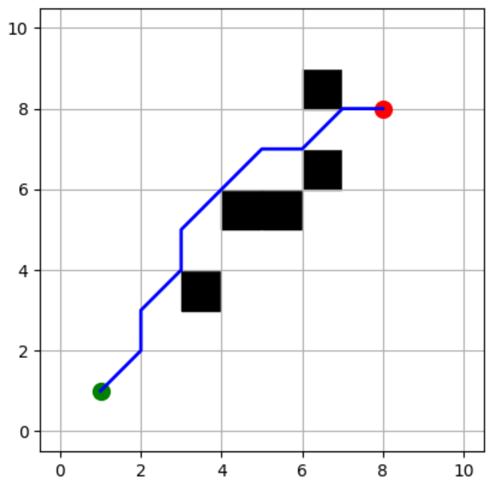
```
# Overriding the comparison operator "<" to compare Cell objects
    def __lt__(self, other):
        return self.f < other.f
```

This code snippet is an override of the < operator for Cell objects.

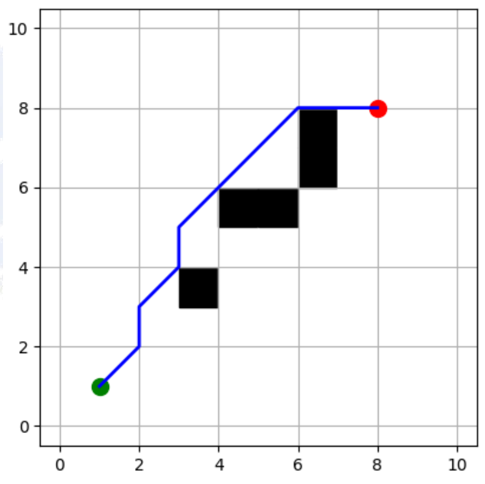
The < operator is used in Python to compare objects. Overriding this operator allows you to define comparison rules for class objects. In this case, the __lt__ method is overridden to compare Cell objects by their f value.

The return value of self.f < other.f means that if the f value of the current object (self.f) is less than the f value of the other object (other.f), then the current object will be smaller than the other object when compared.

The following hardware was used for research: CPU Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, RAM 16 Gb, GPU NVideo GeForce GTX 1660Ti (Ram 8Gb), Web-camera HD WebCam, OS Windows 10 Pro (Version 22H2). A program for implementing route finding and avoiding obstacles based on the A-Star algorithm was developed in the PyCharm 2022.2.3 (Professional Edition) environment in Python. The results of the program are presented in Figure 1.

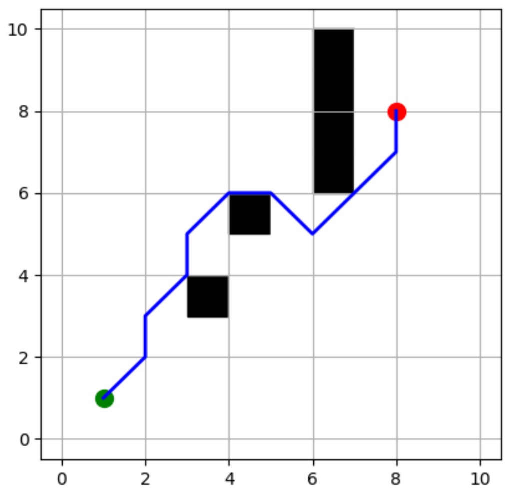


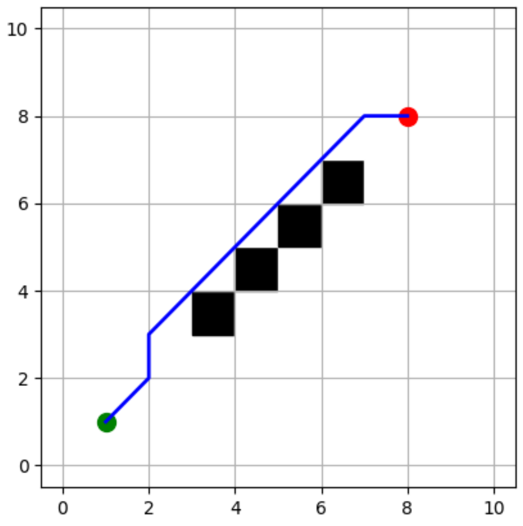
a)



b)

TECHNICAL SCIENCE RESEARCH IN UZBEKISTAN
ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3





c) d)

Figure 1: Results of the program for finding a route and avoiding obstacles based on the A-Star algorithm

However, for the successful application of the A* algorithm for mobile robots, several important points must be taken into account. First, it is necessary to correctly select the heuristic function for estimating the cost of achieving a goal so that it is informative and effective. Secondly, it is important to take into account the environmental features and dynamically changing conditions in order to avoid collisions with obstacles and ensure the safe movement of the robot. It is also recommended to use optimizations such as path caching and computation reduction to improve the performance of the algorithm.

Conclusion

The A* algorithm is an effective method for finding the optimal route and avoiding obstacles for mobile robots. It is based on finding a path in a state graph, where vertices represent possible robot positions and edges represent possible movements between these positions. The advantage of the A* algorithm is its ability to take into account both the cost of moving to the current position and a heuristic estimate of the cost of reaching the goal, which allows it to find the optimal path taking into account obstacles and constraints.

One of the key advantages of the A* algorithm is its ability to find the optimal path while using computing resources wisely. This makes it suitable for real-time applications on mobile robots. In addition, the A* algorithm has the optimality property, which means that it always finds the shortest path to the goal if such a path exists.

Overall, the A* algorithm is a powerful tool for finding the optimal route and avoiding obstacles for mobile robots. Its efficiency and optimality make it one of the most popular path planning methods in robotics.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

REFERENCES:

1. Attar, H., Abu-Jassar, A. T., Yevsieiev, V., Lyashenko, V., Nevliudov, I., & Luhach, A. K. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. Computational intelligence and neuroscience, 2022, 3046116.

2. Matarneh, R., Maksymova, S., Deineko, Z., & Lyashenko, V. (2017). Building robot voice control training methodology using artificial neural net. International Journal of Civil Engineering and Technology, 8(10), 523-532.

3. Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.

4. Maksymova, S., Matarneh, R., Lyashenko, V. V., & Belova, N. V. (2017). Voice Control for an Industrial Robot as a Combination of Various Robotic Assembly Process Models. Journal of Computer and Communications, 5, 1-15.

5. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

6. Sotnik, S., & Lyashenko, V. (2022). Prospects for Introduction of Robotics in Service. Prospects, 6(5), 4-9.

7. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

8. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

9. Lyashenko, V., Laariedh, F., Sotnik, S., & Ahmad, M. A. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal, 10(2), 583-591.

10. Lyashenko, V., & Sotnik, S. (2020). Analysis of Basic Principles for Sensor System Design Process Mobile Robots. Journal La Multiapp, 1(4), 1-6.

11. Lyashenko, V., Tahseen, A. J. A., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.

12. Matarneh, R., & et al.. (2019). Development of an Information Model for Industrial Robots Actuators. IOSR Journal of Mechanical and Civil Engineering, 16(1-V), 61-67.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

13. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.

14. Al-Sharo Y., & et al. (2023). A Robo-hand prototype design gripping device within the framework of sustainable development. Indian Journal of Engineering, 2023, 20, e37ije1673.

15. Akopov, M., & et al. (2023). Choosing a Camera for 3D Mapping. Journal of Universal Science Research, 1(11), 28-38.

16. Yevsieiev, V., & et al. (2022). A robotic prosthetic a control system and a structural diagram development. Collection of scientific papers «ΛΌГОΣ», Zurich, Switzerland, 113-114.

17. Nevliudov, I., & et al. (2023). A Small-Sized Robot Prototype Development Using 3D Printing. In XXXI International Conference CAD In Machinery Design Implementation and Educational Issues, 12.

18. Basiuk, V., & et al. (2023). Mobile Robot Position Determining Using Odometry Method. Multidisciplinary Journal of Science and Technology, 3(3), 227-234.

19. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical Science Research In Uzbekistan, 2(2), 115-124.

20. Igor, N., & et al. (2023). Using Mecanum Wheels for Radio Shuttle. Multidisciplinary Journal of Science and Technology, 3(3), 182-187.

21. Nevliudov, I., & et al. (2023). Mobile Robot Navigation System Based on Ultrasonic Sensors. In 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), IEEE, 1, 247-251.

22. Yevsieiev, V., & et al. (2023). A Small-Scale Manipulation Robot a Laboratory Layout Development. International independent scientific journal, 47, 18-28.

23. Bortnikova, V., & et al. (2019). Structural parameters influence on a soft robotic manipulator finger bend angle simulation. In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), IEEE, 35-38.

24. Babker, A., & Lyashenko, V. (2018). Identification of megaloblastic anemia cells through the use of image processing techniques. Int J Clin Biomed Res, 4, 1-5.

25. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3

State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.

26. Al-Sherrawi, M. H., Lyashenko, V., Edaan, E. M., & Sotnik, S. (2018). Corrosion as a source of destruction in construction. International Journal of Civil Engineering and Technology, 9(5), 306-314.

27. Lyashenko, V. V., Matarneh, R., Baranova, V., & Deineko, Z. V. (2016). Hurst Exponent as a Part of Wavelet Decomposition Coefficients to Measure Long-term Memory Time Series Based on Multiresolution Analysis. American Journal of Systems and Software, 4(2), 51-56.

28. Lyashenko, V. V., Deineko, Z. V., & Ahmad, M. A. Properties of wavelet coefficients of self-similar time series. In other words, 9, 16.

29. Khan, A., Joshi, S., Ahmad, M. A., & Lyashenko, V. (2015). Some effect of Chemical treatment by Ferric Nitrate salts on the structure and morphology of Coir Fibre Composites. Advances in Materials Physics and Chemistry, 5(1), 39-45.

30. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

31. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.

32. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform. International Journal, 2(8), 572-580.

33. Erke, S., & et al. (2020). An improved A-Star based path planning algorithm for autonomous land vehicles. International Journal of Advanced Robotic Systems, 17(5), 1729881420962263.

34. Ju, C., & et al. (2020). Path planning using an improved A-Star algorithm. In 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), IEEE, 23-26.

35. Zhang, Z., & et al. (2021). A-Star algorithm for expanding the number of search directions in path planning. In 2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), IEEE, pp. 208-211.

36. Hong, Z., & et al. (2021). Improved A-Star algorithm for long-distance off-road path planning using terrain data map. ISPRS International Journal of Geo-Information, 10(11), 785.

ISSN (E): 2992-9148 SJIF 2024 = 5.333
ResearchBib Impact Factor: 9.576 / 2024
VOLUME-2, ISSUE-3
37. Tang, G., & et al. (2021). Geometric A-Star algorithm: An improved A-Star algorithm for AGV path planning in a port environment. IEEE access, 9, 59196-59210.

38. XiangRong, T., & et al. (2021). Improved A-Star algorithm for robot path planning in static environment. In Journal of Physics: Conference Series, IOP Publishing, 1792(1), p. 012067.

39. Wang, Q., & et al. (2022). Application of A star algorithm in amphibious hull cleaning robot. In 2022 IEEE International Conference on Mechatronics and Automation (ICMA), IEEE, 269-273.

40. He, Z., & et al. (2022). Dynamic anti-collision A-Star algorithm for multi-ship encounter situations. Applied Ocean Research, 118, 102995.

41. Yudha, M. H. P., & et al. (2022). Optimalization Route to Tourism Places in West Java Using A-STAR Algorithm. CAUCHY: Jurnal Matematika Murni dan Aplikasi, 7(3), 464-473.

42. Mandloi, D., & et al. (2021). Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3d environment. International Journal of System Assurance Engineering and Management, 12(5), 990-1000.