

April 2010

Path Planning of Mobile Robot in Unknown Environment

Pradipta kumar Das

Dhaneswar Rath Institute of Engineering and Management Studies Tangi, Cuttack, Orissa-754022, India,
daspradipta78@gmail.com

Romesh Laishram

Manipur Institute of Technology Imphal, Manipur-795001, India, romeshlaishram@gmail.com

Amit Konar

Department of Electronics and Tele-Communication Engineering Jadavpur University, Kolkata-70032,
India, konaramit@yahoo.co.in

Follow this and additional works at: <https://www.interscience.in/ijcct>

Recommended Citation

Das, Pradipta kumar; Laishram, Romesh; and Konar, Amit (2010) "Path Planning of Mobile Robot in Unknown Environment," *International Journal of Computer and Communication Technology*. Vol. 1 : Iss. 2 , Article 5.

DOI: 10.47893/IJCCT.2010.1032

Available at: <https://www.interscience.in/ijcct/vol1/iss2/5>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Path Planning of Mobile Robot in Unknown Environment

Pradipta kumar Das
Dhaneswar Rath Institute of Engineering and
Management Studies
Tangi, Cuttack, Orissa-754022, India
daspradipta78@gmail.com

Amit Konar
Department of Electronics and Tele-Communication
Engineering
Jadavpur University, Kolkata-70032, India
konaramit@yahoo.co.in

Romesh Laishram
Manipur Institute of Technology
Imphal, Manipur-795001, India
romeshlaishram@gmail.com

Abstract— In this paper, we study the online path planning for khepera II mobile robot in an unknown environment. The well known heuristic A* algorithm is implemented to make the mobile robot navigate through static obstacles and find the shortest path from an initial position to a target position by avoiding the obstacles. The proposed path finding strategy is designed in a grid-map form of an unknown environment with static unknown obstacles. When the mission is executed, it is necessary to plan an optimal or feasible path for itself avoiding obstructions in its way and minimizing a cost such as time, energy, and distance. In our study we have considered the distance and time metric as the cost function.

Keywords- Robotics, Navigation, Real time A* algorithm, path planning.

I. INTRODUCTION

Motion planning is one of the important tasks in intelligent control of a mobile robot which should be performed efficiently. It is often decomposed into path planning and trajectory planning. Path planning is to generate a collision free path in an environment with obstacles and optimize it with respect to some criterion [1, 2]. However, this environment may be imprecise, vast, dynamical and either partially or non-structured [3]. In such environment, path planning depends on the sensory information of the environment, which might be associated with imprecision and uncertainty. Thus, to have a suitable motion planning scheme in a cluttered environment, the controller of such kind of robots must have to be adaptive in nature. Trajectory planning is to schedule the movement of a mobile robot along the planned path. Several approaches have been proposed to

address the problem of motion planning of a mobile robot. If the environment is a known static terrain and it generates a path in advance it said to be off-line algorithm. It is said to be on-line if it is capable of producing a new path in response to environmental changes. Path planning is the art of deciding which route to take for navigation under dynamic environment. The path planning is based on the current internal representation of the terrain. It involves many computations for continuous movement and sequences while moving between the start and end goal.

Many studies have been carried out in path planning for different types of mobile robots. The works in [4] and [5] involve mapping, navigation, and planning tasks for Khepera mobile robot. In these works, the computationally intensive tasks, e.g., the planning and mapping tasks were not performed onboard Khepera. They were performed on a separate computer. The sensor readings and the motor commands are communicated between the robot and the computer via serial connection. The A* algorithm [6-7] was used in these works for planning purposes. However the algorithm was applied only for an environment in which the locations of the obstacles are known in advance. For online path planning time metric is also an important parameter that cannot be avoided in designing the cost function. Recent researches [8-9] have also considered the use Genetic Algorithm (GA) in path planning for a static environment with static obstacles and control of robot motion using GA. The author in [10] also proposed the use of GA in the path planning of mobile robot in static environment with different fitness function. Some of the proposed techniques in [8-10] suffer from many problems. They include (1) computationally expensive (2) requires large memory spaces when dealing with dynamic and large sized environments, (3) time consuming. In [11-12] authors used Fuzzy Logic and Neural Network to handle the path planning problem, respectively.

In this paper the path planning problem of mobile robot is solved using the modified A* algorithm, well known as real time A* algorithm in unknown environment unlike [4-5]. The algorithm is executed in an online manner. The A* algorithm is always superior to other GA based solution in terms of the problems these algorithms suffer as mention above.

The rest of the paper is outlined as follows. Section 2 gives a brief description of Khepera II components and its operation. The algorithm for the path planning is described in detail in section 3. Section 4 provides the result through snapshot taken from the experiment and section 5 concludes the paper.

II. PROBLEM FORMULATION

Consider the environment, where the map of the environment is constructed as set of states is called a grid map and a grid map is build for the robot to navigate. In a grid map the mobile robot has to plan the path from its given initial position to goal position with an optimal way. In the grid map the robot can move in four neighbor directions vertically or horizontally. A* algorithm is used to decide the next best position. Each position has an associated cost function, $f(n)=g(n)+r(n)+h(n)$ where $g(n)$ is the generation cost or movement cost from the starting position to any position in the grid, $r(n)$ is the time taken for the robot to rotate towards the direction for movement and $h(n)$ is the estimated movement cost from a position to the target position. This is referred to as Heuristic cost. There are many different ways of defining the heuristic cost. In our implementation the Heuristic cost is the Euclidean distance between the next possible positions and the target position. The robot chooses the next position having minimum $f(n)$ value if there is no obstacle. If there is an obstacle the robot moves to the direction with next minimum $f(n)$ value. When more than one neighbor has same value of $f(n)$ then the direction is chosen randomly.

III. REAL TIME A* ALGORITHM

Typical A* algorithm works on a specialized search space represented by a tree or a graph. The objective of this algorithm is to identify a specific goal in the process of generating new states(nodes), ultimately terminating at the goal state. The path planning problem of the mobile robot can be solved by modified A* algorithm, well known as real time A* algorithm. To justify the importance of the algorithm, we consider the path planning of a mobile robot on a 2-D grid structure where a grid may contain one obstacle or the robot. Some of the grids in the workspace are empty and the robot has to make its pathway through these empty grid points in each iteration of the algorithm, so as to construct a trajectory of motion towards the prescribed goal. The significance of the real time A* algorithm lies in identifying the vacant until distant grid point which has the shortest distance of the given goal. This is usually evaluated by employing a heuristic function that keeps track of the Euclidean distance of a given neighborhood grid point from the given goal.

Fig. 1 presents a given obstacle map and the trajectory of the planned path of the robot. It is needless to mention here

that the robot traverses one of the neighborhood grid points in one iteration, but cannot move diagonally to the nearest grid points.

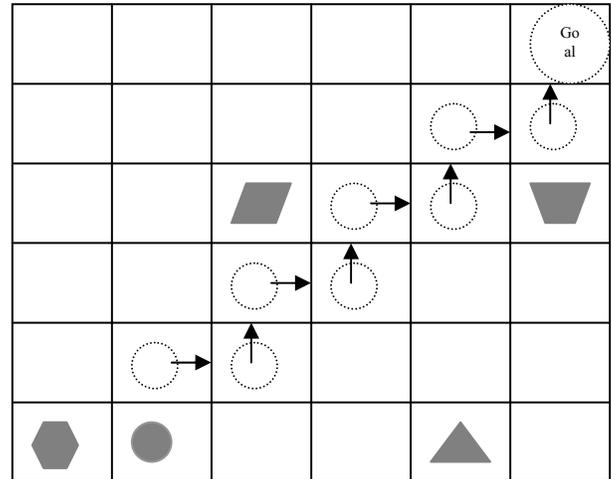


Figure 1. The motion planning of a robot amidst obstacle denoted by Polygons. A circle with a heading direction denotes the robot.

The real time A* algorithm is presented below

//Algorithm Real-Time-A*

1. Set NODE to be the start state.
2. Generate the successor of NODE. If any of the successors is a goal state, then quit.
3. Estimate the value of each successor by performing a fixed-depth search starting at that successor. Use depth-first search. Evaluate all leaf nodes using the A* heuristic function $f = g + h'$, where g is the distance to the leaf node and h' is the predicted distance to the goal. Pass heuristic estimates up the search tree in such a way that the f value of each internal node is set to the minimum of the values of its children.
4. Set NODE to the successor with the lowest score, and take the corresponding action in the world. Store the old NODE in a table along with the heuristic score of the second-best successor. (With this strategy, we can never enter into a fixed loop, because we never make the same decision at the same node twice.) If this node is ever generated again in step 2, simply look up the heuristic estimate in the table instead of redoing the fixed-depth search of Step 3.

Go to step 2.

IV. REALIZATION OF THE PROBLEM

The realization of the problem is carried out using Khepera II robot[13], Khepera II robot (figure 2) is a miniature robot (diameter of 8cm) equipped with 8 built-in infrared range and light sensors, and 2 relatively accurate encoders for the two motors[7]. The range sensors are positioned at fixed angles and have limited range detection capabilities. We numbered the sensors clockwise from the leftmost sensor to be sensor 0 to sensor 7 (figure 2). Sensor values are numerical ranging from 0 (for distance > 5 cm) to 1023 (approximately 2 cm). The onboard Microprocessor has a flash memory size of 256KB, and the CPU of 8 MHz and Terapro software is used to connect the Khepera robot via RS232 serial cable or via telnet to the desktop machine. KTproject is used as an editor for writing the code in 'C' language for Khepera and produce the dot s37 compiled file of the 'C' code and the dot s37 file will be download into the Khepera through the Terapro software. This software is used for interfacing the robot and user through desktop machine. The Fig. 2 shows network connection of the robot with desktop machine through the terapro software.

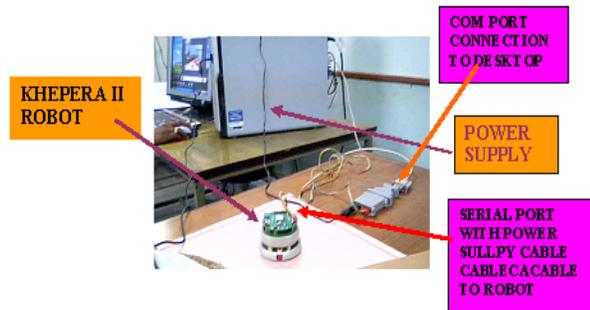


Figure 2. Khepera network and its accessories

A* is a graph search algorithm that finds a path from given initial node to a given goal node (or one passing a given goal test). The A* algorithm stands by combining the greedy search and uniform-cost search (Dijkstra algorithm) [14]. Greedy search minimizes the estimated cost to the goal (that is the heuristic, $h(n)$), and thereby cuts the search cost considerably; but it is neither optimal nor complete. In the other hand, uniform-cost search minimizes the cost of the path so far (that is elapsed cost, $g(n)$); it is optimal and complete, but can be very inefficient. These two methods can be combined by simply summing the two evaluation functions to get advantages of both methods.

$$f(n) = g(n) + h(n) \quad (1)$$

By using the above heuristic cost of the nodes, the heuristic cost of more than one node is same, so it cannot give the optimal path to goal. Hence, we have modified the equation (1) by considering the heading direction as the cost in the heuristic function. The modified heuristic function as follow:

$$f(n) = g(n) + h(n) + r(n) \quad (2)$$

$g(n)$ is the generation cost of the node i.e. no of arc from starting node to 'n' node.

$h(n)$ is heuristic cost of the node; here we consider the Euclidean distance between two points.

$r(n)$ is the cost of rotating the heading direction towards the direction for movement.

By using the evaluation function, A* algorithm can enhance the search speed of the best-first algorithm; Overcome the shortcoming on the searching precision of the local optimal search algorithm. A* algorithm uses less memory space than Dijkstra algorithm, the average out degree of one node is marked as 'b', the search depth of the shortest path from start point to end point noted as 'd', then the time complexity of A* algorithm is represented as $O(b^d)$.

In the path planning problem for finding the optimal path, A* algorithm starts from a point, START (source), to another point, GOAL (destination). A* keeps a list of all the possible next steps, called the OPEN list. It then chooses a next step that is most likely to lead us to the goal in the minimum time. In order to accomplish this we need to have a heuristic that will determine "most likely". Once that step has been chosen, it is moved to the CLOSED list.

The algorithm for the path planning is shown below.

Path_planning ()

```
{
Initialize: Place the robot in the grid map environment,
heading towards y-axis
    Specify the target or goal position (x, y) for the
    robot.
    heading direction =FRONT
```

While (goal_not_reach)

```
{
Calculate  $f(n) = g(n) + h(n) + r(n)$  // cost function for all the next
possible positions.
Find the minimum of all four possible costs and select the
direction of minimum  $Dir_{min}$ 
```

V. EXPERIMENTAL RESULTS

Switch (Dir_{min}) // Dir_{min} = 0 or 1 or 2 or 3 for front, right, left and down respectively

```

{
Case 0:
    Turn the robot heading in FRONT direction
    of its current pose
    if (obstacle (FRONT) )
        Select the next  $Dir_{min}$  direction and allow
        the robot to move.
    Else
        Move in FORWARD direction.

Case 1:
    Turn the robot heading in RIGHT direction
    of its current pose.
    if (obstacle (RIGHT) )
        Select the next  $Dir_{min}$  direction and allow
        the robot to move
    Else
        Move in FORWARD direction

Case 2:
    Turn the robot heading in LEFT direction of
    its current pose
    if (obstacle (LEFT) )
        Select the next  $Dir_{min}$  direction and allow
        the robot to move
    Else
        Move in FORWARD direction

Case 3:
    Turn the robot heading in selected direction
    of its current pose
    if (obstacle(DOWN) )
        Select the next  $Dir_{min}$  direction and allow
        the robot to move
    Else
        Move in FORWARD direction
}
}
Else
    move in left direction
}
default:
}
}

```

Extensive experiments were conducted to test the robot performance in path planning between any two positions i.e. START and GOAL position in the unknown environment with various unexpected obstacles. Figure 3 shows the environment set up of the path planning. The environment has three static obstacles of different shape and size. Figure 4 shows the snapshots taken at different stages of the Khepera II mobile robot path planning in one such case study. Figure 5 gives the grid map representation of the path planning by the mobile robot for the environment shown in figure 3.



Figure 3. Environment setup for robot path planning



(i)



(ii)



(iii)



(iv)

Figure 4. Snapshots taken at different stages of path planning

- (i) Initial movement of the robot
- (ii) - (iii) Intermediate movement of the robot
- (iv) Final position of the robot at the Goal

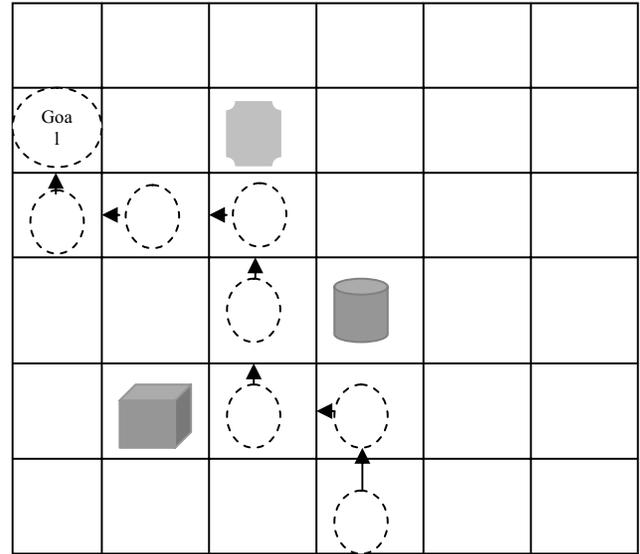


Figure 5. Robot path planning in a Grid map environment

VI. CONCLUSION

In this paper, online path planning of mobile robot in unknown environment with static obstacles is presented using Khepera II mobile robot. The path planning is performed using Heuristic real time A* algorithm. The use of A* algorithm can meet the rapid and real-time requirements of path planning which otherwise is not possible in some path planning using advanced algorithms. However only static obstacles were considered in the present study.

The result from this work provides a platform for developing robot control and also provides a useful tool for robotics education. For future work, we plan to include dynamic obstacles and to include the onboard camera in the study of online path planning of mobile robot.

REFERENCES

- [1] A. Howard, M.J Matari and G.S.Sukhatme, "An Incremental Self-Deployment Algorithm for mobile Sensor Networks, autonomous robots", *special Issue on Intelligent Embedded Systems*, 13(2), September 2002, pp. 113-126.
- [2] S. Florczyk, "Robot Vision Video-based Indoor Exploration with Autonomous and Mobile Robots", *WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim*, 2005.
- [3] Danica Janglova, "Neural Networks in Mobile Robot Motion, pp. 15-22, *International Journal of Advanced Robotic Systems*, Volume 1 Number 1 (2004), ISSN 1729-8806.
- [4] Harlan, R. M, Levine, D. B., and McClarigan, S., "The Khepera robot and the kRobot class: a platform for introducing robotics in the undergraduate curriculum", *proc. 32nd SIGCSE technical symposium on computer science education*, pp. 105-109. 2001.
- [5] Harlan, R. M. and McClarigan, S., "Creating emergent behaviors: two robotics labs that combine reactive behaviors", *proc. 36th SIGCSE technical symposium on computer science education*. Feb. 2005, pp. 441- 445.
- [6] Murphy, R., "Introduction to AI Robotics" first edition, the MIT press, 2000.
- [7] Amit Konar, "Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain", 1st edition, CRC Press, 1999.
- [8] C. E. Thomas, M. A. C. Pacheco, M.M. and B.R.Velasco, "Mobile Robot Path Planning Using Genetic Algorithms", in *foundations and tools for neural modeling*, vol. 1606/1999, Springer Berlin/ Heidelberg, ISBN: 3-540-66069-0, 1999, pp. 671- 679.
- [9] J. Lu and D. Yang, "Path planning based on double-layer genetic algorithm", in *3rd Int. Conf. on Natural Computation (ICNC 2007)* [Online], Hangzhou, Haikou, China, Aug. 24-27, 2007, Vol. 4, pp: 357-361
- [10] I. AL. Taharwa and A. Sheta and M. AI. Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341- 344, 2008.
- [11] A. Ramirez-Serrano and M. Boumedine, "Real time navigation in unknown environments using fuzzy logic and ultrasonic sensing," in *Proc. 1996 IEEE Int. Symp. In Intell. Control*, Dearborn, MI, ISBN: 0-7803-2978-3, Y de Estudios superiores de Monterrey, 15-18 Sept, 1996, pp: 26-30, DOI: 10.1109/ISIC.1996.556172.
- [12] I.K. Jung, K.B. Hong, S.K. Hong and S.C. Hong, "Path planning of mobile robot using neural network," in *Proc. IEEE Int. Symp. On Ind. Electron. Sch. Elect. Engg.*, 1999 ISIE Bled Slovenia, Vol. 3, 12-16 July, 1999, pp. 979-983, DOI: 10.1109/ISIE.1999.796750.
- [13] <http://www.k-team.com/robots/khepera/index.html>
- [14] Wang Feng, YOU Sheng-zhi, Application of Dijkstra and Dijkstra-based n-Shortest Paths algorithm to intelligent transportation systems [J]. *Application Research of Computer*, 2006, 23(9):2 03-206(in Chinese)