International Journal of Scientific Research in Computer Science, Engineering and Information Technology
 | Volume 2 | Issue 6 | ISSN : 2456-3307

Incremental Robot Path Planning With L* Algorithm, For Unknown Environments

Helga V Lobo*

Department of Computer Science Engineering, Centre for P.G Studies, VTU, Belagavi, India

ABSTRACT

This paper proposes a method for Incremental path planning using L* algorithm for building path incrementally when the information obtained by robots sensors is incomplete. A grid is incrementally built by the robot as it moves towards the destination, using the partial information it obtains. The implementation was carried out using MATLAB programming tool, which has proved the correctness of the proposed method.
Keywords: Incremental path planning, Unknown environment, L*.

I. INTRODUCTION

In robot path planning, it becomes essential for the robot to have complete information of its surroundings in order to build the path to the given goal. However, in reality it is not possible to have complete prior information of the surroundings due to limitation in robots sensors. Hence, it is required for the robot to move using the partial information and simultaneously build the complete map as it moves towards the goal. Several algorithms like Dijkstras, A*, and D* have been already implemented which may be used to find path between two points in a known environment [3]. The problem associated with these algorithms is increase in time complexity with increase in size of search area [1][2]. These problems are addressed by L* algorithm, providing linear computational complexity with increased size of the search area [1].

The basic L* algorithm is a variant of A* algorithm. The L* algorithm proposed by Adam Niewola et. al [1] maintains sub-lists rather than a single list of final costs. In the proposed method, L* algorithm is used to work in partially known environments, as it performs better than A* and its variants [1][2].The paper presents a method that is capable of and incrementally and robustly planning a optimal path using partially available information. The proposed method is implemented and verified to prove the correctness of the method.

The rest of the paper is laid out as follows. Section II presents methodology. Section III presents working of incremental path planning. Section IV presents the experimentation results and Section V brings up the conclusion.

II. METHODOLOGY

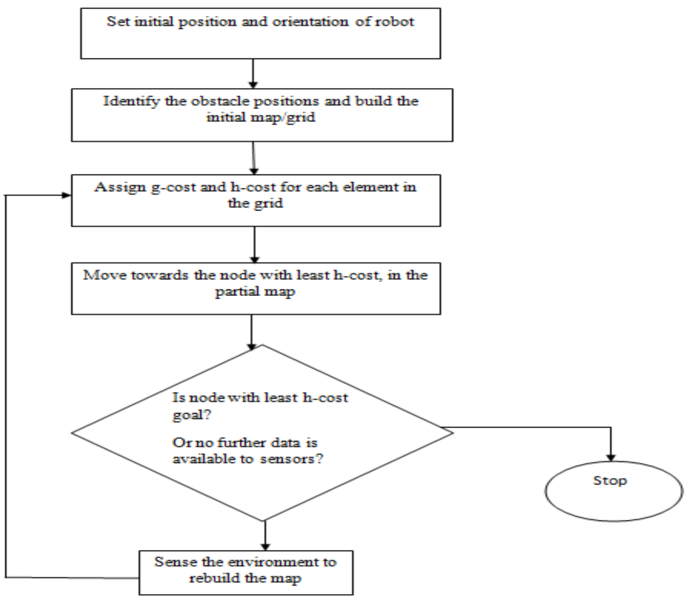

Figure 1. Steps involved in methodology

The flow chart shows the steps involved in incremental path planning. To plan the path, initially the robots initial position and orientation are set. The path-planning module then builds the primary map with the incomplete information it obtains. Then the path is traced by the module to move towards the node with least h-cost. The search terminates if this node is a goal or no further information is available to sensor; else the robot further senses the environment from the relocated position to rebuild the map.

III. INCREMENTAL PATH PLANNING

A. L* Algorithm [1]

The proposed method makes use of L* algorithm[1] to plan path in partially generated map. The L* algorithm creates multiple sub-lists rather than a single list of final costs[1][2]. Sub-list *i* will contain the nodes whose final cost is between (h0+(i-1)*df) and (h0+i*df). The first node extracted by the algorithm is the one from first sub-list, which is non-empty. The nodes are extracted one by one until the goal is found or all the sub-lists get empty. Heuristic estimate of a node to the goal is taken to be the Euclidian distance.

B. Obstacle Sensing

Initially the robots initial position and orientation is set. From this initial position, the robot senses the environment to generate the primary map. The map generated will be usually partial due to sensors limitations and obstacles present in the environment. Maximum area covered by the sensors during this sensing operation is considered a square, whose half diagonal is less than or equal to the maximum range of sensor. This is done to ensure the correctness of sensors data. For the obstacle at distance *d*, the coordinates are generated as follows,

$$(x, y)=(d*\cos\upsilon, d*\sin\upsilon)$$

Where *d*- is the distance to the obstacle as perceived by the sensor.

υ-is the rotation of the robot from the initial orientation.

C. Building the primary map

In this method, workspace is modelled using grid map or cell decomposition method [4]. Each unoccupied cell is taken to be approximately by 1.5 times the size of the robot. This ensures that the robot does not collide with any obstacles and move freely in each cell.

***Height/width of cell=1.5*diameter of the robot*.**

D. Building path to node, which has least h-cost in the partial map obtained

In the partial map obtained the h-costs are assigned to each cell. The h-cost is the heuristic cost, which is taken to be the Euclidian distance to goal. The robot is made to move towards the node with least h-cost in the partial map obtained. In case the goal is present in the partial map, the search terminates; else the robot further senses the environment to rebuild map resetting the node with least h-cost to be its source.

E. Building the complete path

This section explains building path incrementally, using L* algorithm in each updated map.

Algorithm 1. Incremental Map update and path generation

1. Choose the source, Set the orientation of the robot, obtain the primary map by sensing the environment and identify the node with least h-cost in the map.
2. Move towards the node with least h-cost using L* algorithm
    if node with least h-cost is not the goal
        go to step 3
    else
        exit
3. ***while*** the node with least h-cost is not the goal or no further data is available to the sensors
        for small increments of angle or change in orientation of robot
            identify the distance(d) of obstacle present
            *obstacle(x, y)=(d*cosυ, d*sinυ)*
            Update the map
            move towards the node with least h-cost with L* algorithm.
            if node with least h-cost is goal or no further data is available to sensors
                stop search, goal can not be reached
            else
                set the node with least h-cost to be source
            end if
        end for
    end while

IV. IMPLEMENTATION RESULTS

The proposed method was implemented using MATLAB programming. For given co-ordinates of source and destination the path was traced. Following figures show the map built in incremental steps and path traced from source to destination incrementally.

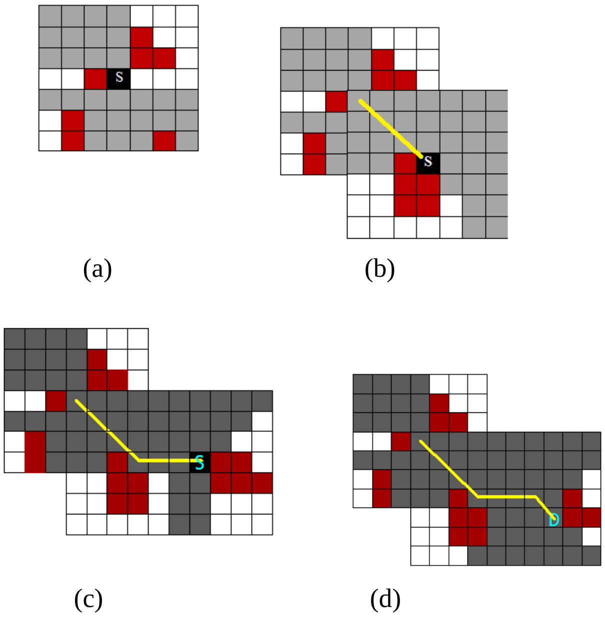

Figure 2. (a)Primary map with cell containing robot marked in the centre.(b)Path traced and updated map after robot senses the environment second time.(c) Path traced and updated map after robot senses the environment third time.(b) Path traced and updated map after robot senses the environment fourth time.(destination reached)

The red areas in the map represent blocked cells due to obstacles present. The grey areas represent the map built by the robot. Unshaded cells are unreachable by the sensor due to obstacles present or the limitations in the sensors, which may be covered in the subsequent steps. It can be observed from the map that the goal is not perceived by the sensors initially, it is perceived in the subsequent sensing as the map is built incrementally.

V. CONCLUSION

This paper has presented a method for robot path planning in unknown environments, where the robots search area is built incrementally. The method assumes no dynamic behaviour of any obstacles after they are sensed during the search process. Hence, the goal may not be reached by the robot if obstacles are in motion. This problem may be tackled by further sensing of environment by the robot during every step of its move, in the partial map obtained.

VI.REFERENCES

[1] Adam Niewola, Leszek Podsędkowski, "Nonholonomic Mobile Robot Path Planning with Linear Computational Complexity Graph Searching Algorithm*" , Proceedings of the 10th International Workshop on Robot Motion and Control, Poznan University of Technology, Poznan, Poland, July 6-8, 2015

[2] Dr.S.A.Angadi, Helga V Lobo, "Robot Path Planning with L * algorithm for Linear Computational Complexity, considering distance andSlope" IJIREEICE, vol 5,no.10, August 2017

[3] Sariff, N. and Buniyamin, N.; "An Overview of Autonomous Mobile Robot Path Planning Algorithms", IEEE 4th Student Conference onResearch and Development, 2006.

[4] Christian Scheurer & Uwe E. Zimmermann, "Path Planning Method for Palletizing Tasks using Workspace Cell Decomposition," ICRA Communications, 2011.

[5] X.-C. Lai, C.-Y. Kong, S. S. Ge, and A. A. Mamun, Online map building for autonomous mobile robots by fusing laser and sonar data,.Proceedings of 2005 IEEE International Conference on Mechatronics and Automation, pp. 993.998, July 2005.

[6] A. Yahja, S. Singh, and A. Stentz, .An ef_cient on-line path planner for outdoor mobile robots operating in vast environments,. Robotics and Autonomous Systems, vol. 32, no. 2&3, pp. 129.143, 2000.

[7] A. Stentz, .The focussed d* algorithm for real-time replanning,. Proceedings of International Joint Conference on Arti_cial Intelligence,1995