Improved Variable-Step-Length A* Search Algorithm for Path Planning of Mobile Robots

By

Md. Hasibul Hasan
ID: 151-35-873

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Software Engineering

Department of Software Engineering
DAFFODIL INTERNATIONAL UNIVERSITY

Spring 2018

APPROVAL

This **Thesis** titled "**Improved Variable-Step-Length A* Search Algorithm for Path Planning of Mobile Robots**", submitted by **Md. Hasibul Hasan**, **151-35-873** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head — **Chairman**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Dr. Md. Asraf Ali
Associate Professor — **Internal Examiner 1**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

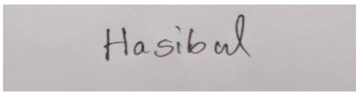
Mohammad Khaled Sohel
Assistant Professor — **Internal Examiner 2**
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Prof Dr. Mohammad Abul Kashem
Professor — **External Examiner**
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

Thesis Declaration

I hereby declare that, this Thesis Report has been done under the supervision of Md. Anwar Hossain, Senior Lecturer, Department of Software Engineering, Faculty of Science and Information Technology, Daffodil International University. I also declare that this hasn't been submitted elsewhere for award of any degree.

Submitted By



………………………………...

Md. Hasibul Hasan

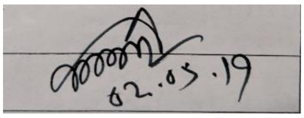
ID: 151-35-873

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

Certified By



…………………………………

Md. Anwar Hossain

Senior Lecturer

Department of software Engineering

Faculty of Science and Information Technology

Daffodil International University

Acknowledgement

Alhamdulillah, all praises to the Almighty Allah who gives me the ability, motivation, patience and wisdom to complete this research work.

I would like to propagate my gratefulness and gratitude to my respectful supervisor Md. Anwar Hossain. This research would not be possible without him. His excellent guidance, motivation, caring, patience and providing me with an excellent facilities and environment for doing this research. I am also grateful to my parents for their support and care throughout the entire period.

Abstract

The use of Mobile Robots in different sectors are increasing day by day. Mobile robots are widely used in industrial sectors nowadays. In different rescue missions people use mobile robots. In order to perform better there needs to be a good path planning process for these mobile robots. A very long time ago A* search algorithm was proposed as a better way of finding a path from source to destination. But in today's world it seems not very likely to perform that better. And for the betterment of path planning this algorithm was modified and improved in many ways. And still the research is going on. One of the research work variable-step-length A* search which can take steps of length larger than one. It produces more optimal path than A* Search. But it uses fixed step length throughout the whole process. But it is possible to get more optimal path if the step length can be changed throughout the process when necessary. So, a modified variable-step-length A* search algorithm is proposed in this research which can update or adjust step length. This algorithm can find a path with less path length.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Path planning refers to a series of action in order to break down the process of reaching from one place to another in an accurate and efficient way. Path planning is a key issue in the growth of mobile robotics. It is absolutely necessary to ensure accuracy and collision or obstacle avoidance for path planning of mobile robot. There are a lot path planning algorithms available. Each one of them is best for some particular performance measurement keys. The research is still going on to find a better solution in the field of mobile robotics to improve the effectiveness of the path planning for mobile robots. The objective of this research is also finding a better solution for the path planning of mobile robots.

1.2 Research Motivation

In the field of mobile robotics, it is very essential to plan an optimal path for the smooth movement of mobile robots. There are many methods and algorithms proposed for this purpose throughout the entire time. Variable-Step-Length A* search algorithm is one of the effective method to solve the path planning problem. But there is a chance to improve that algorithm to do work better. Motivated buy this idea, in this research, an improved Variable-step-length A* search algorithm has been proposed.

1.3 Problem Statement

Path planning is a key issue in the field of mobile robotics. An optimal path planning process ensures the reachability of mobile robots to its target. The main challenge of path planning is to ensure the accuracy and efficiency. It is absolutely necessary to ensure the accuracy and efficiency at the same time. A path will be considered optimal if it uses less resources and gives an accurate process of movement to reach the target. Throughout this whole process, it is a must to ensure that no obstacles have gone through the path.

1.4 Research Question

Path planning is an inevitable part of the working procedure of a mobile robot. Performance of a mobile robot varies due to the effectiveness of the path planning algorithm. There exists plenty of path planning algorithms. But all of them are not ideal for all conditions or all performance measurements keys. We will try to find out the answer to the following questions.

- The path length is already pretty much optimized. Can we actually reduce the total path length from starting to the goal further?

- Can we reduce the total number of moves for reaching the target?

1.5 Research Objectives

The objective of this research is to modify the existing path planning algorithms in a way such that, it can do the path planning with more effectiveness. In order to do that, the following objectives are taken under consideration.

- Take a long straight step where it is possible. This will help to reduce the total path length because of going through a straight line.

- Avoid unnecessary small steps. This will help to reduce the number of steps.

1.6 Research Scope

The main focus of this research is to improve process of path planning. Path planning is an inevitable thing in order to function a mobile robot successfully. It is also necessary for automated vehicles. An effective path planning can change the game in the field of mobile robots and automated vehicles. In this research, the main target is to improve the path planning algorithm so that it can reduce the path length further.

1.7 Thesis Organization

This report contains the contents in the following manner.

Chapter 1 contains the field of this research and the concept and the objective of this research. The purpose and possible outcome of this research are stated in this chapter.

In chapter 2, the much needed topics that should be in the consciousness for this research are stated. It contains some concepts of the related previous works.

Chapter 3 consists of the previous works in this field and the research works on the main concept of this research.

In chapter 4, the related previous work and the new proposed process is described. This chapter also contains the major improvement scheme of the previous algorithm.

Chapter 5 contains the test results and further discussions.

Chapter 2

Background Study

2.1 Mobile Robots

A robot that has the ability to move from one place to another is a mobile robot. Mobile robots are fixed to one physical location. Mobile robots are mostly being used to complete industrial tasks. But the use of mobile robots is growing in different sectors day by day. Mobile robot is the collaboration of advancement in artificial intelligence and physical robotics. In most cases mobile robots are operated by software and use sensors and actuators to discover the environment and take actions according to the commanding software.

Usually there are two types of mobile robots.

- **Autonomous:** Autonomous mobile robots have the ability to move around and explore the environment without outer guidance. Autonomous mobile robots are considerably performs better in dynamic environment.

- **Non-Autonomous:** Non-Autonomous mobile robots need external guidance contains a pre-defined set of rules and operations to complete its tasks.

2.2 Path Planning of Mobile Robots

The term "Path Planning" refers to the process of discovering a way or passage from one location to another through some transit points. Path planning is necessary to find not only just any path but also the optimal path between two points. An effective path planning reduces the amount turning and movements of a mobile robot.

There are several types or techniques of path planning algorithms. They are –

- Artificial Potential Field: In this technique, each unit place of the map of the environment is being assigned a potential value. The value is being distributed according to the distance to the goal from that place. The value gets higher to nearest places and gets lower to the distant places from the goal. The obstacles are assigned with a very remarkably low value. But with this technique, the robots often get trapped.

- Grid Based: The total map gets divided into small square grids. Each grid is considered as a unit place. A robot can stand in any grid in any particular time except the obstacle grids. Then to find the path, a search algorithm used to discover grids from start to goal.

- Reward Based: In this technique the actions of the robots are not predefined. The robot can take actions on its own. So the result of an action is partially random depending on the behavior and decision of the robot. The robot gets positive rewards for reaching target or for taking an action that is predefined as productive. And gets negative reward for taking an action that is predefined as disadvantageous or for colliding with an obstacle. The algorithm finds out the path through these series of actions that maximizes the rewards in reaching the goal.

2.3 Environment Model built by Grid Method

In 1968, W. E. Howden came up with the grid method [1]. The grid method is the most popular and most extensively used method for constructing the environment model of mobile robots. In this method, it is considered that mobile robot moves in a finite area with finite numbers of static obstacles. The fundamental idea is to divide that finite area into small grid units. Every grid is a square and considered as one unit grid. An obstacles situated in a grid fills the whole grid. The mobile robot can be placed in a unit grid and leaves clear distance from the border of the grid.

Figure 1 demonstrates the grid method environment model. The blank grids are free grids and it can accommodate a mobile robot. The black grids are blocked the obstacles. The mobile robot can't be placed in a grid that contains obstacles and also can't get through that grid.

This grid method is applied for this research.

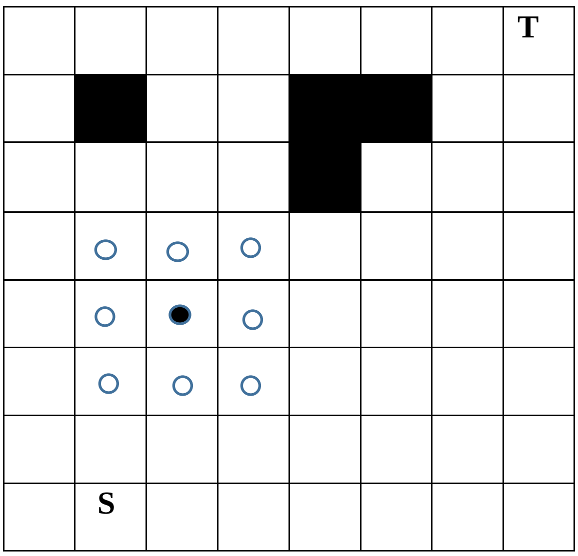


Fig. 2.1: Environment model built by Grid Method

2.4 A* Search

A* search is a extensively known path finding algorithm for also used for graph traversal. A* search gradually develops a path between two points by using the idea of graph traversal.

The path planning process starts at the starting node. At each step A* search considers the 8 adjacent nodes and calculate an evaluation function for those nodes by considering some parameters. The evaluation function is written as f (n).

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

g (n) is the movement cost to node **n** from the current node.

h (n) is the estimated cost to move from the node **n** to the target node.

In the next step, the adjacent node which has least value of the evaluation function f(n) will be set as current node.

This process will go on until the target cell is reached.

2.5 Variable-Step-Length A* Search

Variable-step-length A* search is an improved version of the traditional A* search algorithm. This algorithm takes a radius **R** at the beginning. And at each step it considers all the nodes that within the radius R. It finds the reachable cells inside the radius of the current cell depending on the position condition. The position condition verifies whether the cell is between the current cell and the target cell.

The evaluation function is also slightly updated.

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

Here, g' (n) = g(n) * w(n)

$$w(n) = \frac{\sum_{k=1}^{m} \sqrt{(x_c - x_k)^2 - (y_c - y_k)^2} - \sqrt{(x_c - x_n)^2 - (y_c - y_n)^2}}{\sum_{k=1}^{m} \sqrt{(x_c - x_k)^2 - (y_c - y_k)^2}}$$

Here, **m** is the number of reachable nodes from the current node.

Chapter 3

Literature Review

Since the introduction of A* search algorithm for path planning of mobile robot, further research is still going on to get better performance from A* search.

Akshay Kumar Guruji et al. (2016) [2] introduces Time-Efficient A* search algorithm. The proposed algorithm works like the following procedure. First take an image of the robot's working arena and transform it into binary image and assign starting, destination point and mark the obstacles. Then calculates the distance between starting and destination. If the distance between starting and destination point is considerably less then it follows the traditional A* search. Otherwise follows a process called switching phase. This procedure is being followed from current cell or point of each step. It projects pixel and from current cell and depending on slope finds from where the slope changes direction. And proceed to the point just before it. This process continues until it reaches destination point.

Daniel Drake et al. (2018)[5] introduces an algorithm of path planning of mobile robots with moving goal. At first, the algorithm start expanding node from the goal and set the cost from goal to the expanded or visited nodes and give unique id to this nodes to identify later. Then expand nodes from the previous expanded ones. If the cost is less than the previous one update the lower cost. If two nodes are compared and the cost doesn't change, then the nodes are from different search tree. And by following this process the initial path is determined. When the goal is changed, the algorithm searches around the old goal to find the new goal. Then star to expand from the new goal until finds an old search tree. When it finds an old search tree, it can complete the rest of the path from there.

Ke Da1 et al. (2017)[1] introduces variable-step-length A* search algorithm which is a modified version of traditional A* search. Unlike the traditional A* search this algorithm doesn't use always step of length 1. This algorithm fixed a step length or radius (R) before starting the process of path planning. And at each step considers the points that are within the radius R form the current position as next potential cell.

M.S.Ganeshmurthy et al. (2015)[4] introduces an offline path planning algorithm. This algorithm made significant changes in the process of path planning. This proposed algorithm first finds a path form the starting point to the destination point and then tries to improve that path by using simulated annealing. This algorithm uses temperature cooling process with simulated annealing to find a better path. Whenever the algorithm gets a better path from the previous one it definitely updates the new one. But if it gets a path having larger length than the previous one it goes for it if the probability of taking that path at that time if less than the temperature which is cooling down from the beginning of the process.

Thi Thoa Mac et al. (2016)[6] made an analysis among the different algorithms which are used to do path planning. In this research the performance of different algorithms in different situations are discussed and measured in different environments. This research includes Fuzzy Logic, Neural Network and some other most common algorithm that are used to solve the path planning problem.

Mingxiu Lin et al. (2017)[3] introduces a modified and improved version of A* search algorithm. They proposed a new way to calculate evaluation function for each cell by updating the equation of actual cost function. They updated the actual cost function with the value of its parent's actual cost function value.

Chapter 4

Methodology

This chapter contains the main working procedure of the proposed improved version of the variable-step-length A* search algorithm.

4.1 Variable-Step-Length A*

Variable-step-Length A* (VSLA) works like the traditional A* search. The only major difference is, VSLA uses step length greater than or equal to 1 which is predefined before starting the process. A* only use steps of length 1. In VSLA, a radius is set to do the process. A* search considers the adjacent 8 cells for the next potential cell where VSLA considers the cells that are within the predefined radius R. VSLA also uses some adjustments in the actual cost function of A* Search.

For better understanding, The Grid Method will be used as the environment model which is described in previous section.

4.1.1 Searching Direction

The traditional A* search searches the 8 neighbor cells to set as next cell. But in VSLA a region R is set to consider the cells within R from current cell.

Suppose, the radius is set to R and the current cell, target cell and searchable cell is (x_c, y_c) , (x_t, y_t) and (x_a, y_a) respectively. The cell (x_a, y_a) will be considered as reachable if and only if the two following conditions are satisfied.

- $distance_{ac} = \sqrt{(x_a - x_c)^2 - (y_a - y_c)^2} < R$

- $\begin{cases} x_c \leq x_a \leq x_t & if\ x_c < x_t \\ x_t \leq x_a \leq x_c & otherwise \end{cases}$

  Or

- $\begin{cases} y_c \leq y_a \leq y_t & if\ y_c < y_t \\ y_t \leq y_a \leq y_c & otherwise \end{cases}$

4.1.2 Improved Actual Cost Function

As VSLA is taking step of length larger than 1, the actual cost function **(g (n))** should be adjusted. The heuristic function **(h (n))** will remain the same as the traditional A* search. The main idea of adjustment is to give consideration to the farer cells from the current cell [1]. So, the adjusted actual cost function **(g’ (n))** will be:

$$g'(n) = w(n) * g(n)$$

g (n) is the same Euclidian distance just like the traditional A* search. **w (n)** is the weight function.

$$w(n) = \frac{\sum_{k=1}^{m} \sqrt{(x_c - x_k)^2 - (y_c - y_k)^2} - \sqrt{(x_c - x_n)^2 - (y_c - y_n)^2}}{\sum_{k=1}^{m} \sqrt{(x_c - x_k)^2 - (y_c - y_k)^2}}$$

Here, m is the number of reachable cells from the current cell.

4.1.2 Obstacle Avoidance

From section 4.1.2 it is likely to find a better path planning algorithm. But is still a large factor to avoid obstacles and reach the goal. Suppose the current cell, target cell, searchable cell and obstacle cell is (x_c, y_c) , (x_t, y_t) and (x_n, y_n) (x_b, y_b) respectively.

If the cell (x_n, y_n) satisfies the two following conditions, then it means the obstacle grid (x_b, y_b) is in between the cell (x_c, y_c) and(x_n, y_n).

- The slope condition

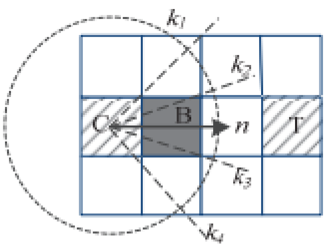


Fig. 4.1.1: Slope for obstacle avoidance

$$k_1 = \frac{(y_b + 0.5) - y_c}{(x_b - 0.5) - x_c}$$

$$k_2 = \frac{(y_b + 0.5) - y_c}{(x_b + 0.5) - x_c}$$

$$k_3 = \frac{(y_b - 0.5) - y_c}{(x_b + 0.5) - x_c}$$

$$k_4 = \frac{(y_b + 0.5) - y_c}{(x_b - 0.5) - x_c}$$

$$k = \frac{y_n - y_c}{x_n - x_c}$$

Min $(k_1, k_2, k_3, k_4) < k <$ Max (k_1, k_2, k_3, k_4)

•

$$distance_{cb} = \sqrt{(x_c - x_b)^2 - (y_c - y_b)^2}$$
$$distance_{cn} = \sqrt{(x_c - x_n)^2 - (y_c - y_n)^2}$$
$$distance_{ct} = \sqrt{(x_c - x_t)^2 - (y_c - y_t)^2}$$

$$distance_{cb} < distance_{cn} < distance_{ct}$$

4.1.3 Procedure

VSLA works in the following way.

1. At first, set the radius R.
2. Set the starting cell as current cell.
3. Now consider the cells that are inside the radius from the current cell and find the reachable cells depending on position condition.
4. Then removes the cells which aren't reachable because obstacles between those cells and the current cell.
5. Calculates evaluation function for the remaining cells.
6. Choose the cell with the least evaluation function value as next cell.
7. Set the next cell as current cell.
8. If the current cell is target, then breaks. Otherwise continue again from step 3.

The following example will be helpful to understand the nature of VSLA.

This will just illustrate the process of going to the target from starting cell through cell changing.

The environment is shown as a 10X10 grid. The starting S is in cell (4.5, 4.5) and the target is in cell (7.5, 8.5). The radius for this is R = 2. The reachable cells are marked grey and the black marked cells are obstacles.

Now, set the starting as current cell. So, the current cell is (4.5, 4.5)

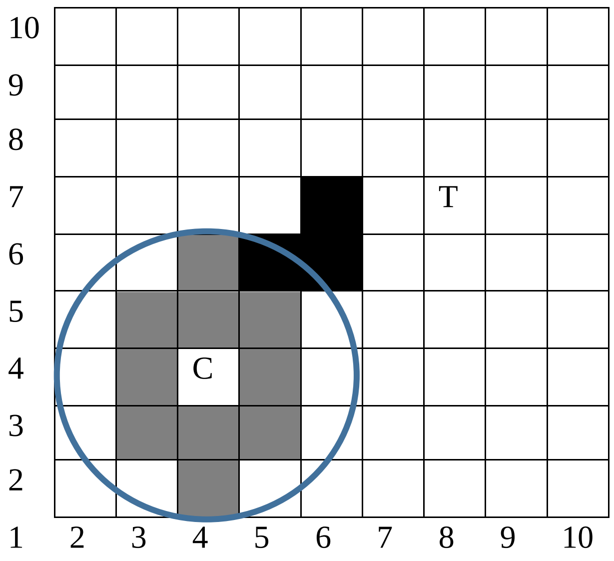


Fig 4.1.2: C– Current Cell, T – Target Cell, Reachable – Grey Marked, Obstacles – Black Marked

In figure 4.1, the cells that are fully inside the radius 2 from current cell are the reachable cells (Grey Marked). From these reachable cells find which is most convenient to reach the target. The most convenient cell is (5.5, 5.5). Set that convenient cell as current cell. So, now the current cell is (5.5, 5.5).

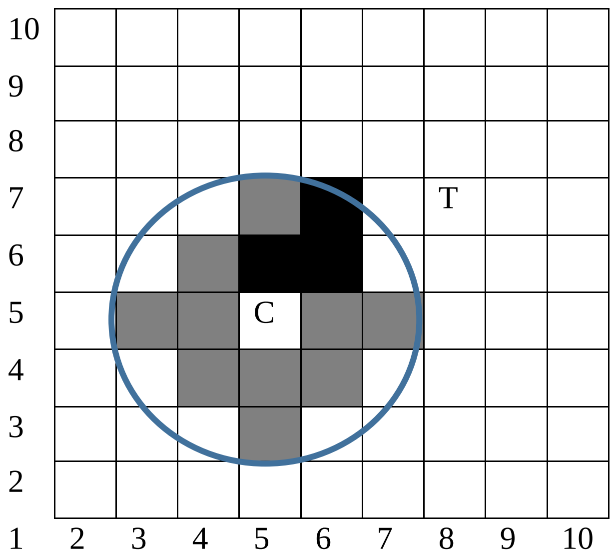


Fig 4.1.3: C– Current Cell, T – Target Cell, Reachable – Grey Marked, Obstacles – Black Marked

In figure 4.2, from (5.5, 5.5), the most convenient reachable cell to reach target is (5, 7.5). So, now set cell (5.5, 7.5) as current cell.

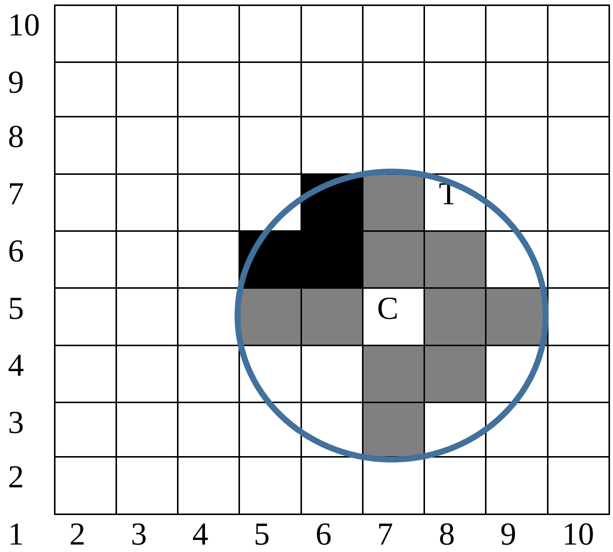


Fig 4.1.4: C– Current Cell, T – Target Cell, Reachable – Grey Marked, Obstacles – Black Marked

In figure 4.3, from (5.5, 7.5), the most convenient reachable cell to reach target is (6.5, 8.5). So, now set cell (6, 7) as current cell.

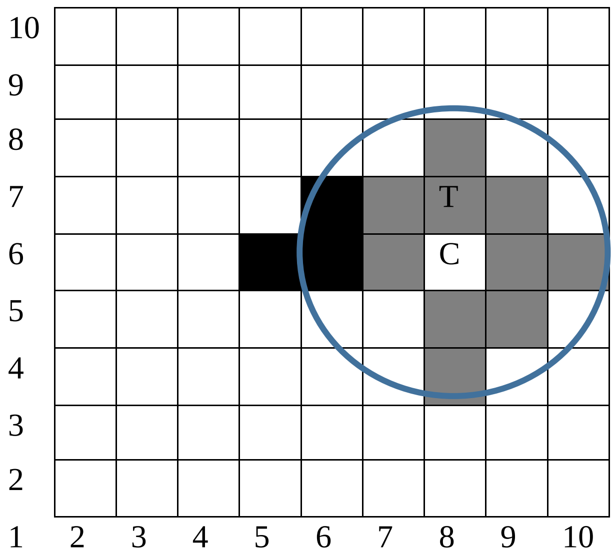


Fig 4.1.5: C– Current Cell, T – Target Cell, Reachable – Grey Marked, Obstacles – Black Marked

In figure 4.4, from (6.5, 8.5), the most convenient reachable cell to reach target is (7.5, 8.5). So, now set cell (7.5, 8.5) as current cell.

Now, the current cell is the target cell. So, path is complete.

So the path planned by VSLA is (4.5, 4.5), (5.5, 5.5), (5.5, 7.5), (6.5, 8.5), (7.5, 8.5)

The path length is = 5.8284

4.2 Improved Variable-Step-Length A*

Variable-Step-Length A*(VSLA) is doing a great job in reducing the path length. It is reducing a significant amount of path length. But it is still possible to reduce the path length further.

VSLA uses a fixed radius or step length during the whole process of path planning. This fixed radius don’t change or update during a single path planning process. There can be a scenario where it is possible to take a step of length greater than the fixed radius. In that scenario, though it is possible to take a large step but the algorithm can’t determine or can’t take that large step because of the fixed step length or radius. Because of this, the step count and path length isn’t reducing at the most significant level. There can another scenario where is not possible to take a step of the predefined step length because of obstacles or other reasons. If there is way to change or update the step length when necessary, the algorithm could work better.

This is where the Improved Variable-Step-Length A* (IVSLA) comes into the picture. The main idea of IVSLA is not to bind the searching radius by a fixed radius or step length before staring the process. There is a flexibility to choose the step length during each step of the algorithm. When it is necessary or possible to take a large step, the algorithm will go for it. And when it is optimistic to take a smaller step, the algorithm will do that too.

4.2.1 Considering partially reachable cells

There might be a scenario where in a step of VSLA the target cell is partially inside the radius. But as VSLA doesn’t consider the cells that are partially inside the region, the path length will increase the number of steps will increase. So, it is optimistic to consider the cells that are partially within the radius.

4.2.2 Variable Variable Steps

As it is clear that there are some certain scenario where the fixed step length doesn't give the optimized result always. So, the step length won't be a fixed value in IVSLA throughout the whole process. The optimal step length will be chosen at each step of the process. The detailed process of choosing step length will be described in the next section.

4.2.3: Calculation for a single step

```
Initialize the Radius
Identify the reachable cells and store in reachable list that are within R
Identify the obstacles that are within R
Initialize the bestEvaluationValue to MAX_VALUE
Initialize temporary cell with both row and column value -1
For i = 1 to reachable list size
        Calculate the evaluation value for cell I and store in a variable Temp
        If Temp is less than bestEvaluationValue
                Set bestEvaluationValue to Temp
                Set temporary cell to cell i
        End If
End For
Return temporary cell
```

4.2.4: Updating the step length

```
Initialize the starting cell as current cell.
Initialize the path length to zero.
Initialize the step count to zero.
While current cell is not starting cell
        Set the minRadiusLength = 1 and maxRadiusLength = ceil (Euclidian distance between
        current cell and target cell)
        Initialize temporary cell with both row and column value -1
        While minRadiusLength is less than or equal to maxRadiusLength
                Set mid = (minRadiusLength + maxRadiusLength) / 2
                If it is not possible to take step of length equal to mid then
                        Set maxRadiusLength = mid – 1
                Else
                        Set temporary cell as the cell which can be set as the next cell inside step
                        length of mid
                        Set minRadiusLength = mid+1
                End If
        End While
        If the temporary cell is -1 or the temporary cell is equal to Current cell
                Notify Trap detected.
                Break
        End If
        Add the Euclidian distance between current cell and temporary cell with the path length
        Increase the value of step count by one.
End While
Return the path length and step count
```

Chapter 5

Result and Discussion

This chapter discusses the performance of the Improved Variable-Step-Length A* (IVSLA) against the Variable-Step-Length A* (VSLA) and A* Search.

The following test results are for IVSLA. These results will be compared with VSLA and A* later in this section.

Test 1:

Number of Row, R = 5

Number of Column, C = 10

Start Cell – (1.5, 1.5)

Target cell – (5.5, 8.5)

Path Length – 9.5765

Time Taken – 0.005

Number of Steps - 7

Test 2:

Number of Row, R = 25

Number of Column, C = 25

Star Cell – (1.5, 2.5)

Target Cell – (25.5, 20.5)

Path Length – 31.6673

Time Taken – 0.096

Number of Steps – 16

Test 3:

Number of Row, R = 20

Number of Column, C = 20

Star Cell – (1.5, 1.5)

Target Cell – (20.5, 20.5)

Path Length – 28.3755

Time Taken – 0.035

Number of Steps – 11

Test 4:

Number of Row, R = 20

Number of Column, C = 20

Star Cell – (2.5, 8.5)

Target Cell – (20.5, 20.5)

Path Length – 23.6786

Time Taken – 0.014

Number of Steps – 8

Test 5:

Number of Row, R = 45

Number of Column, C = 50

Star Cell – (3.5, 48.5)

Target Cell – (22.5, 1.5)

Path Length – 57.7754

Time Taken – 0.87

Number of Steps – 30

Test 6:

Number of Row, R = 45

Number of Column, C = 50

Star Cell – (3.5, 48.5)

Target Cell – (41.5, 34.5)

Path Length – 41.9318

Time Taken – 0.090

Number of Steps – 6

Test 7:

Number of Row, R = 50

Number of Column, C = 55

Star Cell – (4.5, 24.5)

Target Cell – (28.5, 10.5)

Path Length – 40.1172

Time Taken – 0.220

Number of Steps – 15

Grid Scale	Algorithm	Reachable field	Step Number	Step Reduction rate	Avg. Path Length	Path Reduction Rate	Avg. Convergence Time	Time Reduction Rate
5 X 10	A*		12		11.50		0.010	
	VSLA	R = 3	8	33.34%	9.83	14.53%	0.006	40%
		R = 6	7	41.67%	9.58	16.7%	0.007	30%
	IVSLA		7	41.67%	9.57	16.8%	0.005	50%
45 X 50	A*		38		51.01		0.095	
	VSLA	R = 3	34	10.53%	48.77	4.39%	0.051	46.31%
		R = 6	31	18.42%	46.87	8.12%	0.060	36.84%
	IVSLA		6	84.67%	41.93	17.8%	0.08	15.78%

Table 1. Comparison of performance between A*, VSLA and IVSLA algorithms.

Chapter 6

Conclusions and Recommendations

The objectives of this thesis are to improve the Variable-Step-Length A* (VSLA) search algorithm in way such that it can find a path with less path length and complete the process in less number of steps. In chapter 5, the test results highlights that the target and objective of this research have been successfully achieved. The Improved Variable-Step-Length A* (IVSLA) can successfully reduce the total path length and the number of steps to reach target from a starting point. In the proposed method of VSLA, there was a fixed radius for the whole process. In this research a method is proposed where the radius isn't fixed anymore. The radius can vary depending on the situation.

This newly proposed modified algorithm can find the optimal path in any situation. But in the environments where the ratio of obstacles is pretty high, this algorithm takes a bit more time than the previous one.

References

1. K. Da, L. Xiaoyu, Z. Bi, "Variable-step-length A* Algorithm for path planning of mobile robot", Control And Decision Conference (CCDC) 2017 29th Chinese, pp. 7129-7133, 2017.

2. A. K. Guruji, H. Agarwal, and D. K. Parsediya, “Time-efficient A∗ algorithm for robot path planning,” Procedia Technology, vol. 23, pp. 144–149, 2016.

3. Mingxiu Lin, Kai Yuan, Chengzhi Shi , Yutong Wang. Path Planning of Mobile Robot Based on Improved A* Algorithm, 2017.

4. M.S.Ganeshmurthy, Dr.G.R.Suresh. Path Planning Algorithm for Autonomous MobileRobot in Dynamic Environment, 2015.

5. D. Drake, S. Koziol, E. Chabot, "Mobile robot path planning with a moving goal", IEEE Access, vol. 6, pp. 12800-12814, 2018.

6. Thi Thoa Mac, Cosmin Copot, Duc Trung Tran, Robin De Keyser. Heuristic approaches in robot path planning: A survey, 2016.

7. A. Stentz, "Optimal and efficient path planning for partially-known environments", Proc. IEEE Int. Conf. Robot. Autom., pp. 3310-3317, 1994.

8. F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, L. Jurisica, "Path planning with modified a star algorithm for a mobile robot" in Procedia Engineering, Modelling of Mechanical and Mechatronic Systems, vol. 96, pp. 59-69, 2014.

9. S. M. Sombolestan, A. Rasooli, and S. Khodaygan. Optimal path-planning for mobile robots to find a hidden target in an unknown environment based on machine learning. Journal of Ambient Intelligence and Humanized Computing, Mar 2018.