Line based Geometrical Path Planning Algorithm for Extinguishing Forest Fires

M. P. Sivaram Kumar
Research Scholar/CSE
B S Abdur Rahman University
Vandalur,Chennai-48

S. Rajasekaran
Professor/Mathematics
B S Abdur Rahman University
Vandalur, Chennai-48

ABSTRACT
In this work a mathematical approach based path planning of actor which uses equation of a straight line passing through two points is proposed to extinguish forest fires. Forest domain is considered as a square area and is tessellated into n x n cells (small squares) of desired size. The Actor which performs actions similar to the actions performed by a robot is assumed to be available at one cell which is considered as one point and the cell in which fire occurs is considered as the second point. The equation of straight line is found out using these two points. Then using the straight line equation the sequence of intermediate cells between the cell where the Actor is available and the cell where the fire is occurred is computed and stored in memory. Then a check is made for presence of obstacle in the intermediate cells. If there are obstacles present then a cell without obstacle and also nearer to obstacle cell is found out and replaced in the memory. Integrity check is performed to make sure that successive points are incremental points. i.e. whether the next point can be reached in one move. Then the sequence of points stored in the memory is used by the Actor to move to the cell where fire is occurred and after reaching that cell the actor will start to extinguish fire. Computer simulation results shows that the algorithm works well in all aspects for environment with and without obstacles and it computes the path for the actor to extinguish Forest fire without collision with obstacles.

General Terms
Forest fires, Path Planning, Straight Line Equation passing through two points.

Keywords
Forest fires, Path Planning, Artificial Intelligence.

.

1. INTRODUCTION
The direction-finding of a mobile robot to begin from a point which is called as start point and move through a sequence of intermediate points to reach a point called goal point is considered as most significant subject in the robotic applications. The direction finding can be done in offline or online. In this work we use offline direction-finding.i.e.The intermediate cells which avoid collision with obstacles are computed before the actor starts to move. Plenty of works are available in the literature based on conventional approaches and plausible approaches. Also works based on Genetic Algorithm, Neural Networks and ant colony optimization algorithms are available in plenty in the literature [1, 2, 3, and 4]. In this work we have proposed a Mathematical approach based Path Planning of Actor using Straight Line equation to extinguish Forest fires. The path finding is implemented in two phases. In the first phase the sequence of intermediate cells is computed and the locations of intermediate cells are stored in consecutive memory locations. In the second phase alternate cells without obstacles for the intermediate cells which contain obstacles is found and replaced in the memory. The remainder of the section is organized as follows: In Section2 we have presented the modeling and representation of forest domain. Section 3 describes Algorithm for path planning of the actor to extinguish fires. In section 4 the simulation results are discussed while section 5 concludes the paper.

2. Modeling and Representation of Forest
1. Forest domain is tessellated into n x n square grids or cells[1].
2. Anchor sensor is deployed at the center of the each cell [1]
3. Each sensor is assumed to know its Location coordinates based on I quadrant [1].
4. The decomposition of forest using 5 x 5 grids with coordinates based on I quadrant is shown in fig 1.

0,4	1,4	2,4	3,4	4,4
0,3	1,3	2,3	3,3	4,3
0,2	1,2	2,2	3,2	4,2
0,1	1,1	2,1	3,1	4,1
0,0	1,0	2,0	3,0	4,0

Fig 1 Decomposition of forest using 5 x 5 grids with coordinates based on I quadrant

5. The actor is available at the bottom and its coordinates (0, 0) are start point
6. The cell in which fire occurs is assumed to be the goal point.
7. The size of the obstacle is at the maximum size of one cell.
8. Two obstacles are allowed consecutively either length wise or breadth wise but not both.
9. Once fire occurs inside a cell the corresponding sensor will send location coordinates to the actor. There by the actor knows both start and goal positions. Hence path has to be found by actor to reach to that cell in order to extinguish fire.
10. Initially Actor is assumed to face towards east

3. Algorithm for Path Planning
The path for the actor to extinguish Forest fires is computed in two phases. In the first phase the sequence of intermediate cells through which the actor can navigate is computed using equation of straight line and the locations of the cells are stored in memory. In the second phase presence of obstacle in the intermediate cells is checked and if there are obstacles then alternate cells without obstacles is computed and

replaced by the cell location in the memory. After completion of these two phases the cell locations available in the memory are free from obstacles and it is used by the actor for making moves to reach the cell where the fire occurred and then the actor start to extinguish fire. In this work we have modified the work presented in [5] and also we have considered a multi point robot so that the robot can make rotations to change its direction. The algorithm for the two phases is shown below

3.1 Algorithm for Computing Sequence of Intermediate Cells

Let (x1, y1) be the coordinates of the cell where the Actor is located and (x2, y2) are the coordinates of the cell where the fire is occurred. Then sequences of intermediate cells is calculated based on three cases as mentioned below

Case (i): when x1 = x2

a. When x1 is equal to x2 and y1 < y2 then simply increment the value of y1 till it reaches the
value of y2.
i=1
Store x1, y1 in i th location of the memory
While (x1 = x2 and y1 < y2)
i=i+1
y1=y1+1
Store x1, y2 in i th location of memory

b.When x1 is equal to x2 and y1 > y2 then simply decrement the value of y1 till it reaches the
value of y2.
i=1
Store x1, y1 in i th location of the memory
While (x1 = x2 and y1 < y2)
i=i+1
y1=y1-1
Store x1, y2 in i th location of memory

Case (ii): when x1 < x2

When x1 is less than x2 then the sequence of intermediate points is computed using straight line equation passing through the two points which is calculated using the formula

$$\frac{y-y1}{y2-y1}=\frac{x-x1}{x2-x1} \quad ---1$$

Then after simplification of equation 1 by substituting the values of (x1, y1) and (x2, y2) we will get an equation of the form

$$y = mx + c \quad ----2$$

Where m and c are constant values.

Then keeping the base value of x = x1 and substituting in equation 2 we can find the corresponding y-point. This will form one intermediate point. The process is repeated by incrementing the value of x1, assigning it to x and finding the corresponding y values upto the incremented value of x1 is same as x2. Immediately after the calculation of intermediate point, it will be stored in the memory.

i=1
Store x1, y1 in i th location of the memory
While (x1 is less than x2)
x=x1
Substitute the value of x in equation 2 and find the corresponding y value
Round the value of y.
i=i + 1
Store the (x, y) coordinates in the i th Location of the memory
x1=x1+1
End while

Case (iii): when x1 > x2

The algorithm is almost same as case (ii) except the value of x1 is decremented each time.

i=1
Store x1, y1 in i th location of the memory
While (x1 is less than x2)
x=x1
Substitute the value of x in equation 2 and find the corresponding y value
Round the value of y.
i=i + 1
Store the (x, y) coordinates in the i th Location of the memory
x1=x1-1
End while

3.2 Algorithm for checking whether intermediate cells contains obstacles and replacing obstacle cell by another cell

```
Location =2
While (location! = (goal location – 1))
 Begin
   Read x and y coordinates from the location
   If the ((x, y) contains obstacle) then
    Begin
        If up cell (x, y) is free from obstacle then
            Replace x, y value by up cell x, y value in the
            memory
        else
            Replace x, y value by right cell x, y value in the
            memory
        end if
    End
   End if
Location = Location +1
End
End while
```

3.3 Integrity Check

After Calculating the sequences of Intermediate points a check is made to make sure that two successive intermediate points are incremental points i.e. whether the next point can be reached in one move. This can be performed by checking the truth value of the logically connected and condition (((x2-x1) <=1) and ((y2-y1) <=1)). If the condition is true they are incremental points otherwise they are not incremental points. If the points are not incremental a call is made to algorithm which finds sequence of intermediate points. The process is repeated till the points found out are sequence of intermediate points. We have also made sure that call to the algorithm is not going into infinite loop by making some modifications in the program.

3.4 Algorithm for computing the change in the direction of movements of Robot

After performing the integrity check the actor (robot) has to start from the source cell where it is available and move through these intermediate cells to reach the goal cell where fire has occurred and start extinguishing fire. Initially the actor is facing towards east. The actor has to make necessary changes in the direction of movements while moving from one cell to another cell. The direction an actor can move from current cell to another cell is based 8-point Neighborhood as shown in the fig1.The direction the actor has to move is determined using the value. The values and the corresponding directions are shown below in Table 1. Change in direction of movements from one cell to another cell is calculated as follows:

NW	N	NE
W	Current position	E
SW	S	SE

Fig 1 Possible directions Actor can Move

First take the first cell and the next cell locations from memory. Let it be (x1, y1) and (x2, y2)
Initialize the facing number of the actor to be 1.Using the next cell number where the robot has to make a move and the facing cell number (i.e. the direction, in which currently the robot is facing), the angle to be made by the actor is calculated.

Table 1 Direction Actor will Move based on value

Value	Direction Actor has to move
1	E-East
2	NE-North East
3	N-North
4	NW-North West
5	W-West
6	SW-South West
7	S-South
8	SE-South East

The algorithm for calculating the next cell number is shown below

If ((x2- x1) = 0 and (y2-y1) >0) then
nextcellno = 1
If ((x2- x1) > 0 and (y2-y1) >0) then
nextcellno = 2
If ((x2- x1) > 0 and (y2-y1) =0) then
nextcellno = 3
If ((x2- x1) > 0 and (y2-y1) < 0) then
nextcellno = 4
If ((x2- x1) = 0 and (y2-y1) < 0) then
nextcellno = 5
If ((x2- x1) < 0 and (y2-y1) < 0) then
nextcellno = 6
If ((x2- x1) < 0 and (y2-y1) =0) then
nextcellno = 7
If ((x2- x1) < 0 and (y2-y1) >0) then
nextcellno = 8

Once the next cell number is calculated, the angle to be made by the actor is calculated using the formula given below
Angle = (c – f) * 45 ---------- 3
Where c – Target cell number
f- Current facing of the robot
When the value of the angle is positive the robot has to rotate itself in anticlockwise direction and when it is negative it has to rotate itself in clockwise direction by the specified angle value. Once the angle is calculated and robot reaches the next cell the facing number is changed. The angle which the actor has to make is shown below

Cell no /Facing no	1	2	3	4	5	6	7	8
1	0	-45	-90	-135	-180	-225	-270	-315
2	45	0	-45	-90	-135	-180	-225	-270
3	90	45	0	-45	-90	-135	-180	-225
4	135	90	45	0	-45	-90	-135	-180
5	180	135	90	45	0	-45	-90	-135
6	225	180	135	90	45	0	-45	-90
7	270	225	180	135	90	45	0	-45
8	315	270	225	180	135	90	45	0

4. Simulation Results

In this work we have considered the forest domain as a grid decomposed into m x n cells based on co-ordinates of First Quadrant. Also we have assumed that fire can occur at any point inside the domain but only one occurrence of fire at a time is permitted since we have only one actor for the entire forest domain. We have formulated a path planning algorithm for the actor based equation of straight line obtained using the two points (start and goal points) in order to extinguish fire in both types of environment. i.e. Environment with and without obstacles. We have used MATLAB for simulation purposes. The forest is tessellated into a 10 x 10 grid. The Robot is represented by a green square, the cells containing obstacles are represented using black color circles and the cell where fire occurs is shown in red color circle and sequence of diamonds in green color shows the path planning of the Actor to travel and reach the target area to extinguish fire. To test the efficacy of the proposed algorithm we have created fire in

various cells and by varying the obstacle positions. The Computer simulation result shows that in all the cases intermediate sequence of cells which constitute a path is created using the algorithm by avoiding collision with and without obstacles. The simulation results are shown below:
Case (i): x1=x2

In figure 2 the points are (3,3) and (3,8) . Obstacle is not present in this scenario

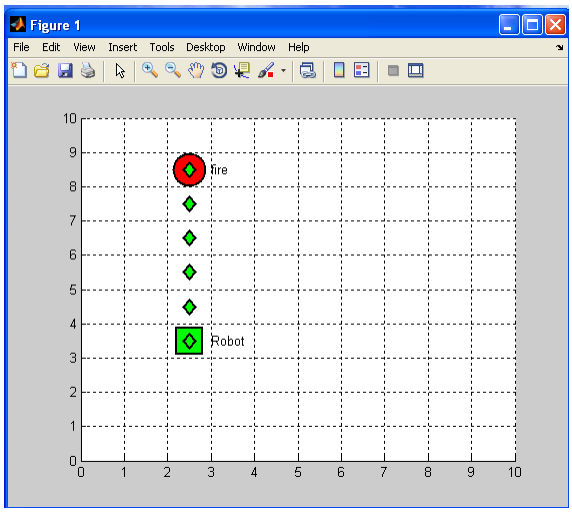

Fig 2 Path without obstacles for x1 = x2

In figure3 the points are (3,3) and (3,8) . Obstacle is present in this scenario

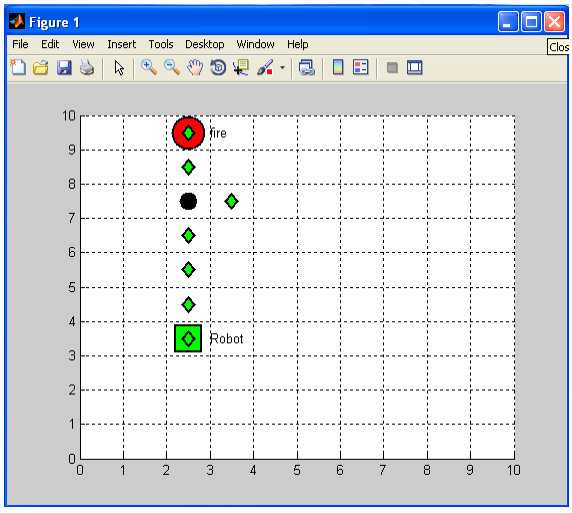

Fig 3 Path with obstacles for x1 = x2

Case (ii): x1 < x2

In figure 4 the points are (3,4) and (8,9) .Obstacle is not present in this scenario.
In figure 5 the points are (3,4) and (8,4) . Obstacle is present in this scenario. The Location of the obstacle is (7, 4). Since the Condition x1 < x2 is satisfied it uses the straight line equation to find the sequence of intermediate points

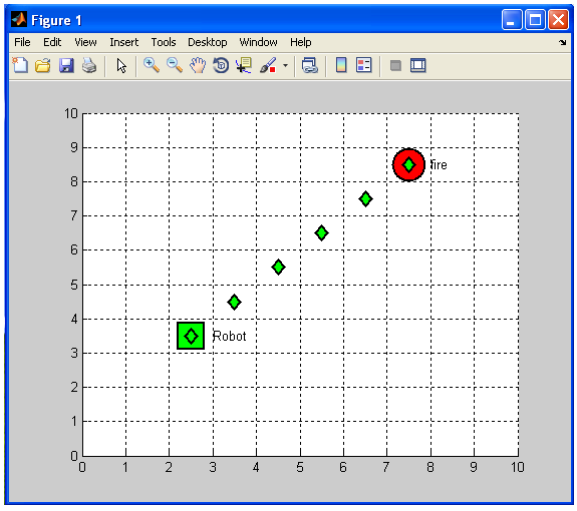

Fig 4 Path without obstacles for x1 < x2

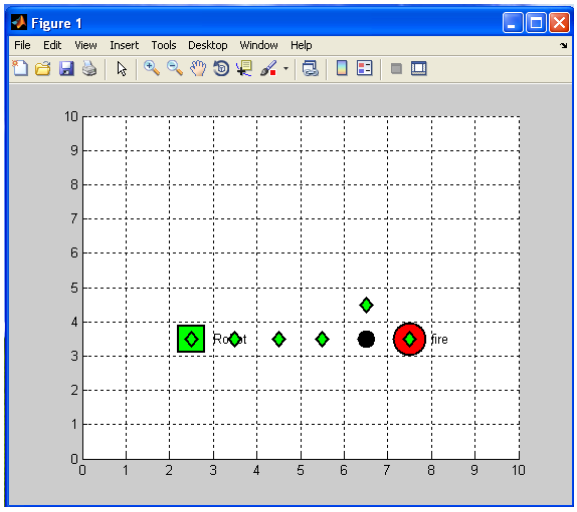

Fig 1 Path with obstacles for x1 < x2

Case (iii): x1 > x2

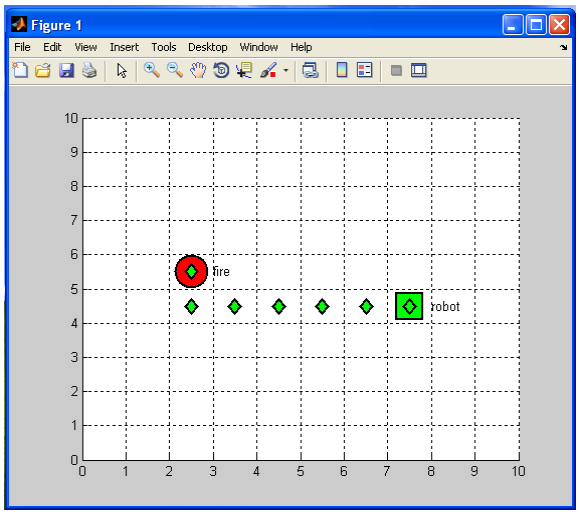

Fig 5 Path without obstacles for x1 > x2

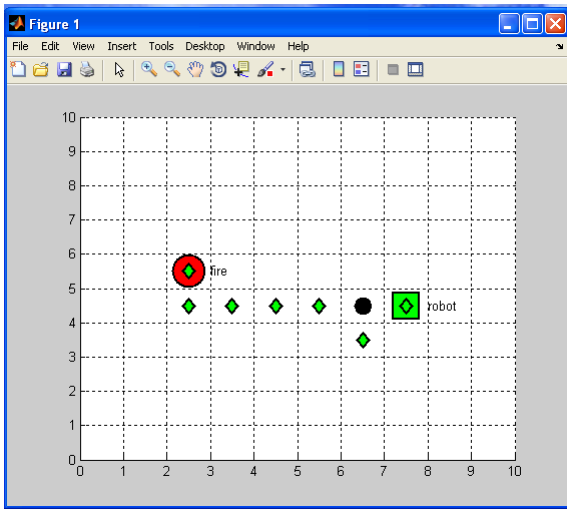

Fig 5 Path without obstacles for x1 > x2

5. CONCLUSION

In this paper we have presented a straight line equation based path planning algorithm for the actor to move to the target cell where the fire occurred and starts to extinguish fire. In this work we have assumed that robot can move in all directions based on 8-point neighborhood. Since the robot can move in all 8-directions we have also given algorithm for robot to perform rotation in order to move to the next cell from the current cell. Since we have found out the intermediate points by using straight line equation we have preformed integrity check to make sure that the moves are incremental moves. This simulation will work for any number of obstacles placed in any position. But because of page constraints we have shown only limited simulation results. In this work we have considered only static obstacles but in future we have plans to incorporate as many dynamic obstacles as possible and also consider time taken for the robot to move towards the target area and also the time taken for the rotation of robot in order to perform next move. Also we plan to compare our work in terms of performance with other algorithms in the forth coming paper.

6. REFERENCES

[1] M.P.Sivaram Kumar, S.Rajasekaran, "Detection and Extinguishing Forest Fires using Wireless Sensor and Actor Networks", International Journal of Computer Application,Vol.24 (1), June 2011, pp.31-35.

[2] M.P.Sivaram Kumar, S.Rajasekaran, "Path Planning Algorithms for Extinguishing Forest Fires using", Journal of Computing", Volume 4, issue 2, February 2012, ISSN 2151-9617,pp.108-113.

[3] M.P.Sivaram Kumar, S.Rajasekaran, "Path Planning Algorithms for suppressing Forest Fires using Genetic Algorithm", CiiT –International Journal of Artificial Intelligent Systems and machine Learning", vol 4, No 4, April 2012, pp.235-240

[4] M.P.Sivaram Kumar, S.Rajasekaran, "A Neural Network based Path Planning Algorithms for Extinguishing Forest Fires", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012, pp.563-568

[5] Richard Fox, Antonio Garcia.Jr and Michael nelson, "A Two dimensional Path Planning Algorithm",Informal publication

7. About the Authors

M.P.Sivaram Kumar Graduated with a Bachelor degree in Computer science and Engineering in 1998 from University of Madras. He then obtained his Post graduate degree in Computer Science and Engineering from Madurai Kamaraj University in 2002.Currently working as Assistant Professor in Computer science and Engineering Department of DMI college of Engineering, pursues part time PhD in B S Abdur Rahman University, Chennai-600048. He has Published 4 papers in International Journals. His areas of Research includes Wireless sensor and Actor Networks, Robotics, Data structures, Algorithms and Theory of Computation. **Mail id:** sivaram_softengineer@yahoo.co.in

Dr.S.Rajasekaran received Predoctoral degree in Mathematics from Madurai Kamaraj University in 1981.He obtained hid Doctororate degree from Indian Institute of Technology Kanpur in 1986.He has authored more than 20 papers in peer reviewed international journals. Currently working as Professor of Mathematics in BS Abdur Rahman University, Chennai-600048. His areas of Research include Modeling, Artificial Intelligence, Algorithms and Theory of Computation. **Mail.id:** rajauv@yahoo.co.in