ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

Graph Traversal Technique for Grid Based Robot Navigation

Lalit Gehlod[1],Vaibhav Jain[2], Mala Dutta[3],Devesh Kumar Lal[4]

Asst.Professor, Computer Engg. Department, IET,DAVV, Indore, India [1]

Asst.Professor, Computer Engg. Department, IET,DAVV, Indore, India [2]

Asst.Professor, Computer Engg. Department, IET,DAVV, Indore, India[3]

Student, Computer Engg. Department, IET, DAVV,Indore, India[4]

Abstract: Navigation is a major challenge in the field of artificial intelligence. There are several techniques for navigation of robot over a grid and these papers represent an approach for navigation on a grid by using breadth first algorithm. Here the entire grid of m*n is converted into a tree and with the help of these tree we apply BFS for traversing .these paper we develop an approach for searching an object and also able to avoid an obstacle which was placed in a junction

Keywords: Grid solver robot, shortest path algorithm, Line follower robot, Breadth First Search (BFS), Material Handling, path planning.

I. INTRODUCTION

There are various tasks which is performed over a grid such as material handling, obstacle avoidance and object detection etc. requires a strong navigation technique which can be fulfill by using these algorithm. The major problem we faces while traversing on a grid is to maintain its current location after encounter a junction and also the path planning is required when an obstacle is detected. These problems are easily tractable with breadth first search. Here we convert the entire grid into a tree type structure. Where we apply BFS for searching the next node. The algorithm is briefly discussed in later section [1][2].

The grid based robot navigation system is one of the most dynamic areas of material handling today. Transportation of raw materials and finished products is typical in an industry [3]. Controlled transportation and product identification, as well as safe movement throughout the process, are the key to such type of installations [8].

A grid is represent as the [m * n] matrix where m is the number of rows and n is the number of columns. The rows and columns are the black line which is drawn over a white surface or white line which is drawn on black surface. The robot which follows a single line is known as line follower robot the line is either black or white. Here we use IR sensors for sensing the line. IR sensors are able to distinguish between white and black lines. IR sensors consist of an IR emitter and IR receiver pair [9]. The high precision IR receiver always detects a IR signal. The white and black colors has different wavelength which can be compares by IR sensors [10][11].

To follow a Black line we require minimum two sensors. The sensors are placed one after another. To follow a line we requires two different condition i.e. when the left most sensors are in black line we have to take slow right turn and when the right sensors are in black line we have to take slow left turn. This is the minimum condition to make the robot on line. Similarly to traverse over the grid it follows the same condition but when the junction has been detected (when all the sensors are attain into the line it is detected as a junction) at such circumstances we have to take the verdict whether to take left or right turn[12].

To navigate over a grid we have to follow Cartesian coordinate system for finding the current location on the grid. The robot set its initial location as (0,0) and maps all the quadrant according to it. The left node as (-1,0) the right node as (1,0)and the node below the origin is taken as(0,-1) and above the origin as (0,1) respectively. The robot has also to maintain its direction while moving forward, left or right it has to update the direction according to the turn [8].

The major problem that we faces in navigation is what is decision taken by the bot to find the next coordinate which can be easily achievable when a tree is constructed

. Another problem we are face that is at the time of obstacle detection which can also be solve with these approach. The obstacle and object is placed on the junction.

The obstacle is a cubical block with half portion is colored with white and another half with black and we consider the purely black or white as an object. The decision taken by the bot after obstacle encounter is discussed in section III (c) and fig 4.

II. Environments used For Robot Navigation

A. Type of Grid

We can use the grid of any dimension of [m * n] as shown in fig 1.The grid may also consist of combination of multiple grid. For such situation we have to place two extra sensors in below the center of both the wheel. These sensors are capable of to keep the robot into the grid with a condition i.e. when all the sensors in (white or black) we have to take (right or left) turn until the line is detected [5].

B. *Grid Mapping*

The grid is map according to Cartesian coordinate system. Robot sets the initial node as an origin and maps the entire grid according to origin and its direction which shows in fig 1.

C. *Figures*

1. Coordinate System of Grid

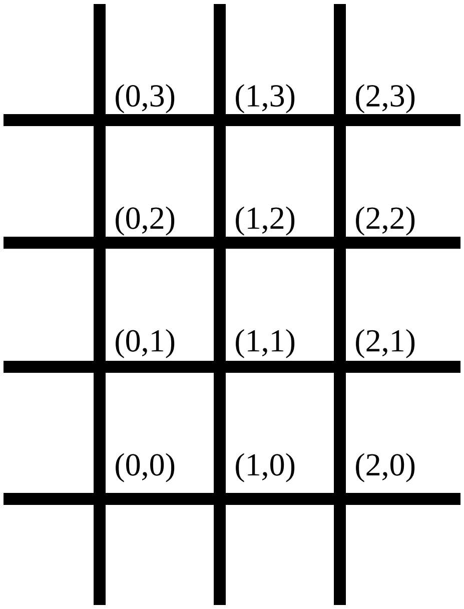


Fig 1 Grid [3*4](shows about a black line which is drawn in a white surface or vice-versa).The junction is labelled as according to its coordinate system.

2. Obstacle

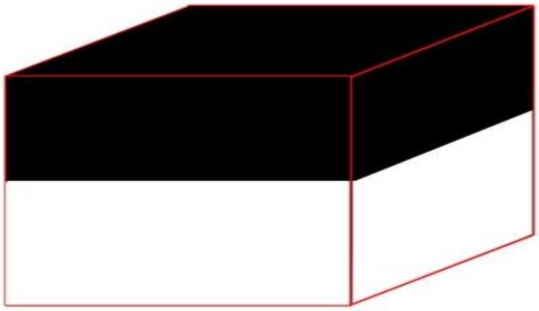


Fig 2 Obstacle (Two IR sensors are placed in front of the robot which is which is used to detect the two different colours are treated as an obstacle).

3. Object

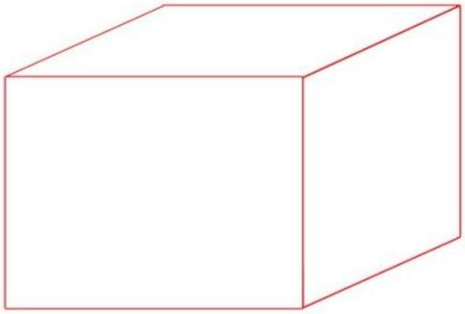


Fig 3 Object (The pure white and black block are consider as an object i.e. if both the sensors give the same value it is an object).

III. ALGORITHM USED

A. *Breadth First Search*

In breadth first search require five different parameter such as unexplored vertex, visited vertex, unexplored edge, discovery edge, cross edge .firstly the node is visited which is closer to the root node or we can say the node which is at the same levels are visited and then the second level and goes to higher level accordingly[4].

BFS(G,s)

```
For each vertex u ϵ V[G] — {s}
Do color[u]← WHITE
D[u]← ∞
Π[u]← NIL
Color[s] ← GRAY
d[s] ← 0
Π[s] ← NIL
Q ← Ø
ENQUEUE(Q,s)
While Q ≠ Ø
Do u ← DEQUEUE(Q)
For each v ϵAdj[u]
Do if color [v] = WHITE
Then color[v] ← GRAY
d[v] ←d[v] +1
Π[v] ← u
ENQUEUE(Q,u)
Color[u] ← BLACK
```

B. *Transformation of Grid*

While using BFS we have to construct a tree by the use of the grid. By using these transformation reduces the complexity while in navigation. Transformation to tree requires one stack such as traverse stack it keep the record of the visited node. This stack is useful to detect next node. If the coordinates of next node is matched with the traverse stack then such nodes are avoided. While constructing the node on a tree we have to move on higher level on a grid. And the junction of two adjacent node of parent node is treated as a connected node. As in Fig 1 Grid the transformation is done accordingly [6][7].

The root node i.e. (0,0) consists two connected node such as (0,1) and (1,0) which is at the same level are placed as a child node of the root node and another node of (1,1) is adjacent node of (1,0) and (0,1) are also used as a connected node of (0,0).while connecting as a child node of (0,0) the connected node is checked into a traverse stack if these nodes are presents in the stack then such nodes are avoided and unvisited node are marked as a child node. So the corresponding tree of the grid is shown below is free from obstacle.

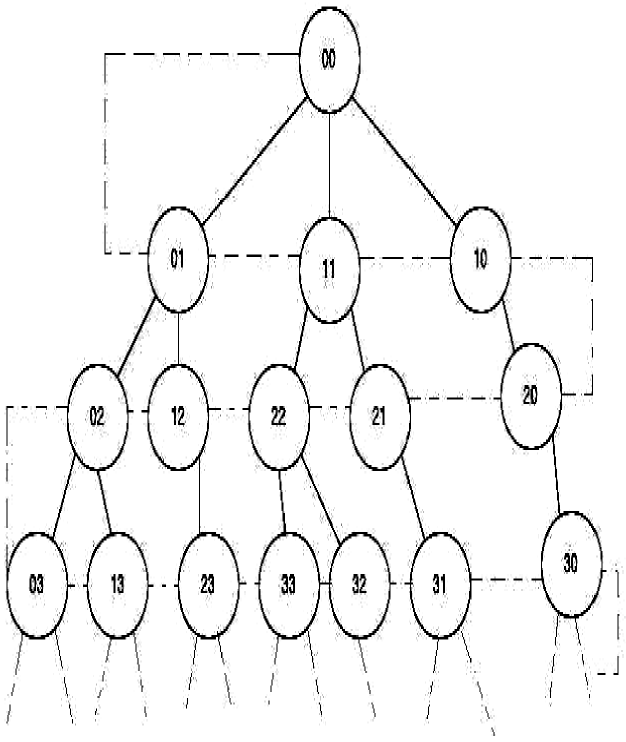


Fig 3. Grid transformation into corresponding tree.

In fig 3 the bot start from the root node i.e. 00(due to reducing the complexity (0,0)(3,4)(6,3) etc. are written as 00,34,63) it will detect two connected node i.e. 01 ,10 and 11 as explained above. So by these properties the entire tree is formed.

C. *Grid Transformation in presence of obstacle*

The transformation is little bit clumsy when an obstacle is arrived in a junction. Such problem can also be solve by tree traversal algorithm but here we require another stack such as obstacle stack it keeps the record of obstacle. When the obstacle is detected by the bot it has to move into the next level and begin with the initial node of the next level and move up to the node it surpasses the obstacle node and again move to the previous node. For example if in a grid consists an obstacle at (1,1) and (2,2). So the bot has to follow the same algorithm as without obstacle but if obstacle is encounter than it has to place the value of coordinate into the obstacle stack and move to next level of starting node i.e. (0,2) and move until the obstacle node is surpasses i.e. (3,1) than it has to move to previous level and accordingly as shown in below graph.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 7, July 2013

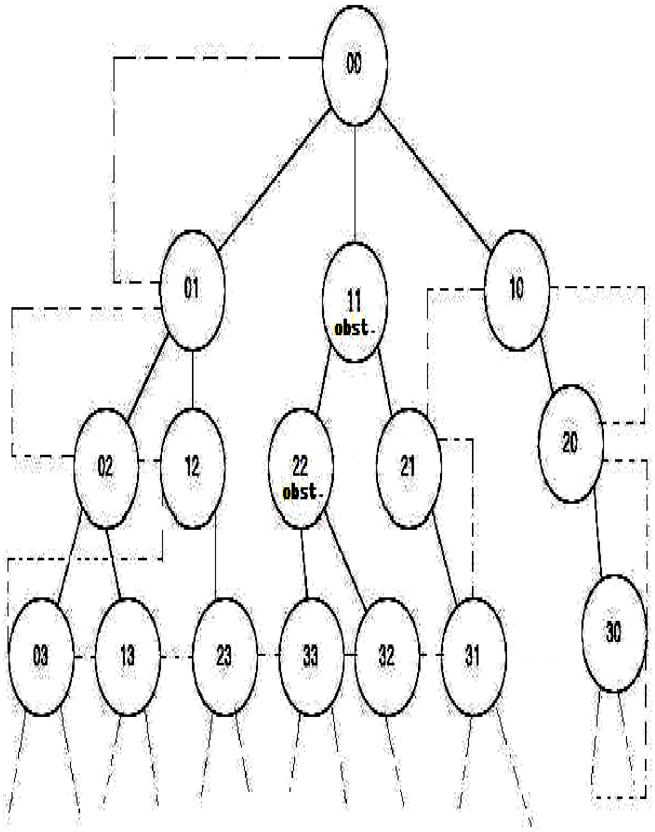


Fig 4.Bot Navigation when grid consist of an obstacle.

In fig 4 grid consist of an obstacle at junction (1,1) and (2,2) whereas the dashed line shows the path i.e. 00->01->02->12->03->13->23->33->32->31->21->10->20->30 . The bot navigate until the desired object is detected and give the shortest path from the root node to the object.

D. Calculating direction

To maintain the location of bot on any arbitrary points on a grid we have update the direction and value (x,y)which was (0,0) initially and maintain according to its turn.

Direction	Turn	Set Direction	Set Co-ord.
North	Forward	North	Y++
North	Right	East	X++
North	Left	West	X- -
East	Forward	East	X++
East	Right	South	Y- -
East	Left	North	Y++
West	Forward	West	X- -
West	Right	North	Y++
West	Left	South	Y- -
South	Forward	South	Y- -
South	Right	West	X- -
South	Left	East	X++

E. sensor placing

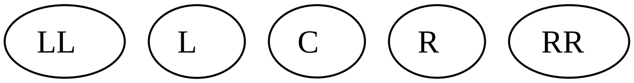
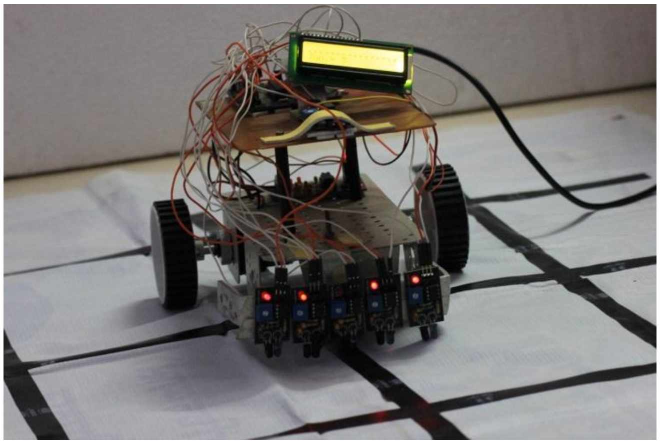


Fig 4 IR Sensor Placing (LL-Left Most sensor, L- Left sensor, C-Center sensors, R-Right sensors, RR-Right Most sensor)

IV. RESULT

For implementing this algorithm we use five sensors in the bottom of the bot. The most corner sensors keep the record of the junction while three sensors in the middle keep the bot in line. We use one sensor in front for detecting an object.



We used arduino atmega8 circuit for the navigation. Here we kept block on arbitrary point (2,1) the bot follows the path according to these approach is 00->01->11->10-> 20->21.

V. CONCLUSION

This paper will give an easier approach to navigate over a grid. The entire grid is transformed into a tree where for searching of obstacle and object we apply Breadth First Search which is an easier approach for searching. It also changes the route when the obstacle is detected. The bot can also able to locate its coordination with the help of its direction. So the overall technique of navigation is quite simpler and errorless. This paper represents path planning and navigation over a grid with a single bot. In future work, we can introduce multiple bot over a grid which will suppose to communicate with each other about its self-position and the obstacle that has been placed over a grid. So, it is easier for the other bot to decide the shortest route.

REFERENCES

[1] Sreekanth Reddy Kallem,“Artificial Intelligence Algorithms”IOSR Journal of Computer Engineering (IOSRJCE).

[2] Sebastian Thrun,”Robotic Mapping:A Survey” Feb 2002, CMU-CS-02-111, School of Computer Science, Carnegie Mellon University Pittsburgh.

[3] Robert J.Szczerba Danny Z.Chen John J. Uhran,”A Grid-Based Approach for Finding Conditional Shortest Paths inan Unknown Environment,Nov 1994.

[4] Algorithm Concept by NarasimhaKarumanchiData Structures and Algorithms Made Easy second edition.

[5] Joachim Hertzberg and Frank Kirchner. Landmark-based autonomous navigation in sewerage pipes. In Proceedings of the First Euromicro Workshop on Advanced Mobile Robots (EUROMICRO).

[6] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. AI Magazine.

[7] Michael J.Milford Janet Wiles Gordon F.Wyeth, “Solving Navigational Uncertainty Using Grid Cells On Robots” november 2010,school of engineering systems, Queensland

[8] CarstenBuschmann Florian Muller and Stefan Fischer,”Grid-BasedNavigation for Autonomous,Mobile Robots”,Institute of Operating Systems And Networks,Technical University of BraunschweigBraunschweig,Germany.

[9] http://www.atmel.com/images/doc2466.pdf

[10] Ahmedullah Aziz, Md. ShafayatHossain and Mohammad WahidurRahman “Programming and construction of Ahmedullah-A Fast Grid solver robot”International Journal of Information Technology, Control and Automation (IJITCA) Vol.3, No.1, January 2013.

[11] IR Sensors reference from Robosoft System “IR Sensor Based Obstacle Detection Sensor Module”.

[12] Richard T.Annoy,“Design a Line Maze Solving Robot”, april2009http://www.pololu.com/file/0J195/line-maze-algorithm.pdf.