Cave Exploration of Mobile Robots using Soft Computing Algorithms

S.Vignesh
Thiagarajar College of Engineering
madurai-15

T.K.Karthik
Thiagarajar college of Engineering
Madurai-15

P.Sonaiya Karthick
Thiagarajar College of Engineering
madurai-15

ABSTRACT

Cave Exploration is main task for most speleologist. There are cave humans can't enter or even survive. Technology can help them to explore cave with modern tiny robots. Mobile robot starts its job in an unknown cave environment and faces lot of obstacles and avoids it using fuzzy logic. A mobile robot records each and every position of traveling environment using Monte Carlo localization. Mobile robots also using camera's to capture each and every grid location and store it in a robot's internal memory. For robot's path planning we introduced a new methodology so that robot can map the whole cave environment in a efficient manner. This type of mobile robots will useful in an environment where people not able to travel and also this mobile robot will helpful for those who are doing researchers about cave.

Keywords: Monte Carlo localization [1][2], path planning[6][7], Micro controller, Fuzzy-logic[9][10], IR sensors.

1. INTRODUCTION

Robotics playing a big role in all fields such as education, manufacturing, defense, etc. There are lot of mobile robots involved in industries which is acting as a surveillance for that industry . Also mobile robots acting as guide for blind people to reach a particular location without any disturbance and also mobile robot able to handle all type of obstacles and should track them. Cave mapping is the toughest task by human. So mobile robots are the suitable agents to track the cave. Cave will have lot of paths and obstacles and other disturbances. Nowadays many mobile robots or some other devices are there in the world to map the cave but efficiency is the important and needed factor .

Localization algorithms are very useful for mobile robots to measure the position's and also camera mapping also one of the better factor to find the cave views and also using that human can easily find the path when they are missing inside a cave. Here we introduced a efficient methodology for path planning of mobile robots. Here we tested initial processes i.e. goal tracking of mobile robots using FIREBIRD mobile robot which uses ATMEGA2560 micro device.

2. MONTE CARLO LOCALIZATION

Position Tracking done using Monte Carlo localization [1][2] algorithm. Monte Carlo Localization is a different approach to the problem of robot localization - estimating a robot's location in a

Fig 1: specifies the grid area of an indoor environment

unknown environment. In this paper , its solve the localization problem, where the robot does not know its starting position but needs to figure out where it is. To implement this algorithm in this paper divided our search area into as square grid. Each grid is given by a specific value which has coordinators .The starting position of mobile robot will not be defined. It will be find by mobile robot(FIREBIRD kit) based on sensor values. The following snapshot(white space) shows our paper environment(grid) which is specified above.

Monte Carlo Localization Algorithm:

1.   Inputs: goal value, Sensor Reading, grid values

2.   Start:

3.   for x=0 to until reach goal

4.   for y=0 to until reach goal

5.   Newgridvalue[x][y] = updategridvalue(x,y) *//Compute new location*

6.   If (Newgridvalue[x][y] == goal value)

//Checking *grid value with the goal value. Goal value is actual position from where robot starts*

7.   LCD out "cave is explored"

8.   Else

9.   Search the goal;

10.   STOP.

3. PATH PLANNING

Path planning is needed factor for robot goal tracking in order to reduce the time to reach the goal. There are many more shortest path algorithms. Here we took new path searching methodology [6][7][12]. The following algorithm used to give intelligence to mobile robot (FIREBIRD) to determining which grid to choose next. We introduced new algorithm for path planning.

ALGORITHM

step 1

  initialize store-grid array to zero

  initialize x,y position's to zero

step 2

  Update the score value of new grid position with getGridXYValues (direction) function into store-grid array

Step 3

  get the left sensor grid value from store-grid array and store into variable x

  get the straight sensor grid value from store-grid array and store into variable y

  get the right sensor grid value from store-grid array and store into variable z

if x,y,z values are same, then

  if x,y,z are infinity then

    turn back

    score = score +1

    move;

    set direction ← back;

  else

    turn left;

    move;

    set direction ← left;

  end if

else

  if x,y are same then

    turn left

    move;

    set direction ← left;

  else

    if x lesser than y

      turn left;

      move;

      set direction ← left;

    else

      move;

      set direction ← straight;

    end if

  end if

if y,z are same

  turn right;

  move;

  set direction ← right;

else

  if x lesser than z

    turn right;

    move;

    set direction ← right;

  else

    move;

    set direction ← straight;

  end if

end if

if x,z are same then

  turn left

  move;

  set direction ← left;

else

  if x lesser than z

    turn left;

    move;

    set direction ← left;

  else

    turn right;

    move;

    set direction ← right;

  end if

end if

end if

getGridXYValues (direction) function

  if direction = right then

    $x \leftarrow x + 1$

  if direction = straight then

    $y \leftarrow y + 1$

  if direction = left then

    $x \leftarrow x -1$

  if direction = back then

    $y \leftarrow y -1$

step 4

Repeat the step 2,3 till robot reaches starting gird again with visiting all the paths from starting grid.

2	2	2	2	■	2
■	2	2	2	2	2
1	1	■	1	2	1
1	1	2	1	1	■
1	■	2	■	1	1
1	1	2	1	2 start	1

Fig 2: Example Grid, after implementing Algorithm. Number inside grid is number of times grid is visited. Black grid represents obstacle.

4. *RELATIVE GRID NUMBERING*

Since, cave is an unknown map. There is no absolute grid position. So, in this work, we propose the relative addressing[3][4] of grids. Starting point of journey is marked as [0,0] ie. X axis 0, y axis 0. if robot is moved in left direction then x value is decrement by one value from its previous value. If robot is moved in right direction then x value is incremented by one value from its previous value. All grid position and its status(score values or obstacle) is stored in separate array. This array is helpful when reconstructing map of the cave.

Example Relative Mapping

$[0,0] \rightarrow 2 [-1,0] \rightarrow 1$ $[1,0] \rightarrow 1$

$[1,1] \rightarrow 2 [1,2] \rightarrow 1 [1,-1] \rightarrow 1$

5. CAMERA SURVEILLANCE

A pin point camera is fitted at the top of robot for capturing scene in its journey. Whenever a robot reaches a next grid from its current grid. A signal is supplied to camera. Camera will capture snaps of current grid. Camera takes 3 snaps per grid. It covers both sides of robot and one from path its following.
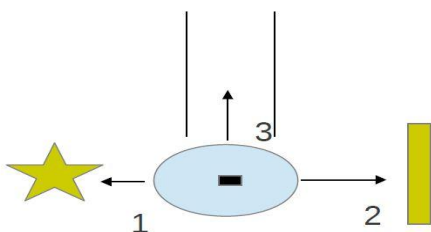

Fig *3*: Taking shoots on 3 sides of its position

After reaching goal, these pictures are sent to android held smart phones. We are proposing a new application, which

collects all these pictures and correlate with Grid positions. If suppose, a person is missing inside a cave, then person can identify one's grid position by taking image of his/her surrounding. This application will do image processing with captured image and find the original grid location he/she missing.

6. OBSTACLE AVOIDANCE

Mobile robot (FIREBIRD kit) faces lot of static obstacles during its tracking. To avoid these obstacles and track the goal mobile robot, uses Fuzzy logic[9]. According to this input's of sensors, mobile robot take decisions such as turn right or turn left or move straight with following optimal path. The following table gives detailed description of fuzzy logic which was used in this paper.

Table 1: Table shows actions performed on robot based on the Right, Left, Straight Sensor values

Straight Sharp Sensor	Left IR Sensor's	Right IR Sensor's	Result of mobile robot
0	0	0	**RSL**
0	0	1	**SL**
0	1	0	**RL**
0	1	1	**L**
1	0	0	**RS**
1	0	1	**S**
1	1	0	**R**
1	1	1	**NO PATH**

. L - Path in Left, R - Path in Right, S - Path in Straight

7. EXPERIMENTAL RESULTS

The following snapshots explains our paper which was taken in an indoor environment.

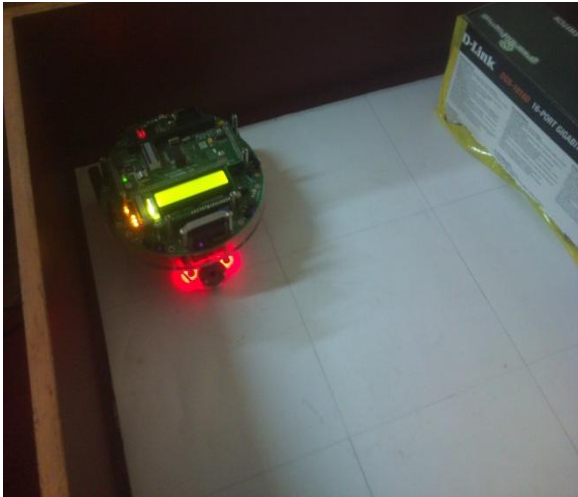**Fig 4: specifies the goal of our paper**

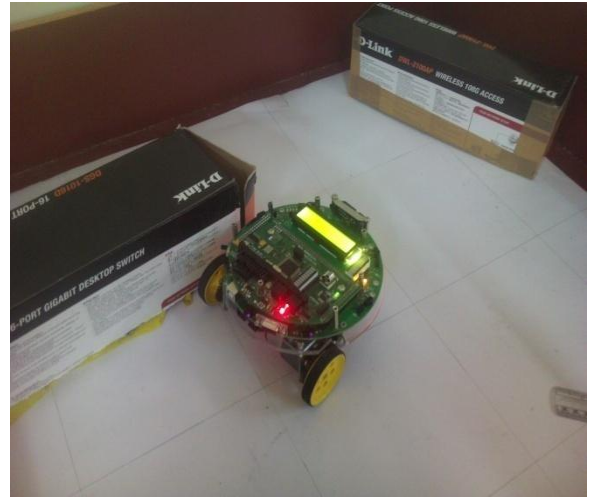**Fig 5: mobile robot's (FIREBIRD kit) initial position in grid environment.**

Fig 6: Mobile robot(FIREBIRD kit) follows path planning according to Monte Carlo Localization.

Fig 7: mobile robot(FIREBIRD kit) found obstacle.

Fig 8: mobile robot(FIREBIRD kit) uses fuzzy logic for obstacle avoidance.

Fig 9: mobile robot(FIREBIRD kit) reaches the goal.

8.CONCLUSION AND FUTURE WORK

This paper explained, how a mobile robot explore the cave using Soft Computing algorithms and also we provided a partial solution in indoor environment using FIREBIRD mobile robot kit. This paper helps for increasing mobile robot's work efficiency, our future work will be expanding procedure for complex caves that contains dynamic obstacles like animals.

9. REFERENCES

[1] Zhiwei Liang, Xudong Ma and Xianzhong Dai,"Monte Carlo Localization:Extended Monte Carlo Algorithm to Collaborate Distributed Sensors For Mobile Robot Localization", Proceeding of the 2007 IEEE International Conference on Robotics and Biomimetics December 2007.

[2] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun, "Monte Carlo Localization: Efficient Position Estimation of Mobile Robots", Proceeding of the 1999 National Conference on Artificial Intelligence (AAAI).

[3] A Francy Golda, S.Aridha, and D.Elakkiya,"Algorithmic Agent for Effective Mobile Robot Navigation in an Unknown Environment." Intelligent Agent & Multi-Agent Systems, 2009. IAMA 2009.

[4] R. Clark, A. El-Osery, K. Wedeward, and S. Bruder "A Navigation and Obstacle Avoidance Algorithm for Mobile Robots Operating in Unknown, Maze-Type Environments" Proc. International Test and Evaluation Association Workshop on Modeling and Simulation, Las Cruces, NM, December 2004.

[5] Wang, L., "Computational intelligence in autonomous mobile robotics- A review", Proceedings of the 2002 International Symposium on Micromechatronics and Human Science, Nagoya, Japan, Oct. 20-23,2002, pp 227-235

[6] Anytime Dynamic A*: An Anytime, Re planning Algorithm Maxim Likhachev, Dave Ferguson, Geoff Gordon, Anthony Stentz, and Sebastian Thrun School of Computer Science Computer Science Department Carnegie Mellon University Stanford University Pittsburgh, PA, USA Stanford, CA, USA

[7] C. W. Warren, "Fast path planning using modified a* method," in Proc.IEEE Int. Conf. Robotics and Automation, Atlanta, GA, May 1993,pp.662-667.

[8] Jurgen Wolf,Wolfram Burgard,Member,IEEE and Hans Burkhardt,Member,IEEE,"Robust Vision-Based Localization by Combining an Image-Retrieval System With Monte Carlo Localization" IEEE transactions on Robotics,vol.21,No.2,APRIL 2005.

[9] "A Practical Fuzzy Logic Controller for the Path Tracking of Wheeled Mobile Robots" by T.H.Lee,H.K.Lam,F.H.F.Leung,and P.K.S.Tam.

[10] M D Hurley ,"Obstacle Avoidance Using Fuzzy Logic Approach".

[11] Han Hongling, Yang Fenglei, "Path Planning of an Indoor Mobile Robot Navigated by Infrared"

[12] MightyMouse: An Autonomous Maze Solving Algorithm, IEEE transactions, Kelly Ridge, Sanjeev Giri, Peter Shaw, Jason Flynn.