Aerial Robotics Kharagpur Documentation Template

Ashwary Anand, Other Author/ Contributor Names in Order of their contribution to the project *

Abstract— **Should be half column long. Should be clear enough to explain your whole documentation. Similar to a TL;DR**

Write in brief what you have done and give a small outline about the fields of application. Eg. for object detection , write about the fields where this can be scaled , for solidworks model write about the place where it will be used in ARK or in general it's uses.

I. INTRODUCTION

Describe the problem statement and your approach using which you have approached the problem. Describe all the things and methods you have tried which might have failed in brief.

II. PROBLEM STATEMENT

This should cover one full column of page.

You and your fellow friends are stuck in a dark maze and you have no idea about your surroundings, since its night and you have to reach home real quick, you have to navigate your way out of this maze. To do so, at any moment you can light a match, and that in turn will give you an instant circular snapshot of your surroundings, with your position as the center of the circle and a fixed radius (in pixels). You have to predict the map of the maze, while traversing through it. You can light as many matches as you want. You will be judged on the basis of accuracy of the rendered map and the number of matches lit by you. You have to design appropriate publisher and subscriber nodes and ROS topics for the aforementioned interaction. After figuring out the maze, you need to tell your friends the path they should follow for traversing the maze. You can use a variety of path-planning Algorithms like Depth and Breadth-First searches, Djikstra's and A-star Algorithm, RRT etc for this purpose. Note that the sample maze has 2 exits and it is given that only half of your friends can pass through one exit. The green dot in the sample image is the start point and the red dots signify the two exits. Clone this repository to get the subscriber set up. The Topics for the interaction will be as follows : MATCH LIGHT : You will publish your coordinates to this topic. SNAP RETURN : You will subscribe to this topic in order to get the circular snapshot image. PUBLISH PATH: After creating the entire environment of the maze, you will have to publish it in this topic.

III. RELATED WORK

Some existing work is available in solving mazes but none of them is close to this.

IV. INITIAL ATTEMPTS

When thinking of this problem RRT or RRT* strikes but we have to give the whole map so they won't work. Now a* could be a preferred algorithm for this problem as it also has to deal with the current neighbourhood, but as we are not allowed to use the coordinates it won't work.

Also we have to output the map that seems to be the toughest problem as we need to store the map and also there are two exits. In the present scenario a BFS or any other brute force method along with compiling a lot of images together may seem the best approach. it also strikes me that RRT may have been the perfect algorithm over here but lets say we wanted to move a few units in a direction the wall would't have stopped us.

probably the solution of this problem could be derived from BFS plus amassing all the parts of map that we called for. In the present scope this problem feels undoable in my current level of understanding. although in the past had implemented various algorithms for maze solving but to figure out the map feels a hard task.

One of the approaches may be to declare a structure that stores the past pixel/node and our code as it moves to a new pixel calls the match light function and sees what regions/pixels are open to explore. then this process continues till we had found the two goals and now then from the goal pixel we could get its previous pixel and then could publish/store the back path from goal to start.

We could map each pixel into a binary matrix that will represent what areas are accessible and what are not. and as we receive data we could complete the matrix and at the last return it.

Write all the equations and results with pictures if applicable.

V. FINAL APPROACH

This should cover both columns (full page) excluding images

Write about the final algorithm used or developed which was able to finally solve the problem statement. Write all the assumptions and equations properly with images and result tables / comparison.

If report is about implementation, write all the steps of implementing the report with step-wise syntax and details about the errors faced and their solution.

VI. RESULTS AND OBSERVATION

Compare your results with all the available algorithms which you may have used to tackle the PS. If possible, present your and their results in tabular / graphical format.

*Write anyone who might have helped you accomplish this eg any senior or someone

Explain the trend of results in details. Mention the drawbacks (if any) of your algo compared to other algo and the reason of picking up the approach over the other if you have implemented any algo over the other.

VII. Future Work

Write about the problems in your algorithm / approach and limitations in testing (if any due to hardware or otherwise) and how to tackle them and any future work which can be done to improve the results further.

CONCLUSION

Write overall about what the problem was, how you solved it, difficulties faced and the net output with it's usefulness to ARK or in general.

References

[1] Khatib, O., ”Real-time Obstacle Avoidance for Manipulators and Mobile Robots,” International Journal of Robotics Research, Vol. 5, No. 1, pp 90-98, 1986.

[2] Write all the papers and links you have referenced to complete your project.One example for format is given above, Format followed should be ::

[3] Author Names, ”Paper Name”, Conference / Journal where the paper was published , Year of Publication