Task 1: Dynamic Path planning

Debjoy Saha

Abstract—
The article proposes a basic approach to dynamic path-planning using RRT as a tool. Dynamic path planning is fundamental to a bot navigating in a transient environment with moving obstacles.

I. INTRODUCTION

The given task is on dynamic pathplanning with obstacles in motion. Basic RRT implementation is used to determine feasible maps connecting the source and the destination real-time. The input is through a pre-recorded video which captures the positions of the dynamic and static obstacles. The code is written on C++ using library OpenCV for video analysis.

II. PROBLEM STATEMENT

The Video comprising of obstacles was provided to us (available in the directory as dynamicobstacles.mp4). The task was to perform a frame by frame analysis of the map and plot a path to the destination in as short time as possible. Traversable path was coloured in white, while static and dynamic obstacles as black boxes and blue circles respectively. Starting and ending points were red and green circles. Every configuration of obstacles continued for 10 frames and a feasible path was required to be made within that time in varying colours.

III. RELATED WORK

Various different forms of RRT were used by different authors to traverse through a map containing dynamic obstacles, each implementing different modifications and optimizations. RRT* was used as an alternive to produce smoother,more feasible paths.

IV. INITIAL ATTEMPTS

Initially A* algorithm was thought of to traverse the path piecewise i.e only certain pixel movement per frame. That is for each frame it will create a square matrix of certain side length signifying the validity of each point (obstacle/free). That along with the heuristic matrix will be passed to a function that will map the initial point to the local destination(that having the lowest heuristic value). Initial point will be updated and the process will be repeated until destination is reached.

This approach was discontinued with due to high time complexity.

V. FINAL APPROACH

The final solution comprised of a basic solution to the path finding problem by implementing RRT. Image was converted to binary form and image segmentation was done to identify start and end points using a point find() that returns Point variable by applying canny and detecting center of circular contours (using openCV function contours and minEnclosingcircle())neglecting stray contours.

Done that, the grayscale image was passed to function findpath() along with an empty frame, initial point details and a stack to store generated points, initially containing only the starting point only. A random point(say p1) was generated in the whole frame and a point already included in the list of visited points having lowest distance from the generated point was determined(say p2). Then a step having a specific stepsize was taken from p2 towards p1 resulting in the occurence of a new point p3 ,the other end point of the step p3(=p1+stepsize*(p2-p1)/—p2-p1—).

If no point falls on obstacle then p3 was added to the list of path points else the code moves on to the next iteration. This algorithm is followed until the path reaches the destination. If in any later frame any point on the map falls on an obstacle a new path was generated, the previous list was cleared.

Due to certain errors, the path at certian times passes through small parts of obstacles and due to time constraint may not for all frames reach the destination. Also errors during image segmentation leads to random clearing of lists and reinitiation of path.

VI. RESULTS AND OBSERVATION

A satisfactory path was generated for most frames. However, above bugs remained and time taken to generate the paths was quite long. Another drawback was final path was not shown in each case. Otherwise RRT algorithm was working.

VII. FUTURE WORK

Optimizations need to be done to generate the path in a smaller interval. Also, the bugs need to be corrected and the final appearing path needs to be identified on a dynamic basis.

CONCLUSION

The solution presents a basic approach to dynamic obstacles problem and a lot of improvements would be required for it to be pracically viable.

REFERENCES

[1] https://www.youtube.com/watch?v=qFc6g9KrnOk