Aerial Robotics Kharagpur Documentation Template

Abhinav japesh, 18EE30001

***Abstract*— it is basically a path planing on a video having 2 types of obstacles....static and dynamic .in video black is the static obstacle and blue is the dynamic one.it can be used for automatic moving bots as while moving it can encounter some moving objects as well so has to update its path accordingly.**

I. INTRODUCTION

II. PROBLEM STATEMENT

Task 1¿¿ task was to encounter dynamic obstacles and static obstacles and find path accordingly in a video given startng pont is green and end point is red....(video=dynamic-obstacls.mp4) encountering a static obstacle ,we have to just check if newly generated pixel is valid or not.in this case white is accesible region and black is obstacle so,if pixel has (B,G,R) values corresponding to black then we have to find a new point....but in moving obstacle we not only have to check this but also have to encounter that moving object would have come in our way after we have visited that path.....so we have to now generate new path as that path while traversing will go from that object .....so we have keep a track of our visited path as well,if our erly path is still valid in new frame,ifnot ,go for new path.

III. RELATED WORK

my first approch was trough A* to find the shortest path b/w initial and final path.but we have to use rrt instead as it would be fast (a-star.cpp) then implemented rrt in 3 attempts.in first rrt aproach on video random tree is generated but it has problem while retracing the main path(tree.cpp) so in 2nd attempt of rrt on static image it run succesfully with a proper path(rrt-static.cpp)...but its implementation on the video seems to be bit complicated ...it is showing path but result is not as expected..(rrt-vid.cpp)

IV. INITIAL ATTEMPTS

1.in a video we are inputing a frame then first we are we are re assigning the RGB vales i.e if a pixel is light red or some shade of red then converting to complete red.

2.finding the center of initial and final spot.(applying canny::fnding contours::finding center;)

3. we have defined a struct (cordi) storing x and y cordinate, and another struct(node) having its position(cordi) and a pointer(*node) which points to its parent point, a vector(vec) storing these nodes..

4. after finding initial point pushing it into the vec(//parent of initial point is initial point itself)

5. now starting video by taking a frame,finding a random point on it(rand() (mod) img.rows to limit random point within img.rows and cols)

6.finding a node from the vec whose distance is closest to the the random point generated. by iterating through a vector using a iterator.(as we pushing nodes in this vec so now traversing this vec and checking distance with random point generated ....initialised min=999999,if distance is less than min then min is updated as it and recquired point becomes that which iterator is pointing....it will continue till vec.end();)

7.now increasing d distance in diection of random point(d is a scalar +ve value so we have to take care of direction of increase with basic approach newx=x+dcos(theta);newy+dsin(theta);)

8.now checking if this new point is within the range of our window and is not lying on obstacle.. if condition verified then drawing line b/w new and previous point....and pushing this nw oint into the vec..if not then random point re generated (//also assigning the parent of this new point points to the last point) (for moving obstacle not in this code in next and final code)

9.this infinite loop will continue til the distance b/w the new point generated is within a particular range of final point...after that it will join the final point and last new point(line func)

10.tree is generated.. (//for retracing i have written code bul had some problem .....this code is still there in code but in comments) 11.(//assigning parent of final point is last ne point) 12 (//defining 2 nodes up and down up stores the parent and down stores current point. 13.(//initially down is final point and up is final.parent....now updating (down as up) and up as up.parent) going in reverse direction.generating the final path)

failed in retracing back as it was just updating the corinates of final and last newpoint so same point was inputing...and after using pointers it gave some big random no. due to which core dumped..and also had written everything in a single function so was told to use severall function so gave another try with same idea but different approach of retracing path ...

V. FINAL APPROACH

1¿ input video took the first frame¿¿imgs,img=imgsdid colour reassigning in this first frame by binary func. which convert every colour into black ,white, blue, red and green.

2¿defining a struct point which will store x and y cordinate...a struct edge which will store 2 point parent and

children and a struct line storing coefficients of x,y,1 for eqn. of line(ax+by+c=0) ....a vector nodes storing point and vector edges storing edge..

3¿found center of the initial spot of particular color as it will cntain lot of pixels of same colour so finding mean of those cordinates and pushing this initial point into our vector nodes.

4¿now starting while loop for video and inputing 2 frams imgs and imgf subtracting it(imgd=imgs-imgf) now if imgd is all black then img=already defined as 1st frame if imgd show some changes then img is updated as new frame img=imgs.(idea is that if 2 corresponding is same then then rrt wil work on that current frame till frame show some difference..... as frame gap maybe less so for rrt to complete path in a particular frame)

5¿generation of random point–rand() mod img.cols/rows

6¿now we applying condition for case if moving obstacle comes in our visited path.... before generation of nearest point from vector of nodes we are traversing in vector nodes and checking that the pixels of old nodes are not blue ...(nodes have 2 iterators itr and ptr)..if so then we are erasing the nodes in vector correspondin edge from edges from that to end..(if (*ptr) shows blue then nodes.erase(itr) where itr goes from (prt+1) to vec.end(); and also redrawing those blue line with white line//erasing drawn lines) vector(nodes),vector(edges) updated....

7¿finding nearest point in this new vector of nodes...(same as previous approach)

8¿increasing d distane in random point direction.to gt the new point and checking its validity(of pont) if its inside the frame and not on black and blue objects.else continue;

9¿ now generating line b/w last 2 nodes(newnode and nearest node) and checking its validity that all the points on the line joning these two points should also lie in white region;

10¿pushing newpoint in vector nodes and pushing (newpoint,nearstpoint) in edges;

11¿this loop will break when newpoint is our final point or else if total no. of nodes exceed a limit it will exit;

12¿now retracing the path.....a point tmp defined which refers to last point before final point(*(nodes.end()-1); and pushing it into a vector qu which has nodes containing fnal path now traversing in edges from final point and seeing which is connected to each another as ones paren will be childen of another going in reverse direction...and higlighting it with line if different color;

impleted succesfully on static image but getting unexpected result in video!!

VI. Results and Observation

I have tried both A* as well as rrt so,rrt will be more preferrable as A* finds shorter path and rrt faster but also we have to change our path according to the current situations i.e. moving obstacles .....we have to find another path to same final point .....in rrt it can be done easily as it is randomly generating trees which shows multiple ways to reach there ...but in A* it will check for next shorter path which may be blocked as well so it will take plenty time by finding all possible path..

VII. Future Work

implementation on video can be done in a better way...

CONCLUSION

task was to find path in a video with moving obstacles...first difficulty was deciding which method to choose ..I chose first A* but then used rrt...had to learn about these paths ,how they work,some of the predefined functions like line,and didn't know anything about vectors and struct.....initially was storin data in arrays but size of it was creating problem..in my first attempt was writing every code in a single main function....but was told to use different functions for easy understanding and eficient working....moving obstacle was challenging as it may go to the visited path as well.....different types of error occoured whle compiling prog...lot of segmentation fault and handling it.....

References

[1] wikipidea, https://en.wikipedia.org/wiki/Rapidly-exploring$_random_tree$github, $https://github.com/RoboJackets/rrt$

[2] https://www.geeksforgeeks.org/vector-in-cpp-stl/

[3] https://www.geeksforgeeks.org/iterators-c-stl/

[4] https://2019.robotix.in/tutorial/pathplanning/rrtplanner/

[5] https://www.youtube.com/watch?v=Ob3BIJkQJEwt=68s