This is the final submission folder for task 1.

There are 8 code files supplied namely,

1) final\_1.cpp → A\* path planning.

2) final\_2.cpp → dilation applied for obstacle inflation

3) final\_2\_new.cpp → creates path for a particular given orientation of bot by obstacle inflation.

4) configuration\_space\_generator.cpp → generates configuration space for a particular desired orientation

5) final\_3.cpp → This code does not take into account the 3D CONFIGURATION SPACE for bot. Just uses simple dilation for obstacle inflation and angle between final path points for bot orientation

6) smoothing.cpp → This code uses dilation for obstacle inflation and somewhat smoothens the path for bot to move without much jerking.(The reason for using dilation is -> The angles after smoothing might change from 0,45,90,135 to any arbit values and I have made configuration space only for these 4 angles of bot).

7) bitwise\_and.cpp → This generates a configuration space(from the 4 configuration spaces), for bot, consisting of paths possible for at least one orientation of bot.

8) c\_space.cpp → This generates the path based on configuration space of bot and shows it.

I could not integrate this code and the “smoothing code” as here I have considered only 4 orientations of the bot while on smoothing, the angles might change hence the bot orientation may be erroneous. \*\*THIS CODE IS STILL UNDER IMPROVEMENT\*\* \*\*THE BOT AT SOME POINTS DRIFT SIDEWAYS INSTEAD OF CHANGING ORIENTATION AND MOVING PARALLEL TO THE PATH\*\*

One Makefile is also supplied.

\*\*Please use the supplied Makefile for compilation purpose\*\*

The original Makefile was modified by me to include the flag "-std=c++11" for the bool operator to work smoothly, as the syntax used for the bool operator is supported by c++11.

A video is also supplied for subtask\_3. It shows the movement of bot with orientation consideration. Here the bot is in white color.

A video is also supplied for smoothing.cpp .

Subtask\_x is the code file named final\_x or (final\_x\_new + configuration\_space\_generator) (x=1,2,3).

To execute any code file, pass the obstacle map as command-line argument(Ex- $ ./final\_1 image.png ).

\*\*\* For final\_2\_new pass-The original obstacle map and configuration space generated.\*\*\*

\*\*\*For configuration\_space\_geneator.cpp pas-The original obstacle map and desired orientation of bot (int degrees).

I have shown the instantaneous traversal going on under the window "instant".

Final output is shown afterwards.

For Subtask\_3, animation of bot(Here, blue color) moving around the map will be displayed once the destination is found.

All the three final\_x.cpp codes upon execution, after final termination of namedWindows by the user, prints the path length in PIXELS and total points traversed(\*\*total points SEARCHED by the algorithm\*\*) in the terminal.

Hope you like my effort =)

Thank You :)

Submitted by : Deepank Agrawal 17CS30011