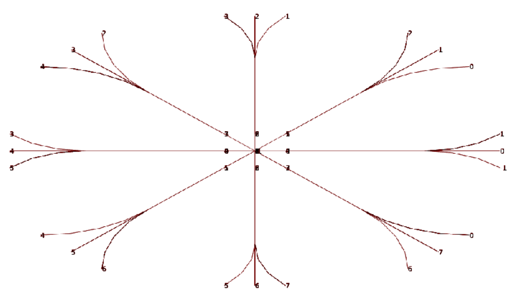
Multi-Resolution Path Planning in Constrained Environments

INTRODUCTION:

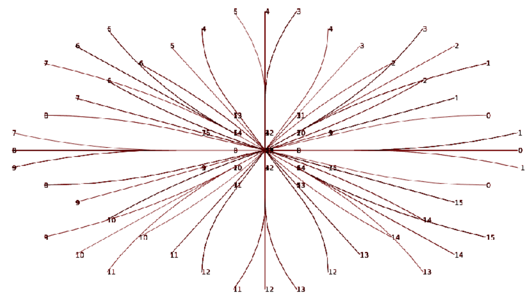
Mobile robot motion planning in large and dynamic environments is very challenging. In real-world scenarios with wide unstructured areas, quickly generating path plans that are as optimal as possible given constraints of minimal time, distance, computational cost, or energy is extremely important. Precise Plan in constrained environments requires planning in high resolutions whereas in large open areas and in the far future of the path, planning can be done more coarsely. So to balance both optimality and computational cost, planning in a multi-resolution lattice-based search is explored in this project.

APPROACH:

The model of the robot contains state space and action space. 3-dimensional state space (x,y, θ) is considered for this problem. Predefined motion primitives are used as action space. For this project, motion primitives are generated for both resolutions based on the sbpl library. For each θ in a grid cell, there are 7 feasible actions possible for the robot. In this project, two levels of resolution are considered. High resolution with 16 possible directions in θ and low resolution is with 8 possible directions in θ. The total number of actions in high resolution would be 112 and in low resolution would be 56. The same is shown in the map below.



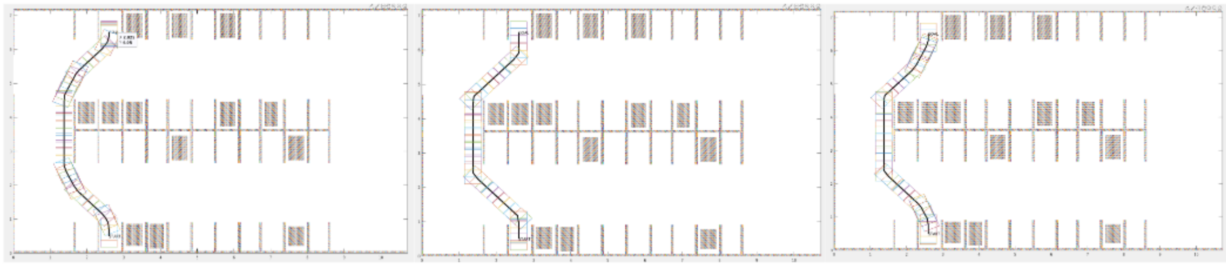
a) Low-resolution lattice - 7 actions per direction and in 8 directions



b) high-resolution lattice - 7 actions per direction and in 16 directions

The low-resolution primitive set is a strict subset of high-resolution motion primitives. This is extremely important in order to make the transition between high resolution to low resolution and vice versa. Transitions from low resolutions to high resolutions are trivial. As each member of Low-resolution lattice is also a member of the high-resolution lattice. For the transitions from high resolution to low resolution, only a specific set of motion primitives from high resolution are valid in low resolution. We only allow the transitions which are valid from high resolution and discard other motion primitives.

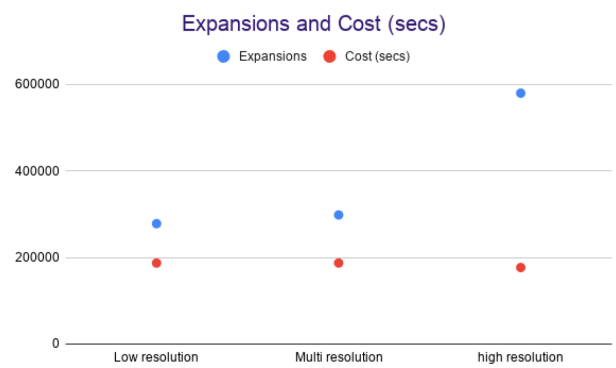
Areas within a predefined radius around robot start position and robot goal position are searched in high resolution and the rest of the path is searched in low resolution. The maximum of 2d search and Euclidean distance is considered as heuristics. The cost function is defined as (the amount of time the car is taking to travel) x (cost of action). This project is built upon the SBPL library. Modifications are done mainly in main.cpp, environment_navxythetalat.cpp files and araplanner.cpp files. The environment has been modified so that the library can take two motion primitive files and store the set of actions defined. Further, araplanner has been modified to take care of transitions from one resolution to another resolution. The multiresolution algorithm has been tested on a map of 10 meters x 7 meters with a car footprint as 0.45 meters x 0.65 meters. On executing the algorithm, a solution file is generated by the solver. This solution file is then passed on to Matlab to visualize the solution. The map, the footprint of the car, and the Plans generated by the algorithm are shown in the figure below.



a) High resolution b) Low resolution c) Multi resolution

CONCLUSIONS:

For the comparison, a set of start and goal positions are tested in only high resolution, only low resolution and in multiresolution. Average results of 10 experiments are calculated and below are the average results. Details of all the experiments are mentioned in an excel file for reference. We can see from the graph that the number of expansions and planning time with high resolution is almost double the number of expansions and planning time in low resolution. The cost of the solution of high resolution is less than that of low resolution. This difference would be very high if the area is extremely constrained. Sometimes low-resolution search may not find a solution. Multiresolution as we can see perfectly balances both the aspects. The average cost of the solution is almost equivalent to that of high resolution but with a ~ 7% increase in the number of expansions compared to low-resolution search. Choosing the zone of high resolution is crucial to achieving meaning full results. Currently, this radius was chosen by experimenting with the brute force method for this map. This is hugely environment-dependent. Automating this can be a further scope of work.



	Expansions	Cost (secs)	Planning time(secs)
Low resolution	278306.2	187221.5	8.0
Multiresolution	298381.8	187363.9	8.6
high resolution	579859.9	176876.3	15.7

Executing the code.

```
>> lakshmi@trex:~/code$ cd build
>> lakshmi@trex:~/code/build$ make
```

To execute the low-resolution search in the map

```
>>lakshmi@trex:~/code/build$ ./test_sbpl ../matlab/map8dir.cfg ../matlab/8dir.mprim
>> plot_3Dpath('sol.txt','map8dir.cfg',0.025,8,1) in Matlab to visualise solution
```

To execute the high-resolution search in the map

```
>>lakshmi@trex:~/code/build$ ./test_sbpl ../matlab/map16dir.cfg ../matlab/16dir.mprim
>> plot_3Dpath('sol.txt','map16dir.cfg',0.025,16,2) in Matlab to visualise solution
```

To execute a multi-resolution search

```
>>lakshmi@trex:~/code/build$ ./test_sbpl ../matlab/map8dir.cfg ../matlab/8dir.mprim ../matlab/16dir.mprim
>> plot_3Dpath('sol.txt','map16dir.cfg',0.025,16,3) in Matlab to visualise solution
```