Chapter 2.2 Report for A\* planning algorithms by ROS

User ID: rabbit5024

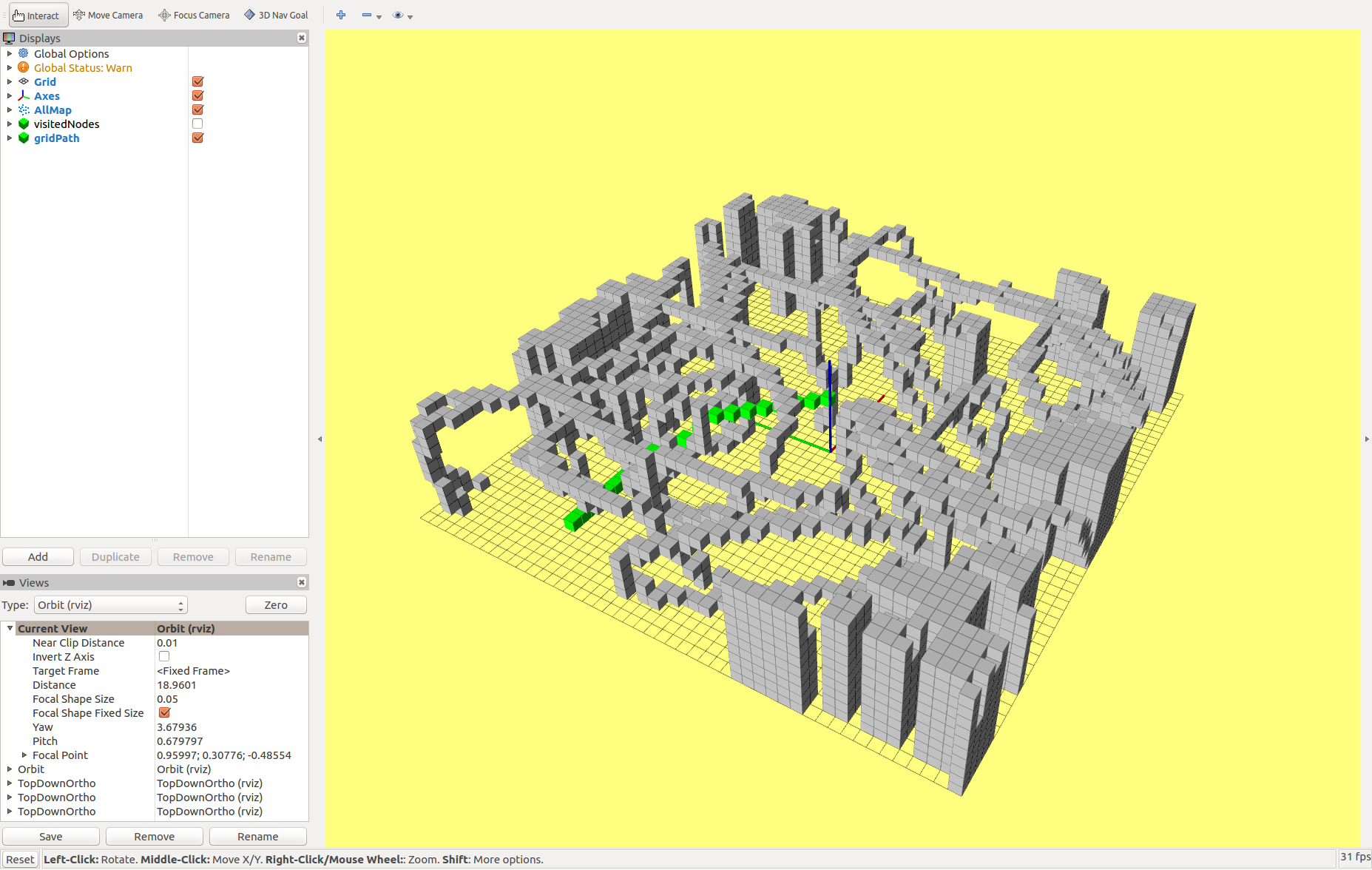
1. Algorithm Flowchart

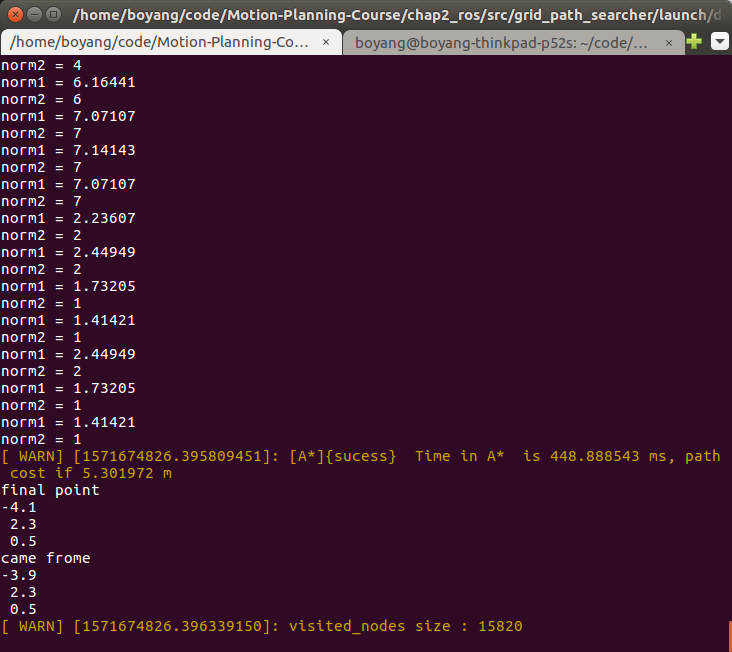
When the main() function in demo\_note.cpp calls “\_astar\_path\_finder->AstarGraphSearch(start\_pt, target\_pt);”, the planning function receive the start and end points of planning task.

Then the “AstarGraphSearch” fuction in “Astar\_searcher.cpp” start.

1. Recode current time time\_1
2. Initial the start point pointer startPtr
3. Main loop for expanding
   1. if open list is empty or reach the goal, finish the cicle
   2. get the minimum f node from the open list, delete it and mark it as visited
   3. get all the neighbours of current node
   4. decide whether to put the neighbours to the open list or change the g value in the open list
4. Get current time time\_2 and print the search time
5. Track the path back from the terminatePtr to the start
6. Show the path at RVIZ interface

2. Planning Results





3. Comparison of different heuristic functions for the performance of A star planning

3.1 Euclidean

3.2 Manhattan

3.3 Diagonal Heuristic

3.4 The effect of Tie Breaker

4. The comparison of A\* and JPS (when to use A\* and when to use JPS)

Optional: finish src/grid\_path\_searcheer/src/readonly/JPS\_searcher.cpp (void JPSPathFinder::JPSGraphSearch(...))