

Matlab 实现 RRT 作业

1. 打开工作文件夹，按照 STEP 完成 RRT.m

C++ 实现 RRT* 作业

2. 创建工作空间；

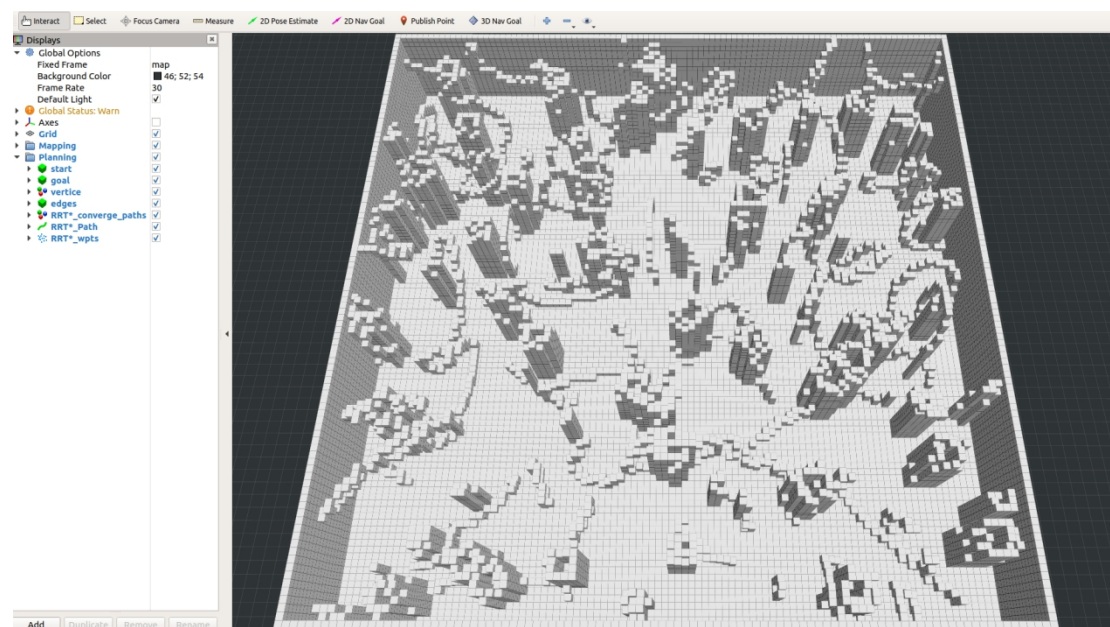
3. 将此 src 文件夹放入工作空间，使用 catkin_make 命令进行编译，

编译成功后，在 terminal 中运行：

```
source devel/setup.bash
```

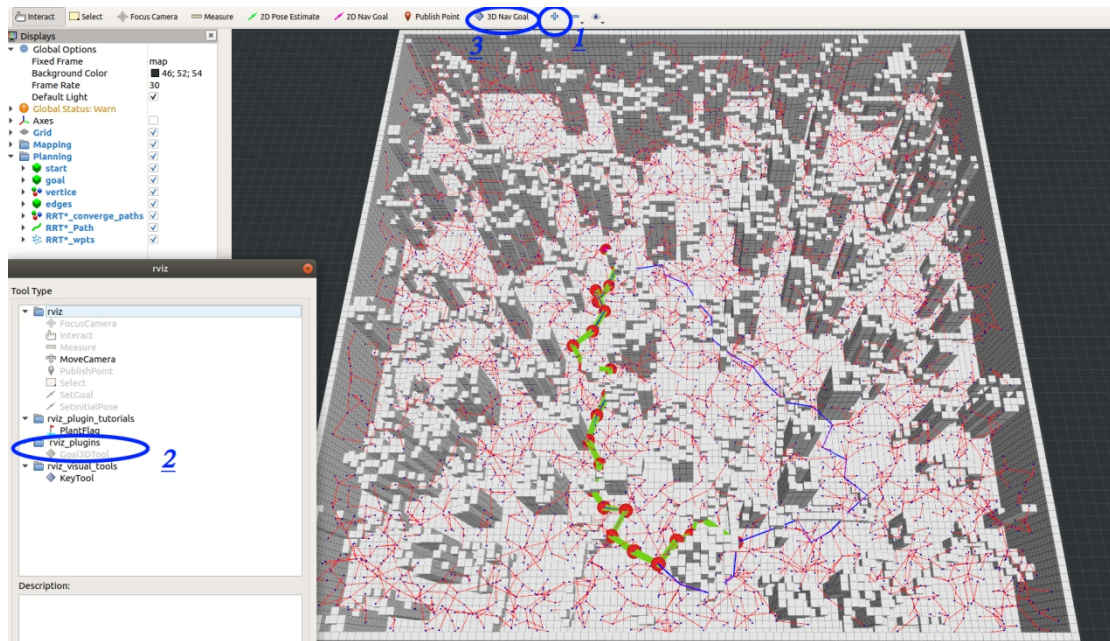
```
roslaunch path_finder rviz.launch
```

若 rviz 成功打开，将得到如下界面：



之后添加 Rviz 插件 3D Nav Goal，可使用鼠标或者键盘的 g 按键设置规划的目标点。

将得到 RRT 的规划结果。



4. 找到 `src/path_finder/include/path_finder/rrt_star.h` 文件 , 修改此文件中两处代码 :

1) 实现 ChooseParent

```

275 // TODO Choose a parent according to potential cost-from-start values
276 // ! Hints:
277 // ! 1. Use map_ptr->isSegmentValid(p1, p2) to check line edge validity;
278 // ! 2. Default parent is [nearest node];
279 // ! 3. Store your chosen parent-node-pointer, the according cost-from-parent and cost-from-start
280 // !    in [min_node], [cost_from_p], and [min_dist from start], respectively;
281 // ! 4. [Optional] You can sort the potential parents first in increasing order by cost-from-start value;
282 // ! 5. [Optional] You can store the collision-checking results for later usage in the Rewire procedure.
283 // ! Implement your own code inside the following loop
284 for (auto &curr_node : neighbour_nodes)
285 {
286     // ! Implement your own code inside the above loop
287 }

```

2) 实现 Rewire

```

320 /* 3.rewire */
321 // TODO Rewire according to potential cost-from-start values
322 // ! Hints:
323 // ! 1. Use map_ptr->isSegmentValid(p1, p2) to check line edge validity;
324 // ! 2. Use changeNodeParent(node, parent, cost from parent) to change a node's parent;
325 // ! 3. the variable [new_node] is the pointer of X_new;
326 // ! 4. [Optional] You can test whether the node is promising before checking edge collision.
327 // ! Implement your own code between the dash lines [-----] in the following loop
328 for (auto &curr_node : neighbour_nodes)
329 {
330     double best_cost_before_rewire = goal_node->cost_from_start;
331     // ! -----
332     // ! -----
333     if (best_cost_before_rewire > goal_node->cost_from_start)
334     {
335         vector<Eigen::Vector3d> curr_best_path;
336         fillPath(goal_node, curr_best_path);
337         path_list.emplace_back(curr_best_path);
338         solution_cost_time_pair_list.emplace_back(goal_node->cost_from_start, (ros::Time::now() - rrt_start_time).toSec());
339     }
340 }
341 }

```

5. 再次编译

6. 运行

关闭之前所开的 launch ,

重新在 terminal 中运行 :

```
source devel/setup.bash
```

```
roslaunch path_finder rviz.launch
```

若代码编写无误，将得到 RRT* 的规划结果，效果如下：

