CSE-525 Introduction to Robotics

Final Project Report

Implementation and Evaluation of Geometric Motion Planners

Instructor: Dimitrios Samaras

Aim of this project is to implement and evaluate the geometric motion planners RRT and RRTConnect.

Geometric Motion Planners chosen:

- **RRT** (Rapidly Exploring Random Trees)
- **RRTConnect** (Bi-directional RRT)

Problem Statement:

To determine the best motion planner by implementing and evaluating geometric motion planners like RRTConnect and RRT based on the parameters such as planning time, trajectory length and number of nodes in the path and applying path simplifiers to check if the results are varied.

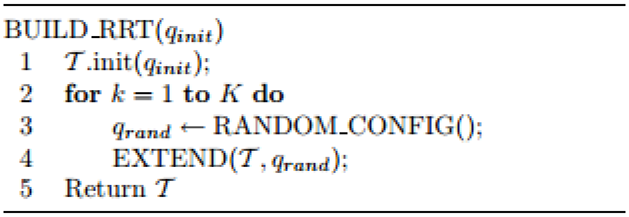
List of sub problems:

- Implement chosen motion planners
- Implement path simplifiers for the chosen motion planners
- Compare the results based on the evaluation parameters for a simple experiment

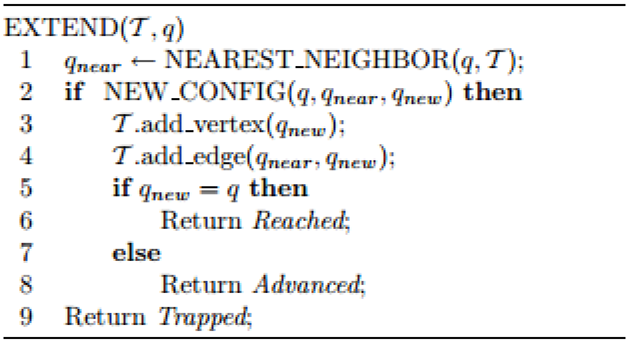
RRT (Rapidly Exploring Random Trees):

Planner typically grow a tree of states connected by valid motions. These valid motions randomly reach the goal state giving a path.

Basic RRT algorithm:



```
BUILD_RRT($q_{init}$)
    $\mathcal{T}$.init($q_{init}$);
    for $k = 1$ to $K$ do
        $q_{rand} \leftarrow$ RANDOM_CONFIG();
        EXTEND($\mathcal{T}, q_{rand}$);
    Return $\mathcal{T}$
```



```
EXTEND($\mathcal{T}, q$)
    $q_{near} \leftarrow$ NEAREST_NEIGHBOR($q, \mathcal{T}$);
    if NEW_CONFIG($q, q_{near}, q_{new}$) then
        $\mathcal{T}$.add_vertex($q_{new}$);
        $\mathcal{T}$.add_edge($q_{near}, q_{new}$);
        if $q_{new} = q$ then
            Return *Reached*;
        else
            Return *Advanced*;
    Return *Trapped*;
```

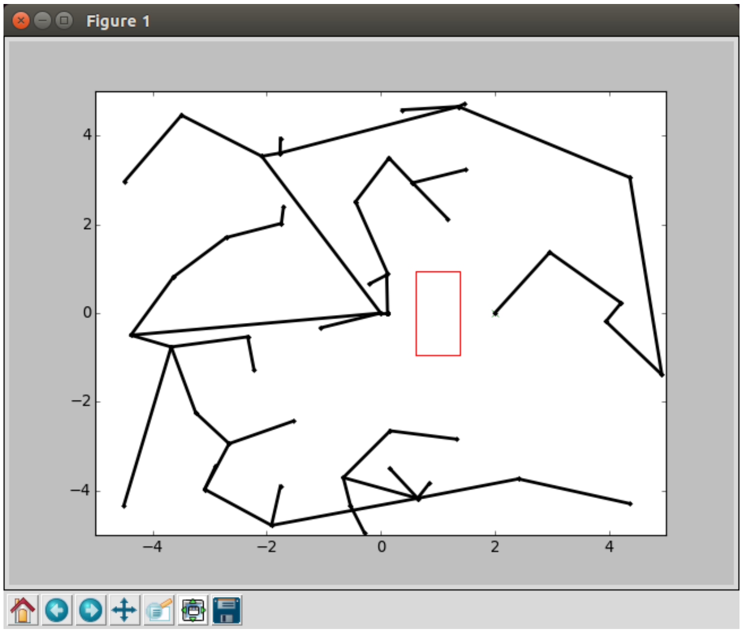
CSE-525 Introduction to Robotics

Final Project Report

Implementation and Evaluation of Geometric Motion Planners

Instructor: Dimitrios Samaras

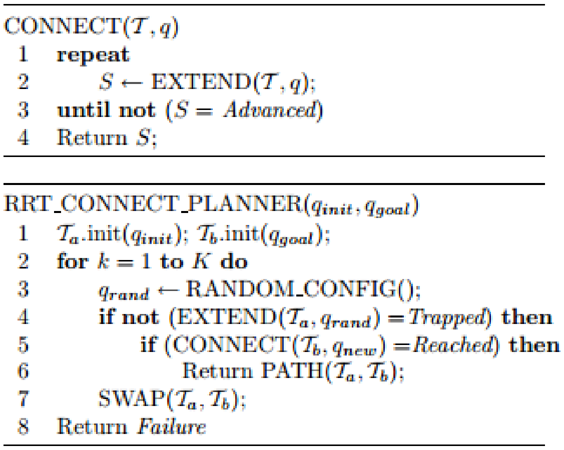
Example:



RRTConnect (Bi-directional RRT):

Bidirectional version of RRT which grow *two* trees: one from the start and one from the goal and then attempts to connect a state in the start with another state in the goal tree.

Basic RRTConnect algorithm:



```
CONNECT($\mathcal{T}, q$)
  repeat
      $S \leftarrow$ EXTEND($\mathcal{T}, q$);
  until not ($S =$ Advanced)
  Return $S$;
```

```
RRT_CONNECT_PLANNER($q_{init}, q_{goal}$)
  $\mathcal{T}_a$.init($q_{init}$); $\mathcal{T}_b$.init($q_{goal}$);
  for $k = 1$ to $K$ do
      $q_{rand} \leftarrow$ RANDOM_CONFIG();
      if not (EXTEND($\mathcal{T}_a, q_{rand}$) = Trapped) then
          if (CONNECT($\mathcal{T}_b, q_{new}$) = Reached) then
              Return PATH($\mathcal{T}_a, \mathcal{T}_b$);
      SWAP($\mathcal{T}_a, \mathcal{T}_b$);
  Return Failure
```

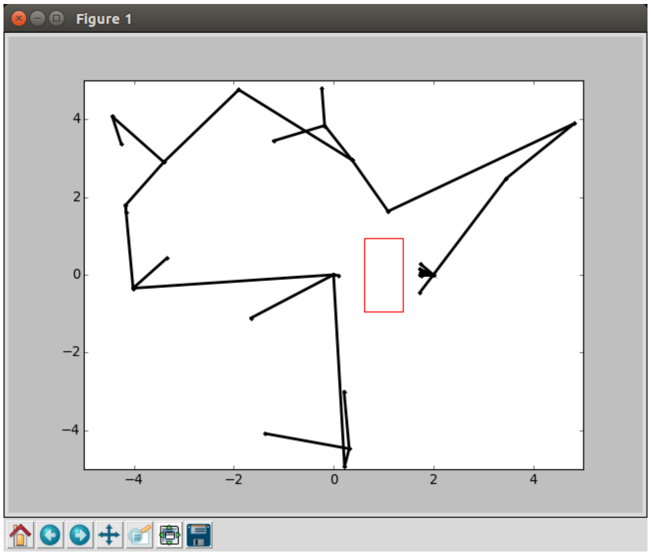
CSE-525 Introduction to Robotics

Final Project Report

Implementation and Evaluation of Geometric Motion Planners

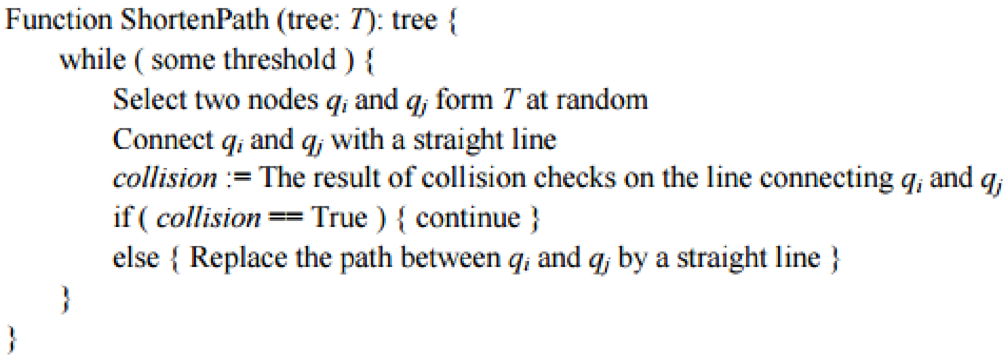
Instructor: Dimitrios Samaras

Example:



Path Simplifier

Plans generated by RRT and RRTConnect contain lot of extra necessary movements. Path simplifier helps to smooth the plan. A path shortening function is implemented in the environment as per the following algorithm



```
Function ShortenPath (tree: $T$): tree {
    while ( some threshold ) {
        Select two nodes $q_i$ and $q_j$ form $T$ at random
        Connect $q_i$ and $q_j$ with a straight line
        $collision$ := The result of collision checks on the line connecting $q_i$ and $q_j$
        if ( $collision$ == True ) { continue }
        else { Replace the path between $q_i$ and $q_j$ by a straight line }
    }
}
```

CSE-525 Introduction to Robotics

Final Project Report

Implementation and Evaluation of Geometric Motion Planners

Instructor: Dimitrios Samaras

Platform and Resources:

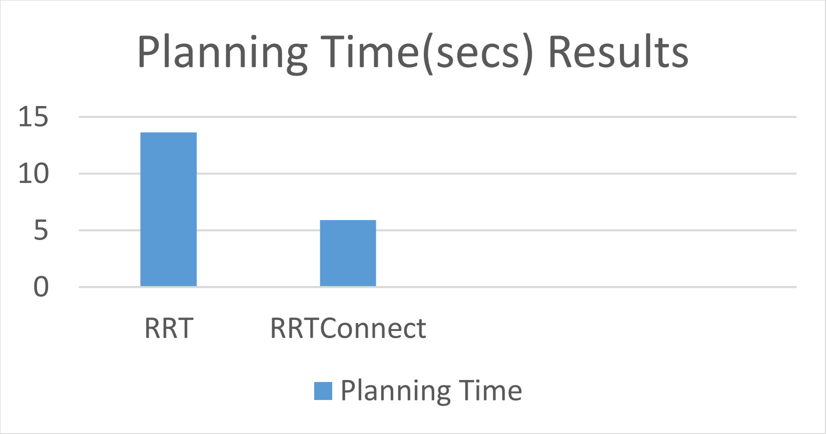
- OpenRAVE 0.8 or above for simulation of robot behavior and other capabilities

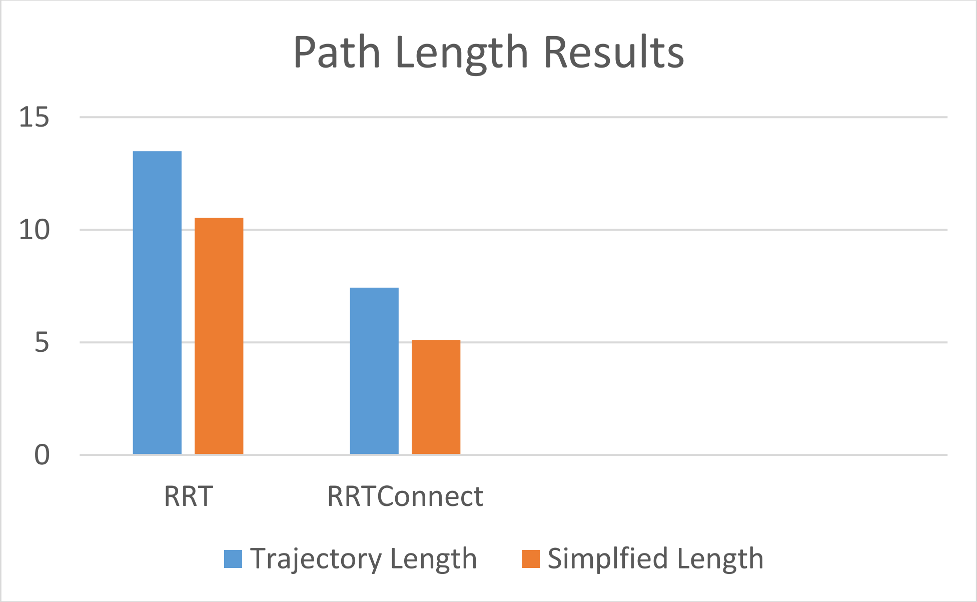
Experiment:

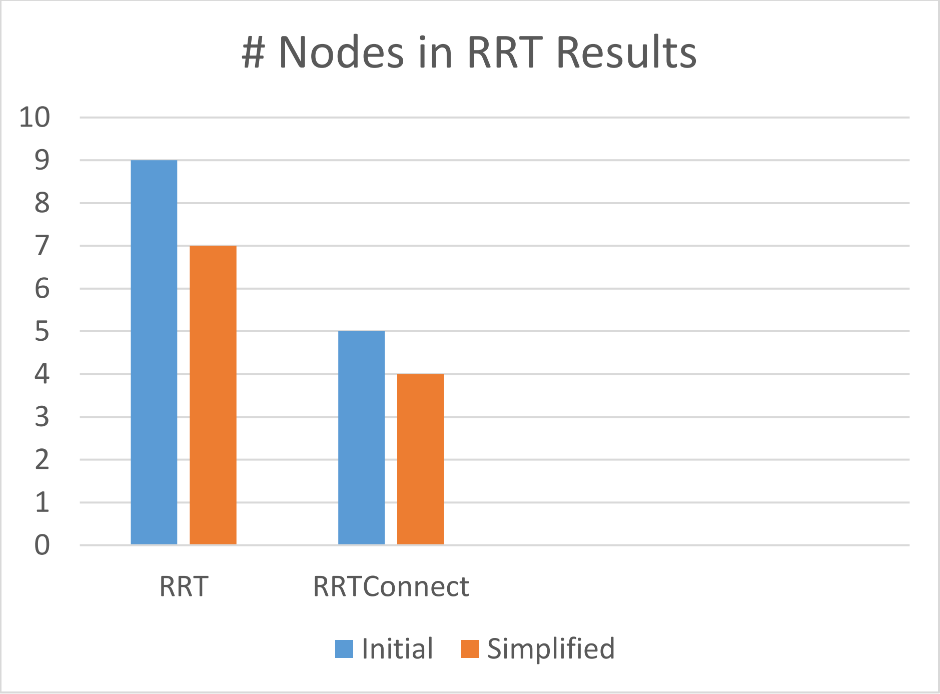
Simulation of a simple navigation experiment for a PR2 robot in 2 dimension space with table object as an obstacle.

Results:

Captured for 10 runs and average results are plotted







Conclusion:

As per the evaluation results in terms of various parameters RRTConnect algorithm outperforms the original RRT algorithm

References:

- http://ompl.kavrakilab.org/planners.html
- https://en.wikipedia.org/wiki/Rapidly_exploring_random_tree
- http://msl.cs.uiuc.edu/~lavalle/papers/Lav98c.pdf
- http://coecsl.ece.illinois.edu/ge423/spring13/RickRekoskeAvoid/rrt.html

CSE-525 Introduction to Robotics

Final Project Report

Implementation and Evaluation of Geometric Motion Planners

Instructor: Dimitrios Samaras