16-782: Planning and Decision-Making in Robotics

Homework 2

Professor: Maxim Likhachev *Submitted by: "nsanghvi"*

Figures:

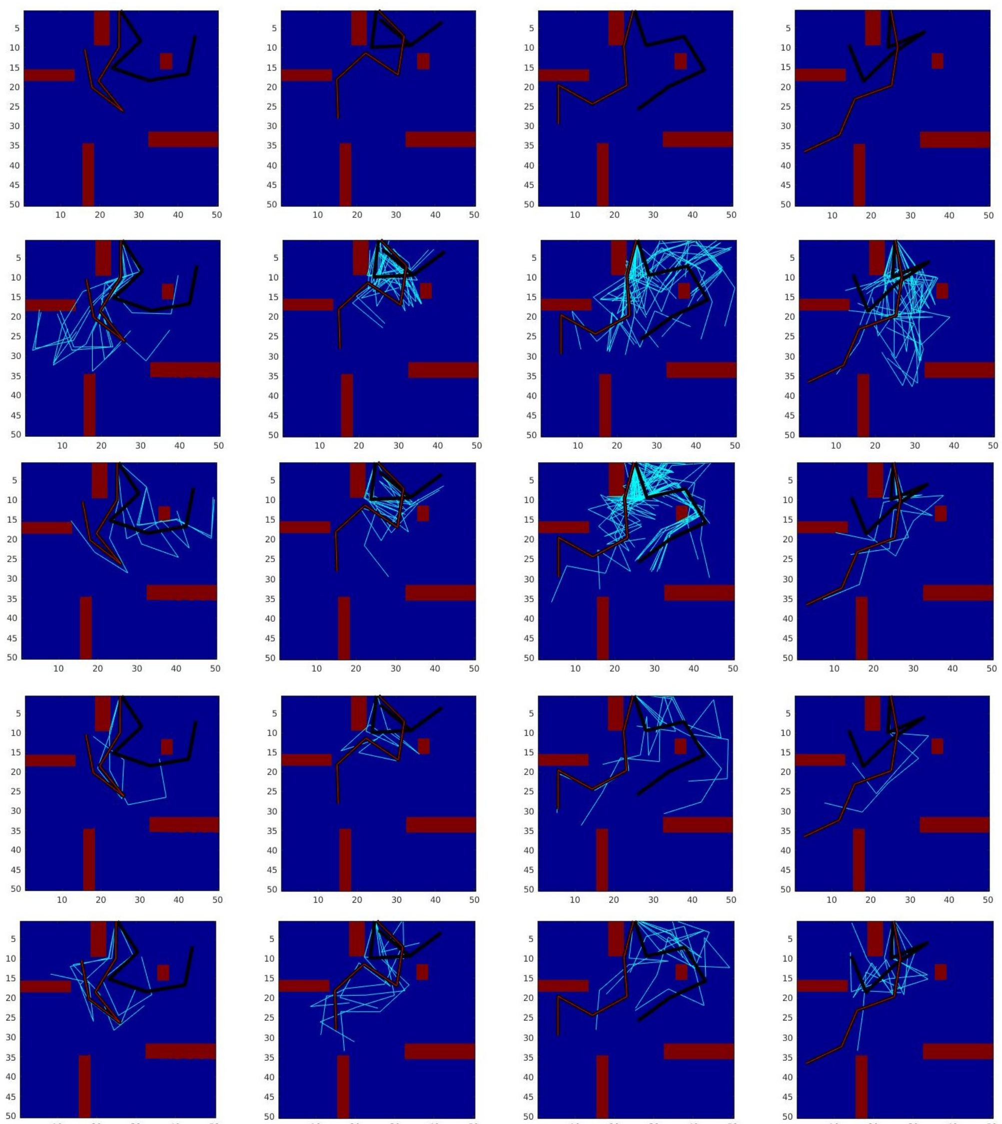

Figure 1: Paths returned by algorithms RRT, RRT-Connect, RRT*, PRM in a particular run on 4 example configurations (black = start, maroon = goal). Each row corresponds to an algorithm.

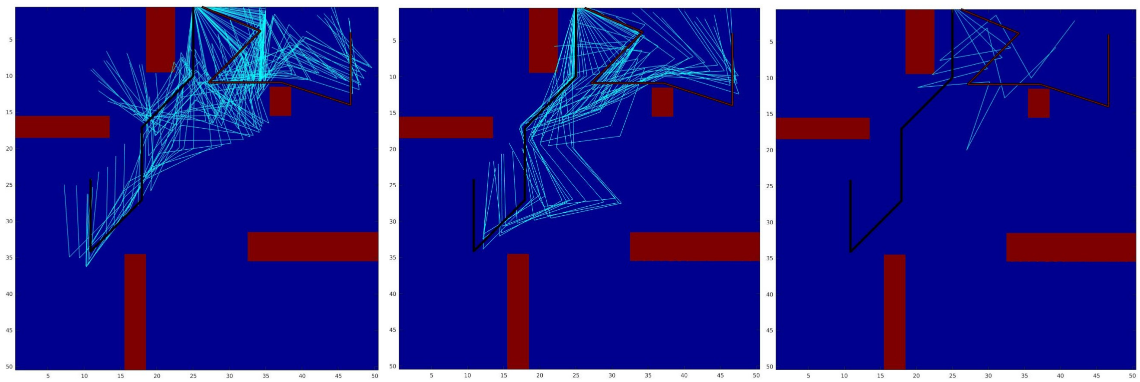

Figure 2: Constrast among paths returned by algorithms RRT, RRT-Connect, RRT*, in a run with low extend radius = 0.2 rad.

Test Statistics:

	Statistic	RRT	RRT-Connect	RRT*	PRM
(1)	All reached?	**Yes**	**Yes**	**Yes**	**Yes**
(2)	Mean no. of steps till goal	16.30	12.25	**4.65**	6.90
(3)	Mean no. of samples (till goal/soln. found)	533.05	**56.85**	871.20	N/A
(4)	Mean no. of samples (total)	533.05	**56.85**	2875	5000
(5)	Mean planning time (s)	0.407	**0.007**	6.961	12.097
(6)	Mean path cost (rad)	9.889	8.414	**4.544**	8.503

Table 1: Algorithms' test statistics (wins in bold)

Implementation:

All hyperparameter values were chosen empirically.

- RRT: Radius considered for extend = 1.0 rad, validity of extended path checked at a discretization of 0.1 rad, goal bias = 5%, with a small goal perturbation, sample upper limit set at 5000 samples.
- RRT-Connect: Radius considered for extend = 1.0 rad, validity of extended path checked at a discretization of 0.1 rad, goal (or start) bias = 5%, with a small goal (or start) perturbation, sample upper limit set at 5000 samples.
- RRT*: Radius considered for extend = 1.0 rad, validity of extended path checked at a discretization of 0.1 rad, goal bias = 5%, with a small goal perturbation, radius considered for rewiring = 2.0 rad, sample upper limit set at 5000 samples.
- PRM: Radius considered for neighbor = 2.0 rad, validity of extended path to neighbor checked at a discretization of 0.1 rad, maximum degree of vertex = 6, sample hard limit set at 5000 samples.

Discussion:

In terms of planning time, from Table 1, we see that **RRT-Connect** < **RRT** < **RRT*** < **PRM**. This is as expected, since:

- RRT-Connect grows two trees toward each other, and, as we can see from rows 3 and 4, the number of samples it requires before a solution is found is much lower than the others.
- RRT* has a higher planning time than the other RRT algorithms as it requires two rewiring operations and cost look-ups in each of these operations. Moreover, rows 3 and 4 differ here as planning has not been stopped as soon as the goal is found. Samples continue to be added in hope of finding a lower-cost path.
- PRM has high (amortized) time as a large graph is constructed before-hand so that a path between any start-goal pair may be found.

In terms of path costs, from Table 1, we see that **RRT*** < **RRT-Connect** < **PRM** < **RRT**. This is about as expected, since:

- RRT* spends time planning in terms of best-cost paths while the other RRT algorithms do not. Moreover, it considers more samples (and, hence, potential paths) than them on average since sampling is not stopped once the goal is found. Empirically, variance in best-cost path found by RRT* was low.
- RRT, PRM and RRT-Connect have similar path costs, with RRT-Connect performing slightly better. This is likely due to the unlimited extend operation considered when joining trees, which allows bigger jumps between configurations in RRT-Connect. However, this may not be significant as we empirically observed high variance in path-costs found by RRT and RRT-Connect algorithms, since they do not explicitly try to optimize cost.
- PRM does not have explicit bias sampling toward a goal and is not tailor-made for any start/goal pair. This is likely why, even though it finds paths with lesser number of steps and considers path cost optimization, it performs similar to RRT and and RRT-connect. We believe it will perform better if we add more samples to the graph, which we did not experiment with due to time constraints.

In terms of steps to goal, from Table 1, we see that **RRT*** < **PRM** < **RRT-Connect** < **RRT**. This is as expected, since:

- RRT* spends considerable time rewiring within a decently large radius. While this neighbor radius considered by PRM is the same, RRT* has the added advantage of sampling biased to the goal, and is tailor-made for a particular start-goal pair. From Figure 1, we see how much smaller the paths for RRT* are (4th row).
- RRT-Connect performs slightly better than RRT as it is allowed an unlimited jump between trees (if it's a valid path). From Figure 1, while RRT-Connect might have a difficult time with multiple close obstacles (example 3, since it would not be able to connect trees as easily), it is seen to have visibly shorter paths than RRT on examples 1 and 4 due to this reason.

- The contrast between versions of RRT algorithms is best seen in Figure 2, where we have intentionally set a low extend radius and a start-goal pair with a seemingly difficult path among obstacles. RRT-Connect is seen to have a jump in the middle which helps its pah length, while RRT* continues almost immediately to the vicinity of the goal due to rewiring. For RRT, RRT-Connect and RRT*, Path lengths: 60, 38, 6. Path Costs: 11.3, 9.4, 4.5

Running Instructions:

In MATLAB run (all in one line):

```
mex planner.cpp GraphVertex.cpp GraphEdge.cpp Compare.cpp
PlanSample.cpp Rrt.cpp RrtConnect.cpp RrtStar.cpp PRMap.cpp
```

before

```
runtest('mapX.txt', start, goal, planner_id)
```
