SR-RRT:
Selective Retraction-based RRT Planner

Junghwan Lee[1], OSung Kwon[1], Liangjun Zhang[2], Sung-eui Yoon[1]

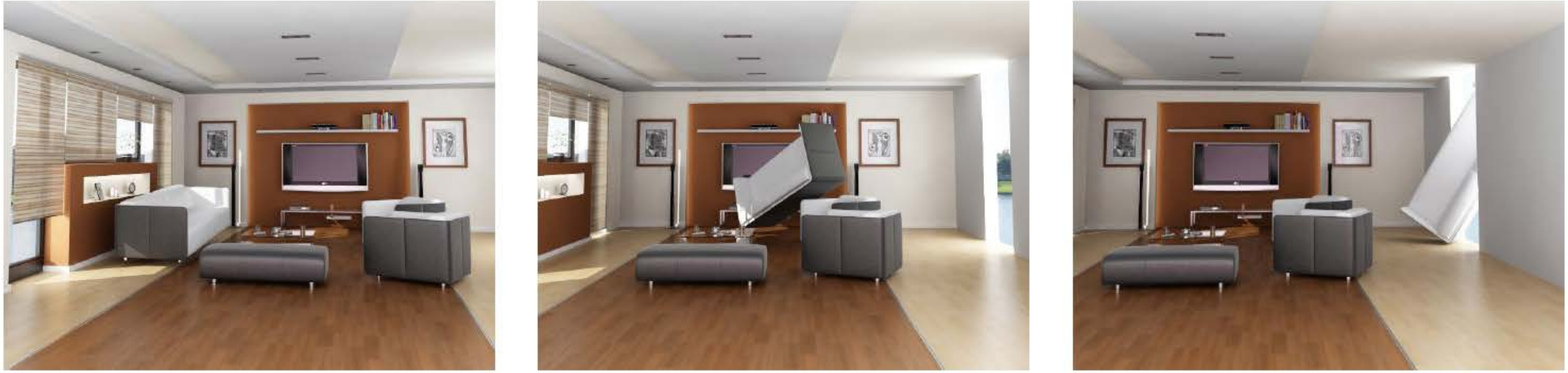
[1]Dept. of CS, KAIST

[2]Dept. of CS, Stanford University

http://sglab.kaist.ac.kr/SRRRT/

Problem statement

- Motion planning problem



  - Single query
  - Rigid-body robot
  - No dynamic / kinematic constraints
  - 3D workspace including narrow passages

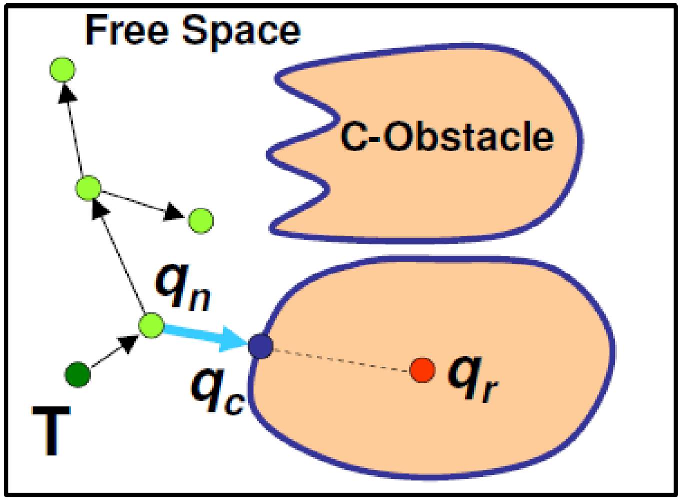
Sampling-based planner

- PRM [Kavraki et al. 96], RRT [Kuffner & LaValle, 00]

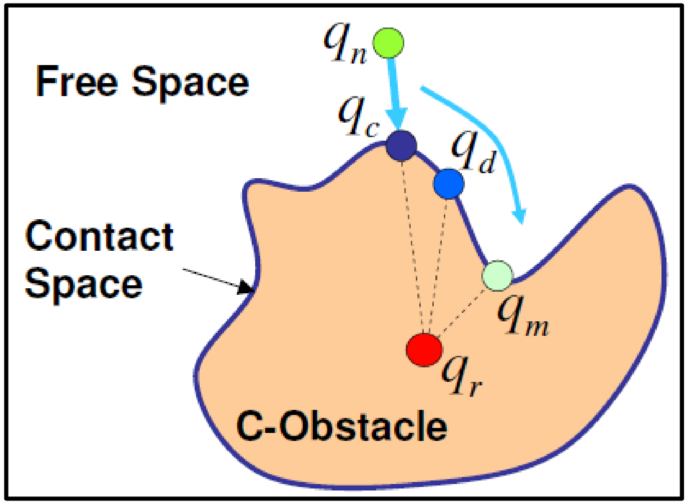
- Handling narrow passages
  - Filtering technique
    - Adaptive sampling by filtering out some samples
    - Gaussian PRM [Boor et al. 99], Visibility PRM [Simeon et at. 00], Bridge test [Hsu et al. 03], Ball Tree [Shkolnik & Tedrake 11], etc.
  - Retraction-based technique
    - Retracts in-collision samples towards more useful regions
    - SSRP [Saha & Latombe 05], [Redon & Lin 05], OBRRT [Rodriguez et al. 06], RRRT [Zhang & Manocha 08], etc.

Retration-based RRT
[Zhang & Manocha 08]

- **Retraction-based RRT technique handling narrow passages**



Standard RRT



RRRT

image from [Zhang & Manocha 08]

- **Iteratively retracts a sample near the contact space while minimizing the distance to original sample**

Retration-based RRT [Zhang & Manocha 08]

- **Retraction-based RRT technique handling narrow passages**

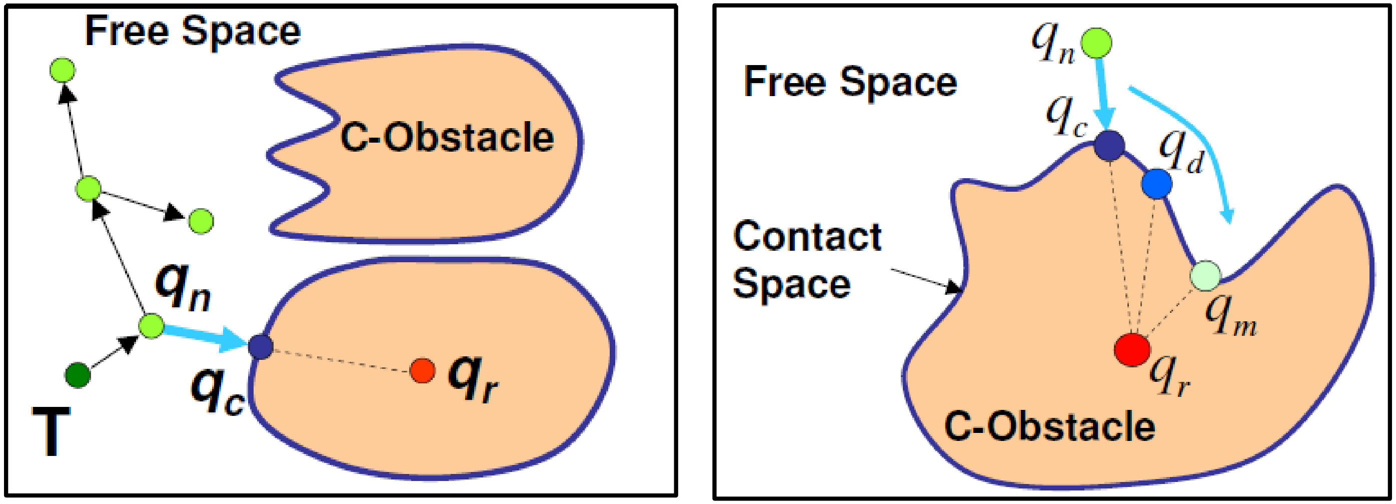
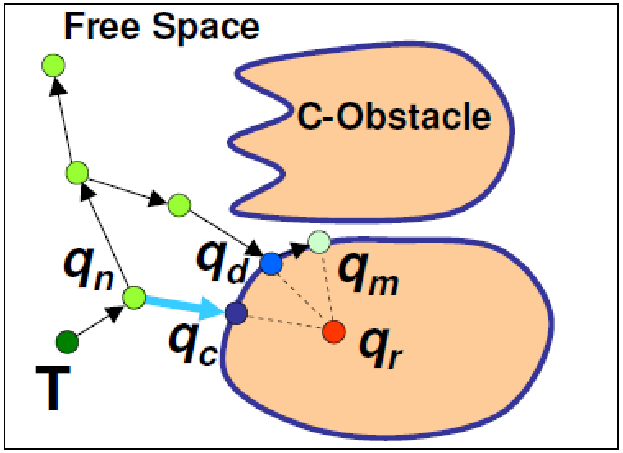


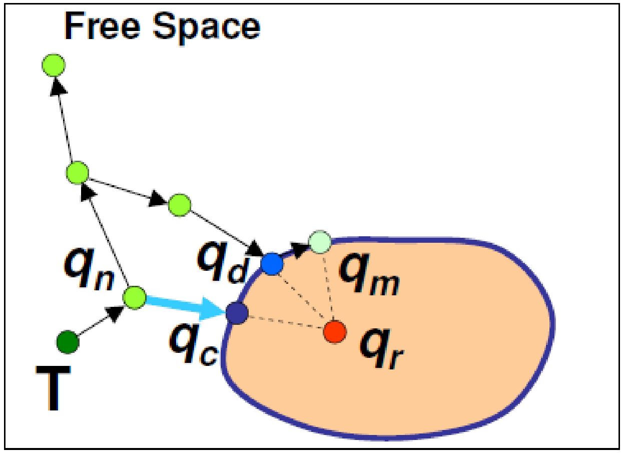
image from [Zhang & Manocha 08]

- **General characteristic:**
  Generates more samples near the boundary of obstacles

RRRT: Pros and Cons

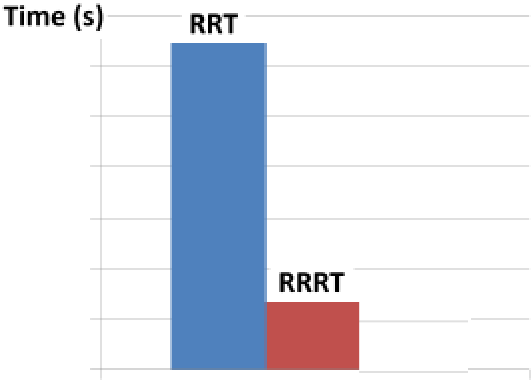


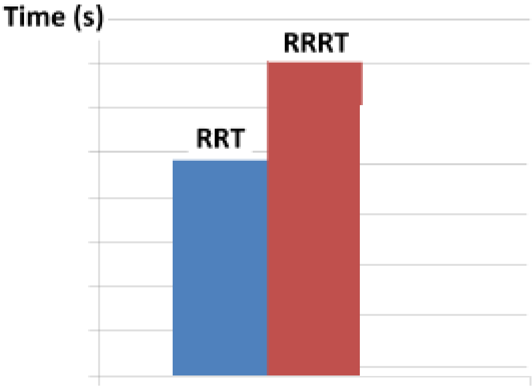
with narrow passages



without narrow passages

images from [Zhang & Manocha 08]





Motivation

- Cons of RRRT:
  Excessive sampling on the boundary of obstacles
  - Computational overhead
  - Not helpful for the problem without narrow passages

- If a planner can identify narrow passages,
  it can selectively perform retraction operations
  only on narrow passages

SR-RRT

- **Selective Retraction-based RRT**

- **Can efficiently handle various types of problems that have or do not have narrow passages**

	w/ narrow passages	**wo/ narrow passages**
RRT	Slow	Fast
RRRT	Fast	Slow
Ours	**Fast**	**Fast**

SR-RRT

- **Bridge line-test**

- **Non-colliding line-test**

Bridge line-test

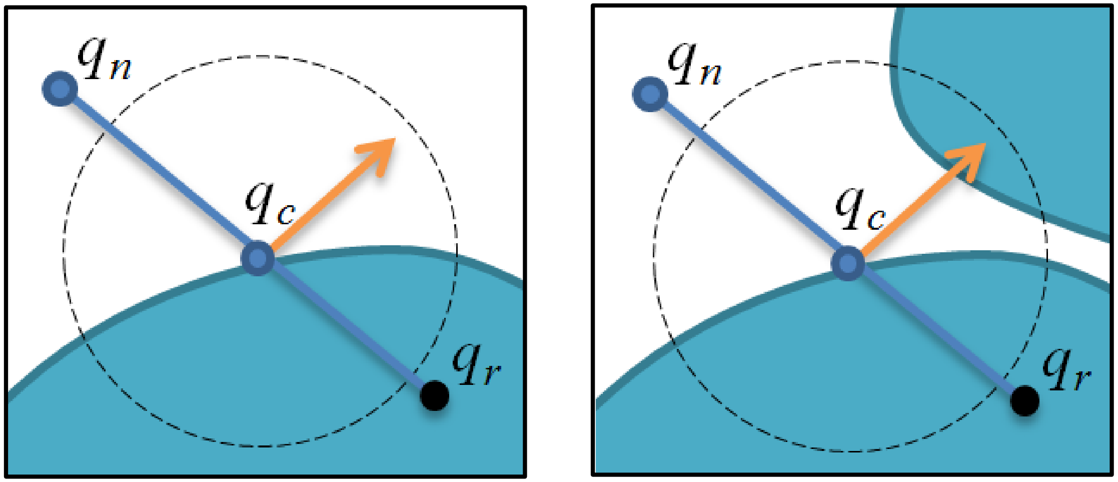
- To identify narrow passage regions
- To determine whether retraction operations will be performed

- Bridge line-test
  1. Generate a random line
  2. Check whether the line meets any obstacle
  - Inspired by Bridge test [Hsu et al. 03]

Bridge line-test

- To identify narrow passage regions

- Bridge line-test
  1. Generate a random line
  2. Check whether the line meets any obstacle



Bridge line-test

- **Approximation of hypersphere search to line search**

- **It can fail with some probability to identify a narrow passage region (false negative)**

- **Re-test Bridge line-test at a node where the tree expansion is stuck during iteration**
  - **in-contact node selected as nearest neighbor of sample**

- **The accuracy of the Bridge line-test become probabilistically ensured after multiple re-testing**

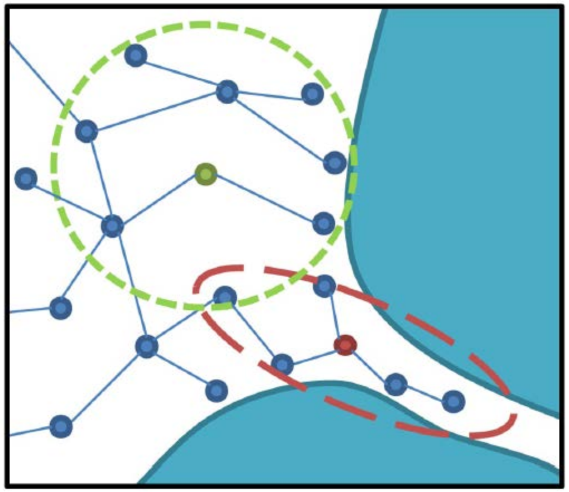
For high-dimensional space

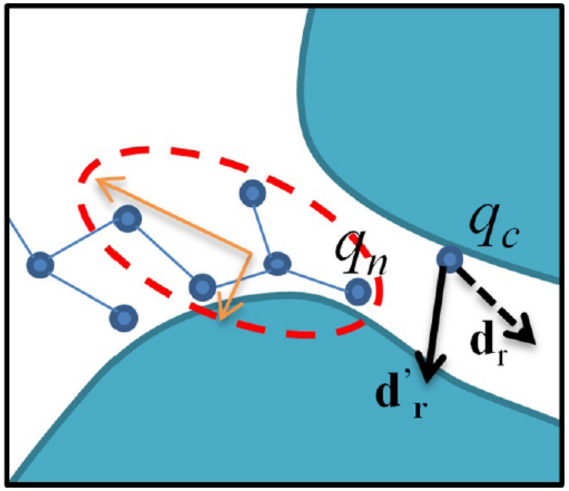
- Use the local free space distribution to improve the accuracy of Bridge line-test
  - measured by PCA (Principle Component Analysis)
  - PCA-RRT [Dalibard & Laumond 11]

For high-dimensional space

- Use the local free space distribution to improve the accuracy of Bridge line-test

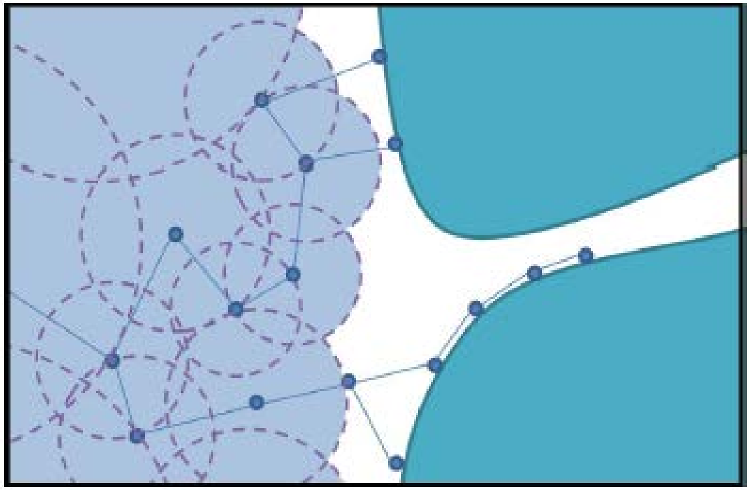
- Treat a region to be near narrow passages when the local free space is biased





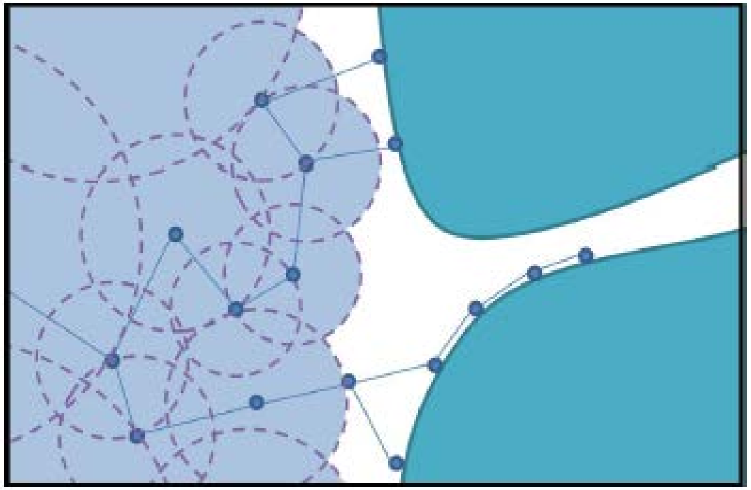
Non-colliding line-test

- Sampling bias technique using a dual operator to the Bridge line-test
  - More samples potentially towards narrow passages
  - Reject samples on wide-open free space
- Free hypersphere:
  - Approximation of wide-open free space
  - Center is a node in the tree, radius is distance to NN node



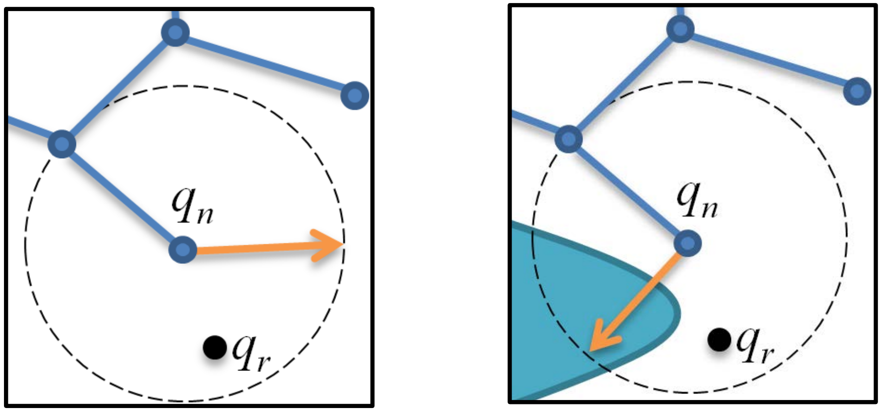
Non-colliding line-test

- A sample inside a free hypersphere is discarded
- Non-colliding line-test determines whether it's free
    - Very similar to the Bridge line-test



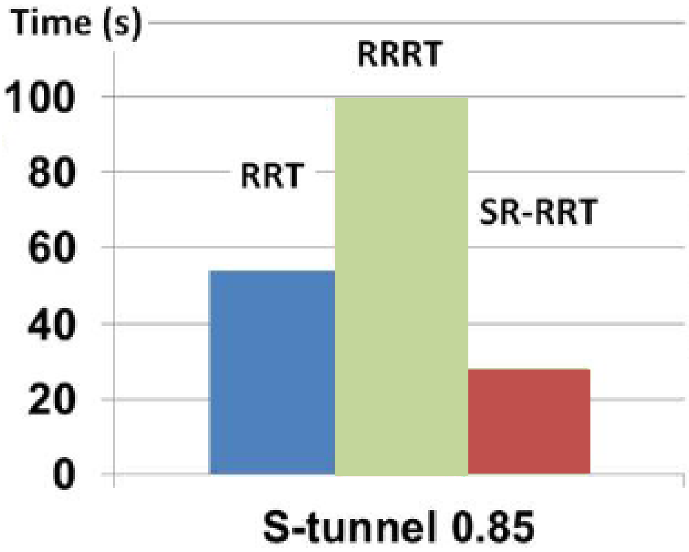
Non-colliding line-test

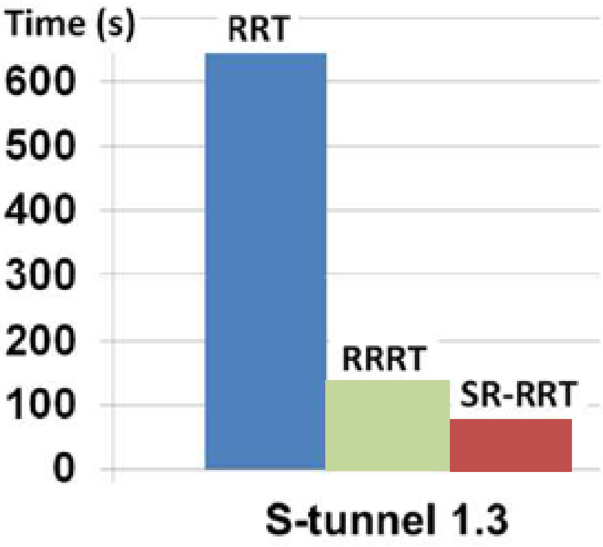
- A sample inside a free hypersphere is discarded
- Non-colliding line-test determines whether it’s free
  - Very similar to the Bridge line-test

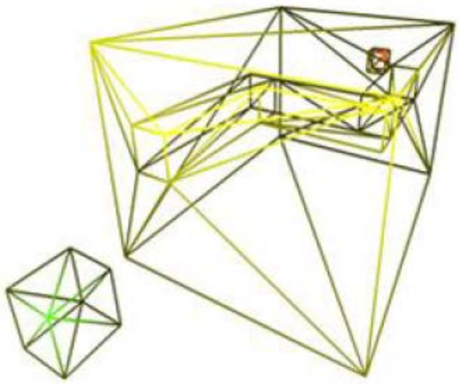


- Similar probabilistic manner to Bridge line-test

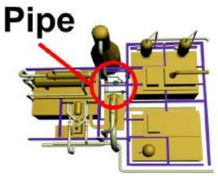
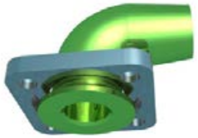
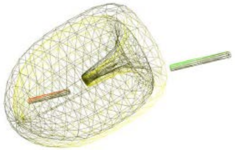
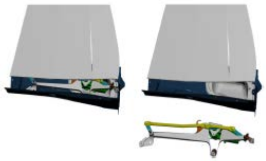
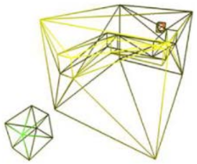
Results


Time (s)
100
80
60
40
20
0
RRT
RRRT
SR-RRT
S-tunnel 0.85


Time (s)
600
500
400
300
200
100
0
RRT
RRRT
SR-RRT
S-tunnel 1.3



Results: benchmarks

						
RRT	776.5	634.0	589.2	145.0	283.6	53.7
RRRT	140.9	225.8	33.1	39.7	141.5	98.8
Ours	**77.2**	**166.1**	**28.0**	**22.2**	**82.9**	**28.1**

On average, 6.7 times and 2 times improvement over RRT and RRRT

Summary

- **SR-RRT:**
  Selectively performs the retraction operations only near narrow passages
  - **Bridge line-test:**
    efficiently identifying narrow passage regions
    - **With high probability by performing PCA using local free space**
  - **Non-colliding line-test:**
    efficiently identifying wide-open free regions

- **Shows improvements on variety environments which have or do not have narrow passages**

Thank you