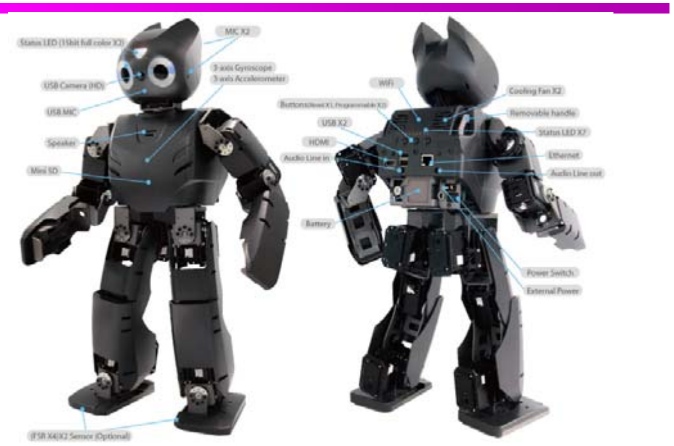
로봇경로생성 기술 및 응용
Robot Motion Planning and Applications

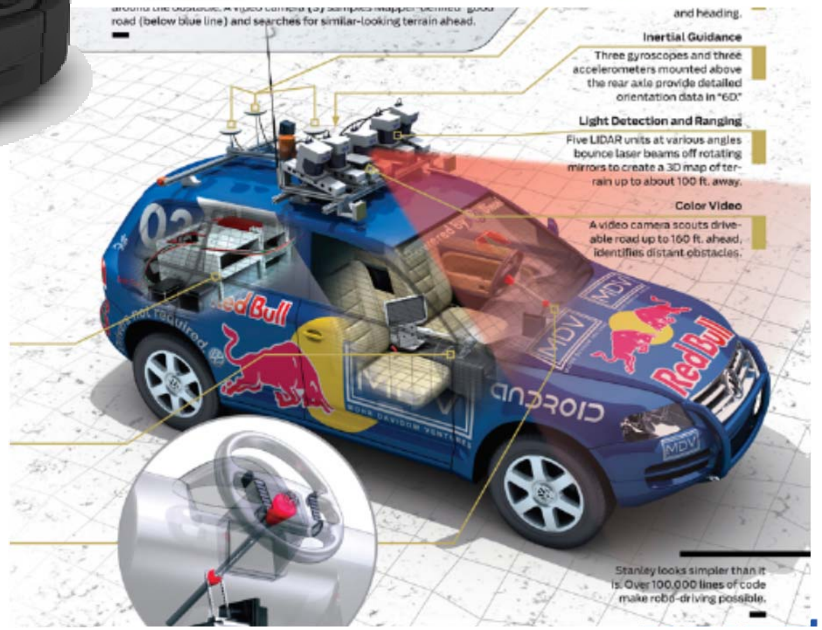
Sung-eui Yoon (윤성의)
Associate Professor
KAIST

http://sglab.kaist.ac.kr

Real World Robots



ASIMO



Da Vinci

Courtesy of Prof. Dinesh Manocha

Autonomous Robots

- Autonomous robots that sense, plan, and act in real and/or virtual worlds
- Algorithms and systems for representing, capturing, planning, controlling, and rendering motions of physical objects
- Applications:
  - Manufacturing
  - Mobile robots
  - Computational biology
  - Computer-assisted surgery
  - Digital actors

KAIST

Goal of Motion Planning

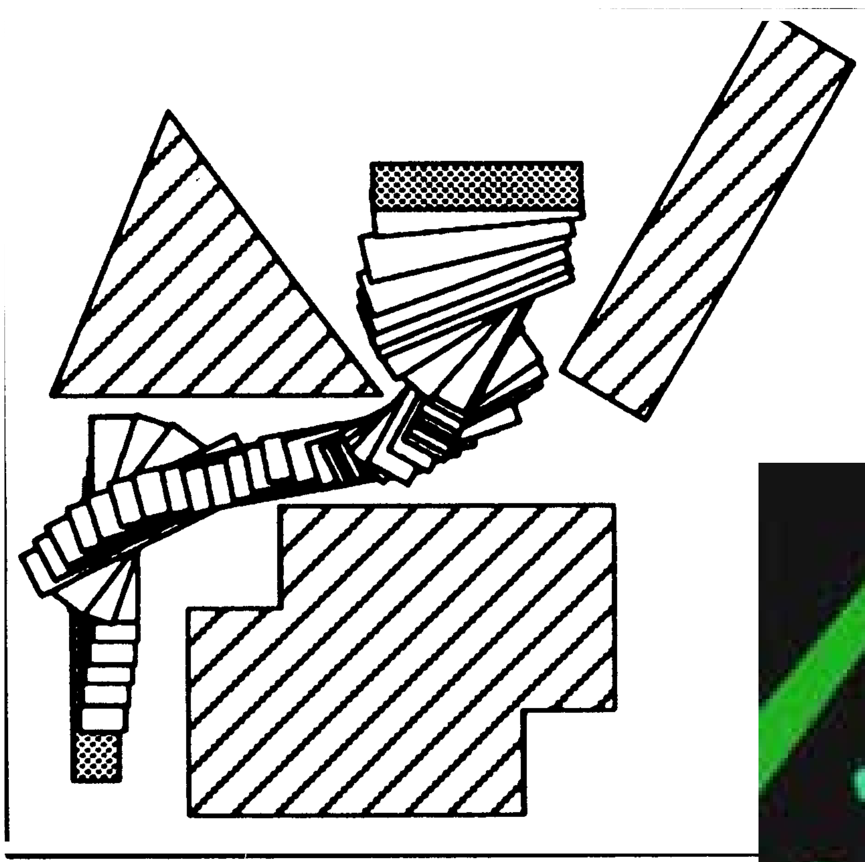
- Compute motion strategies, e.g.:
  - Geometric paths
  - Time-parameterized trajectories
  - Sequence of sensor-based motion commands
  - Aesthetic constraints

- Achieve high-level goals, e.g.:
  - Go to A without colliding with obstacles
  - Assemble product P
  - Build map of environment E
  - Find object O

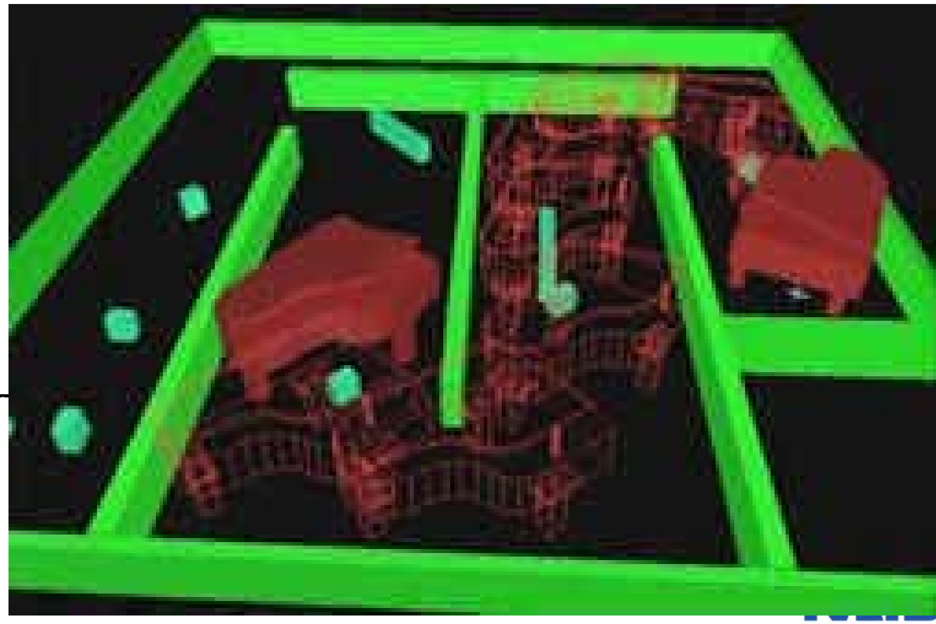
Basic Motion Planning Problem

- **Statement**:
  - Compute a collision-free path for an object (the robot) among obstacles subject to CONSTRAINTS
- **Inputs**:
  - Geometry of robot and obstacles
  - Kinematics of robot (degrees of freedom)
  - Initial and goal robot configurations (placements)
- **Outputs**:
  - Continuous sequence of collision-free robot configurations connecting the initial and goal configurations

Examples with Rigid Object

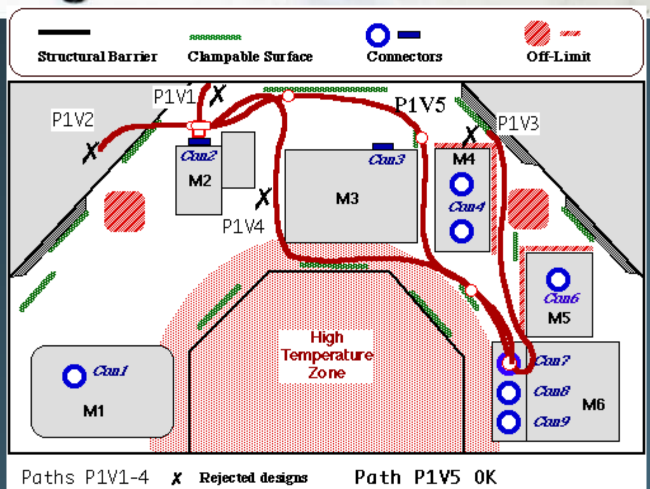


→ Ladder problem

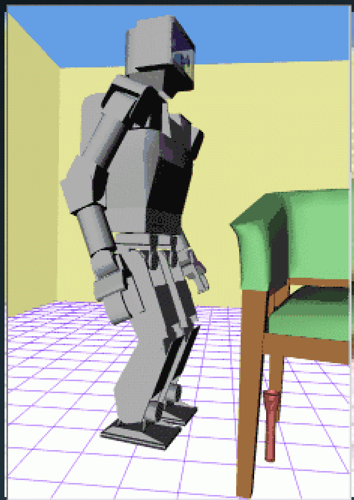


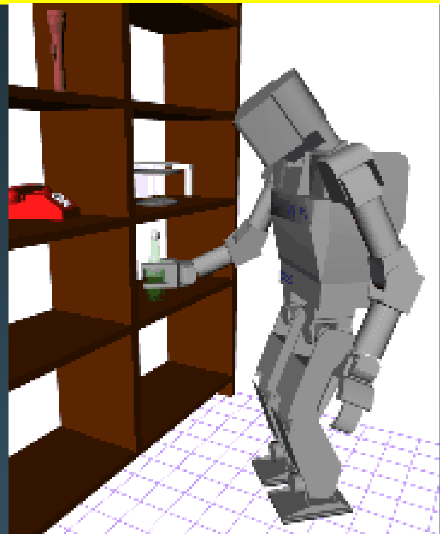
Piano-mover problem ←

Cable Harness/ Pipe design



Humanoid Robot





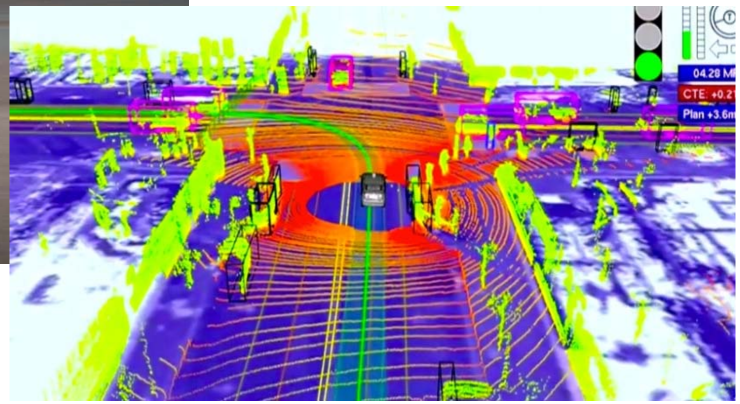



[Kuffner and Inoue, 2000] (U. Tokyo)

DARPA Grand Challenge

Planning for a collision-free 132 mile path in a desert

Google Self-Driving Vehicles



KAIST

Car is the next IT platform

WeeklyBiz

[Weekly BIZ] 실리콘밸리는 '자동차 밸리'… 세계 1~8위 車 회사 모두 몰렸다

팰로앨토·벨몬트(캘리포니아)=최원석 기자 ws

기사 | 100자평(0) | +크게 | -작게

입력 : 2013.08.31 03:05

왜 실리콘밸리로 가나
자동차는 갈수록 전자제품化, 첨단 소프트웨어 기술 확보 필요

러브콜 받는 한국 모바일 부품 업체
스마트폰과 연결 시키는 작업 중 실력 뛰어난 한국 업체와 연구 돌입

▲ 구글·애플 등 실리콘밸리의 터줏대감 IT 업체들 사이로 자동차회사 연구소들이 속속 모여들고 있다. ①스탠퍼드대가 있는 팰로앨토에 위치한 GM 연구소 ②벨몬트의 폴크스바겐 연구소 ③레드우드시티에 있는 전기차 업체 테슬라모터스의 전시장 ④실리콘밸리를 남북으로 관통하는 101 고속도로 위를 달리고 있는 구글 무인주행차. /실리콘밸리=최원석 기자

KAIST

Overview of This Tutorial

- **4:30pm: Optimal path planning and its applications**
  - 윤성의, 전산과, **KAIST**
- **5:10pm: Data-driven planning for multi-contact poses of human avatars**
  - 이성희, **CT, KAIST**
- **5:40pm: High performance geometric computation for robot motion planning**
  - 김영준, 컴퓨터공학과, 이대
- **6:10pm: Path Planning and Execution for Real Worlds**
  - 심현철 (이웅희), 항공우주, **KAIST**

Motion Planning Techniques

- Classical techniques
  - Roadmap, cell decomposition, potential fields
- Silhouette
  - First <u>complete</u> general method that applies to spaces of any dimension and is singly exponential in # of dimensions [Canny, 87]
  - Slow

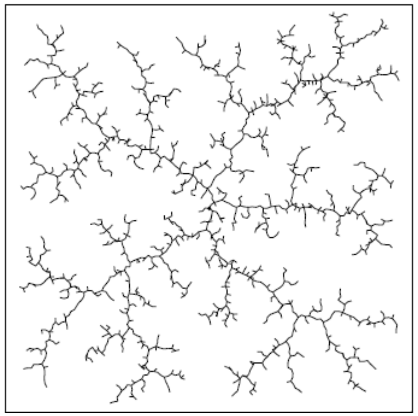
Motion Planning Techniques

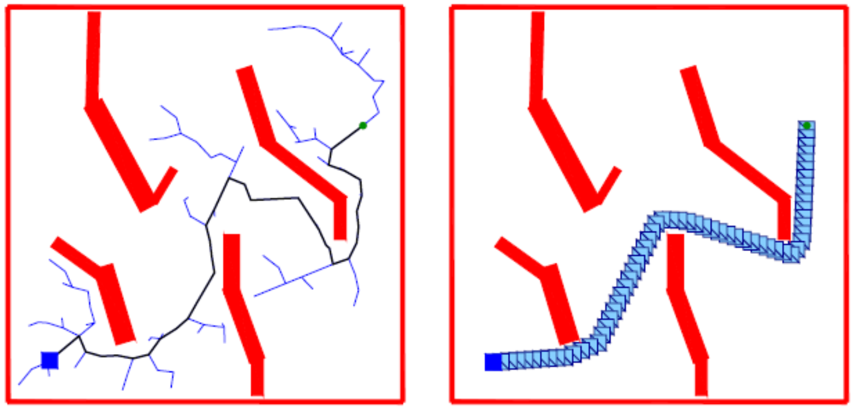
- Classical techniques
- Silhouette

- Sampling techniques w/ probabilistic completeness
  - Intuition: If there is a solution path, the algorithm will find it with a high probability
  - Probabilistic roadmaps
  - RRT techniques

Rapidly-exploring Random Trees (RRT) [LaValle 98]

- Present an efficient randomized path planning algorithm for single-query problems
  - Converges quickly
  - Probabilistically complete
  - Works well in high-dimensional C-space





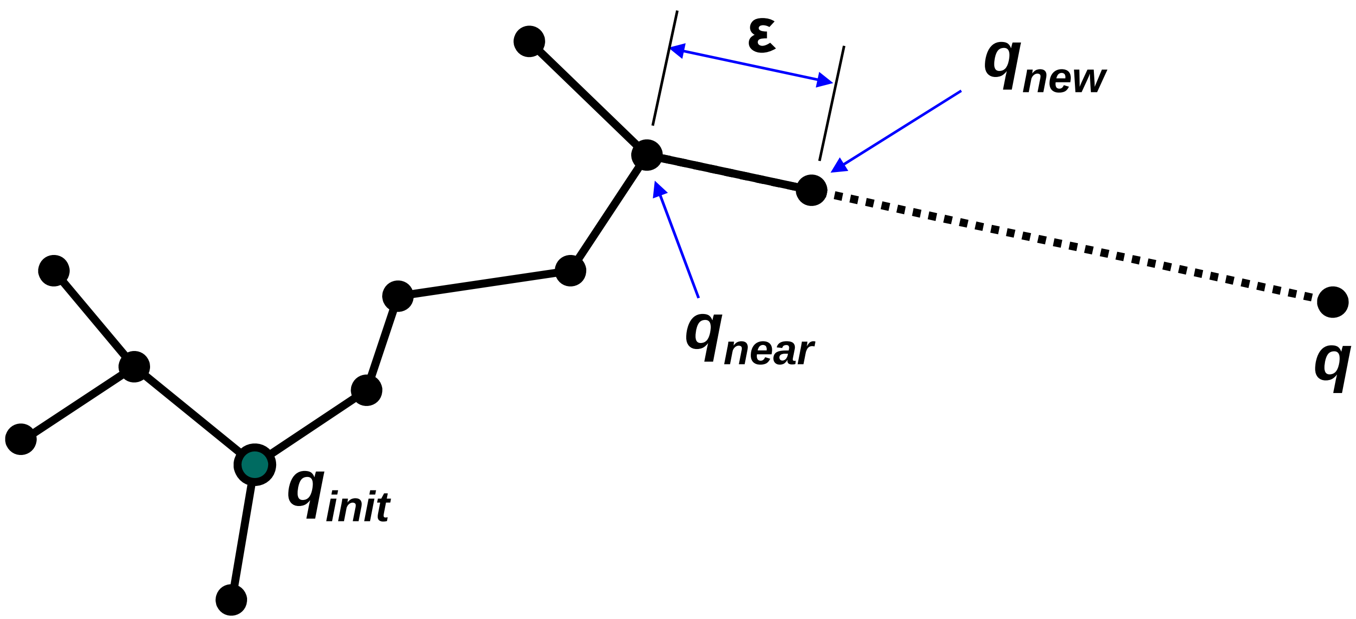
KAIST

Rapidly-Exploring Random Tree

- **A growing tree from an initial state**

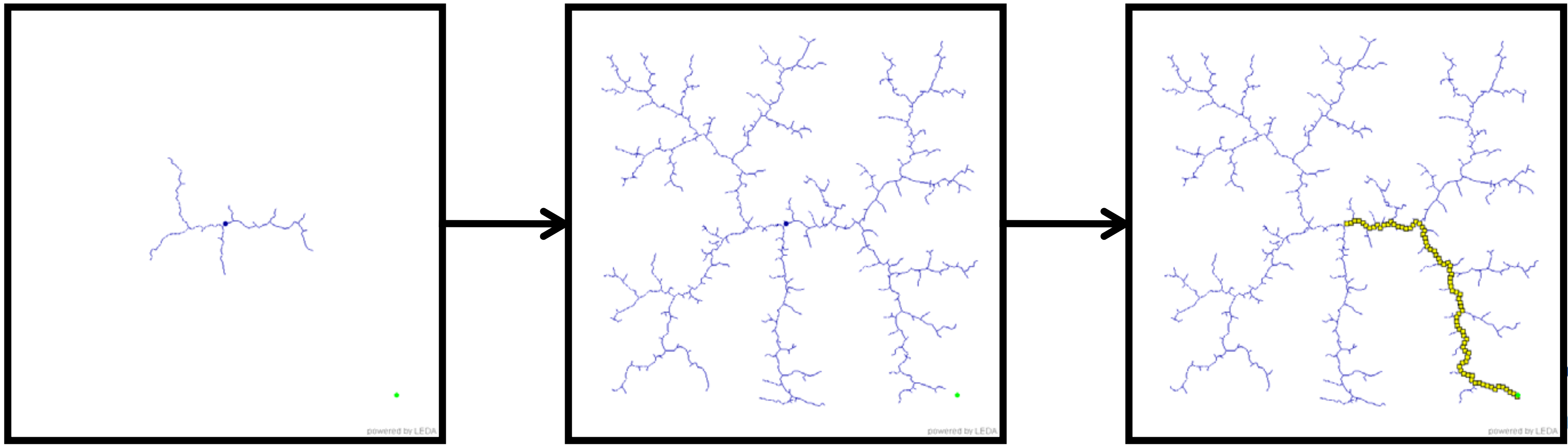
RRT Construction Algorithm

- Extend a new vertex in each iteration



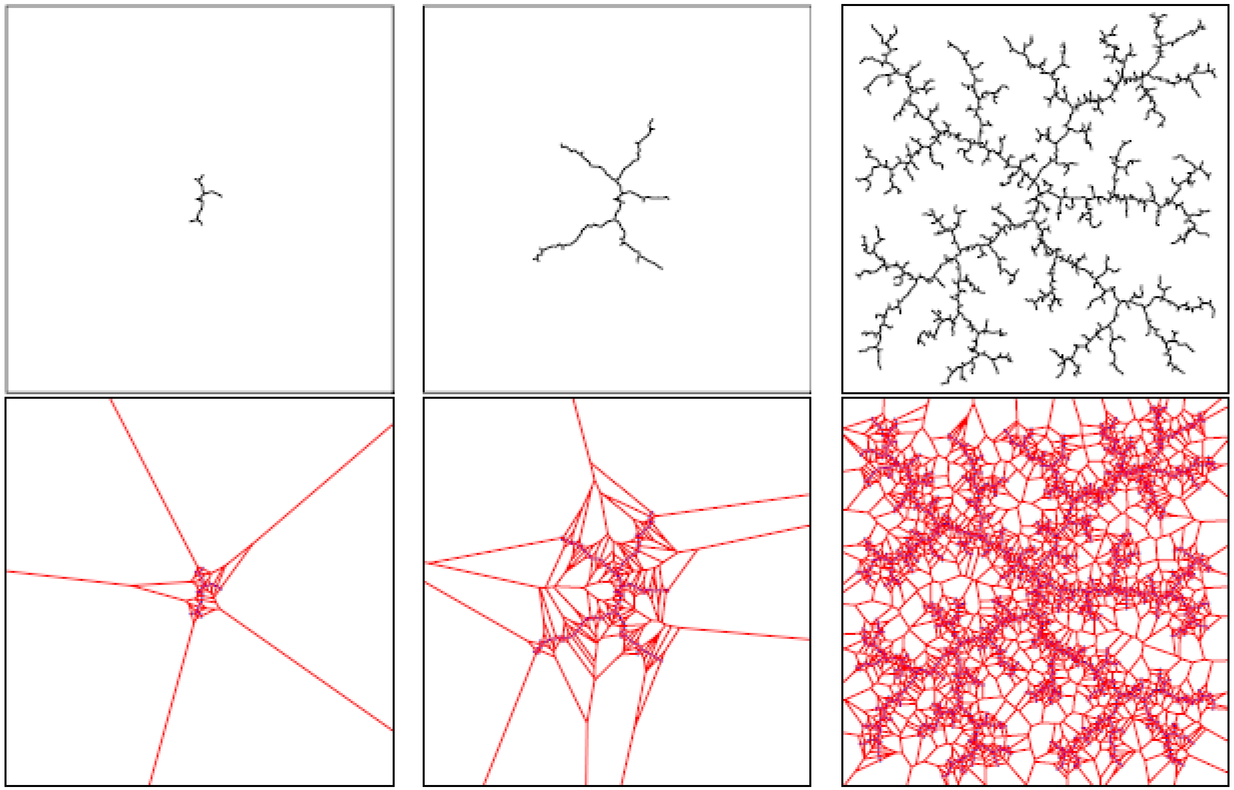
Overview – Planning with RRT

- Extend RRT until a nearest vertex is close enough to the goal state
  - Biased toward unexplored space
  - Can handle nonholonomic constraints and high degrees of freedom
- Probabilistically complete, but does not converge



Voronoi Region

- An RRT is biased by large Voronoi regions to rapidly explore, before uniformly covering the space



RRT Construction 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 $\mathcal{T}$

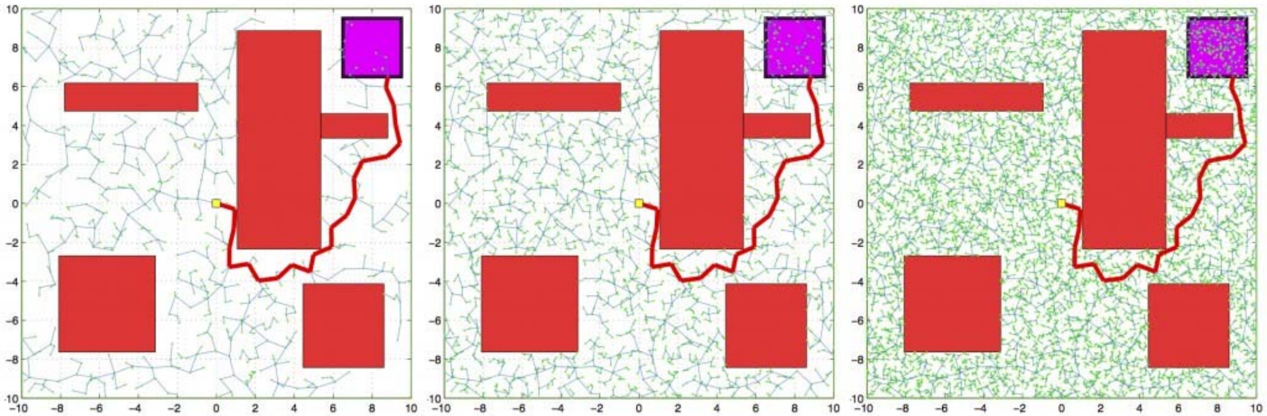
EXTEND($\mathcal{T}, q$)

$q_{near} \leftarrow$ 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*;

RRT*

- **RRT does not converge to the optimal solution**

RRT



(a) RRT in iteration 1,000 (b) RRT in iteration 3,000 (c) RRT in iteration 10,000

RRT*

RRT*

- **Asymptotically optimal without a substantial computational overhead**

Theorem [Karaman & Frazzoli, IJRR 2011]
(i) The RRT* algorithm is asymptotically optimal

$$\mathbb{P}\left(\left\{\lim_{n\to\infty} Y_n^{\mathrm{RRT}^*} = c^*\right\}\right) = 1$$

(ii) RRT* algorithm has no substantial computational overhead when compared to the RRT:

$$\lim_{n\to\infty} \mathbb{E}\left[\frac{M_n^{\mathrm{RRT}^*}}{M_n^{\mathrm{RRT}}}\right] = \text{constant}$$

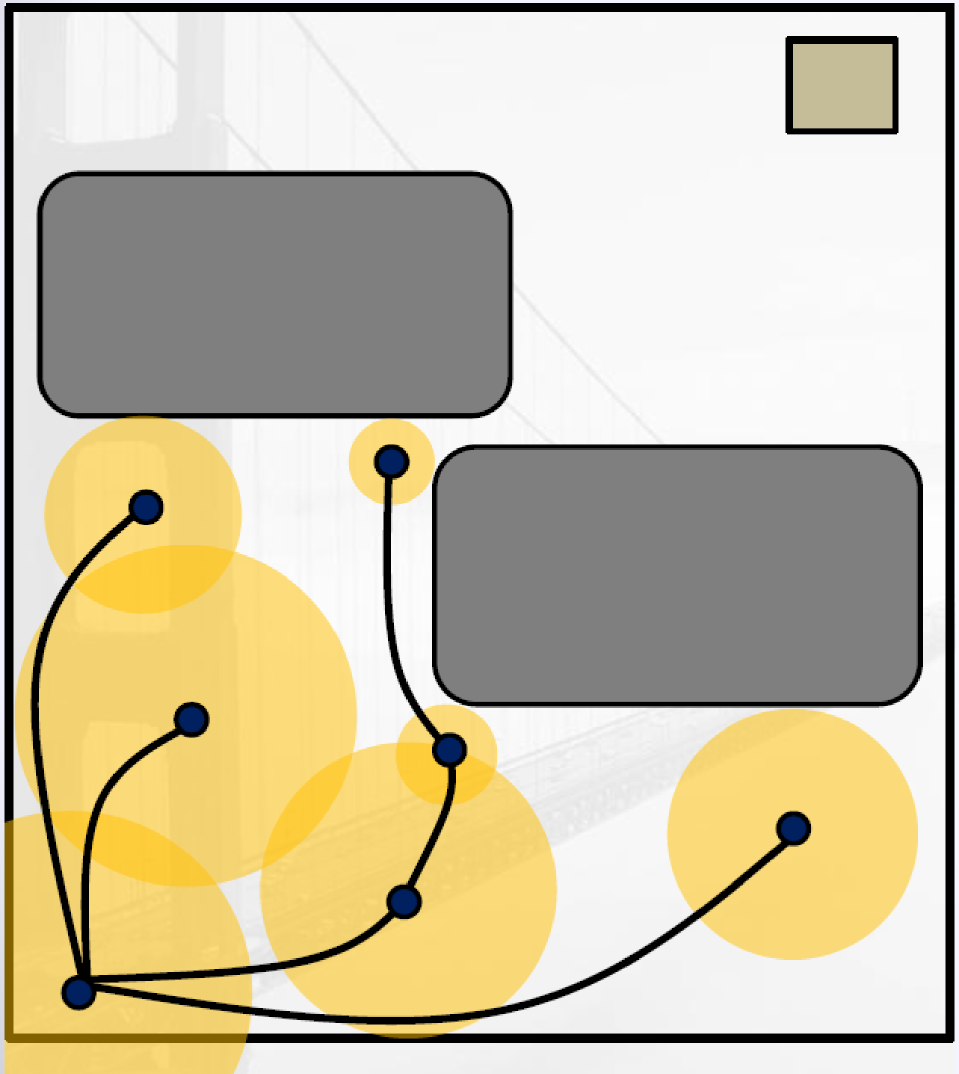
- **Y_n^{RRT*} : cost of the best path in the RRT***
- **c^* : cost of an optimal solution**
- **M_n^{RRT} : # of steps executed by RRT at iteration n**
- **M_n^{RRT*}: # of steps executed by RRT* at iteration n**

From DH's homepage

Key Operation of RRT*

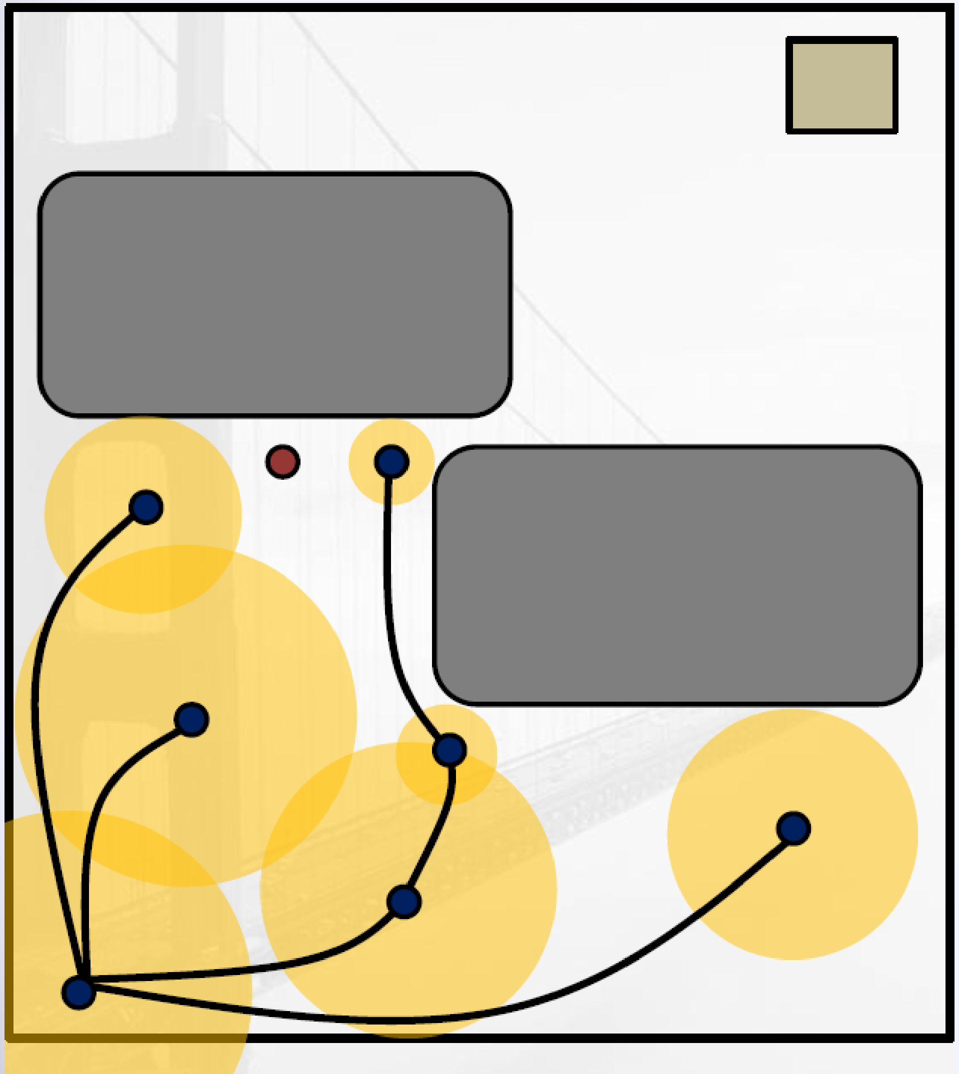
- RRT
  - Just connect a new node to its nearest neighbor node
- RRT*: refine the connection with re-wiring operation
  - Given a ball, identify neighbor nodes to the new node
  - Refine the connection to have a lower cost

Example: Re-Wiring Operation



From ball tree paper

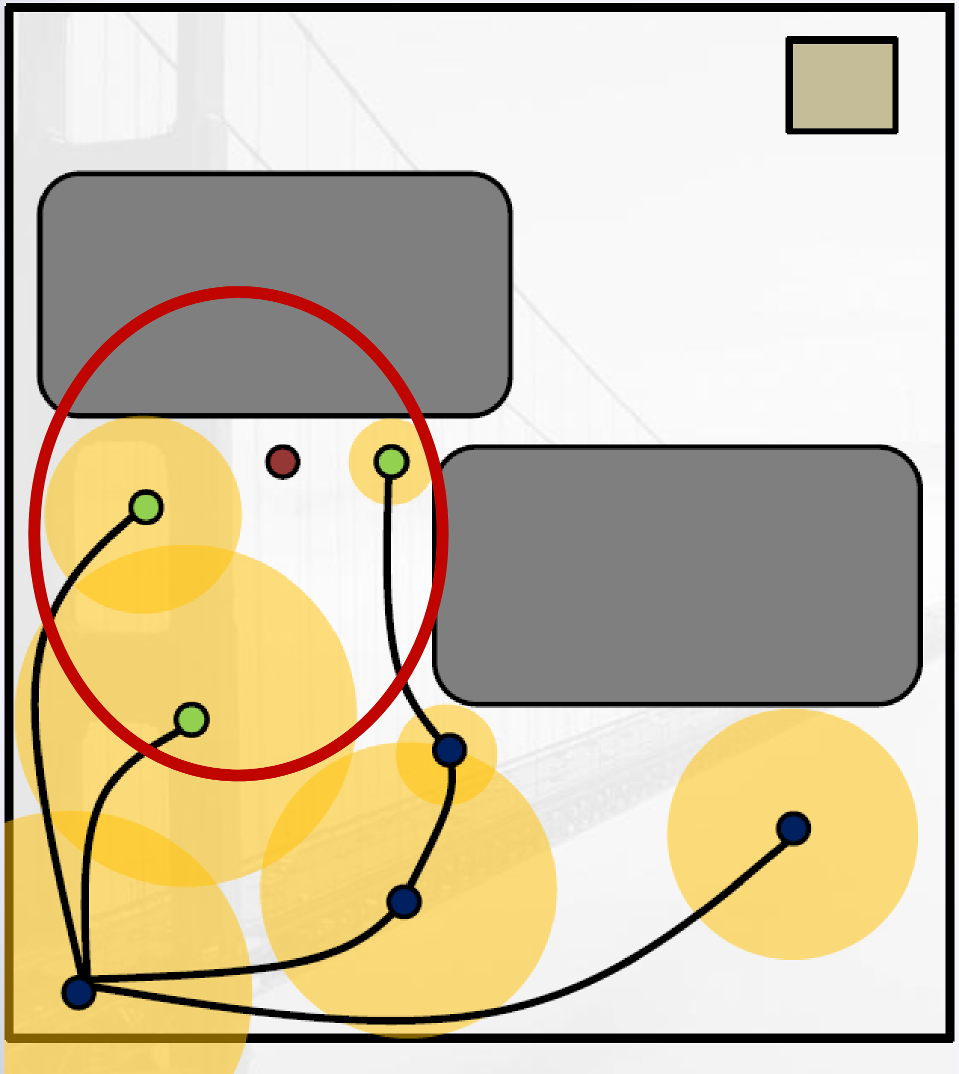
Example: Re-Wiring Operation



Generate a new sample

From ball tree paper

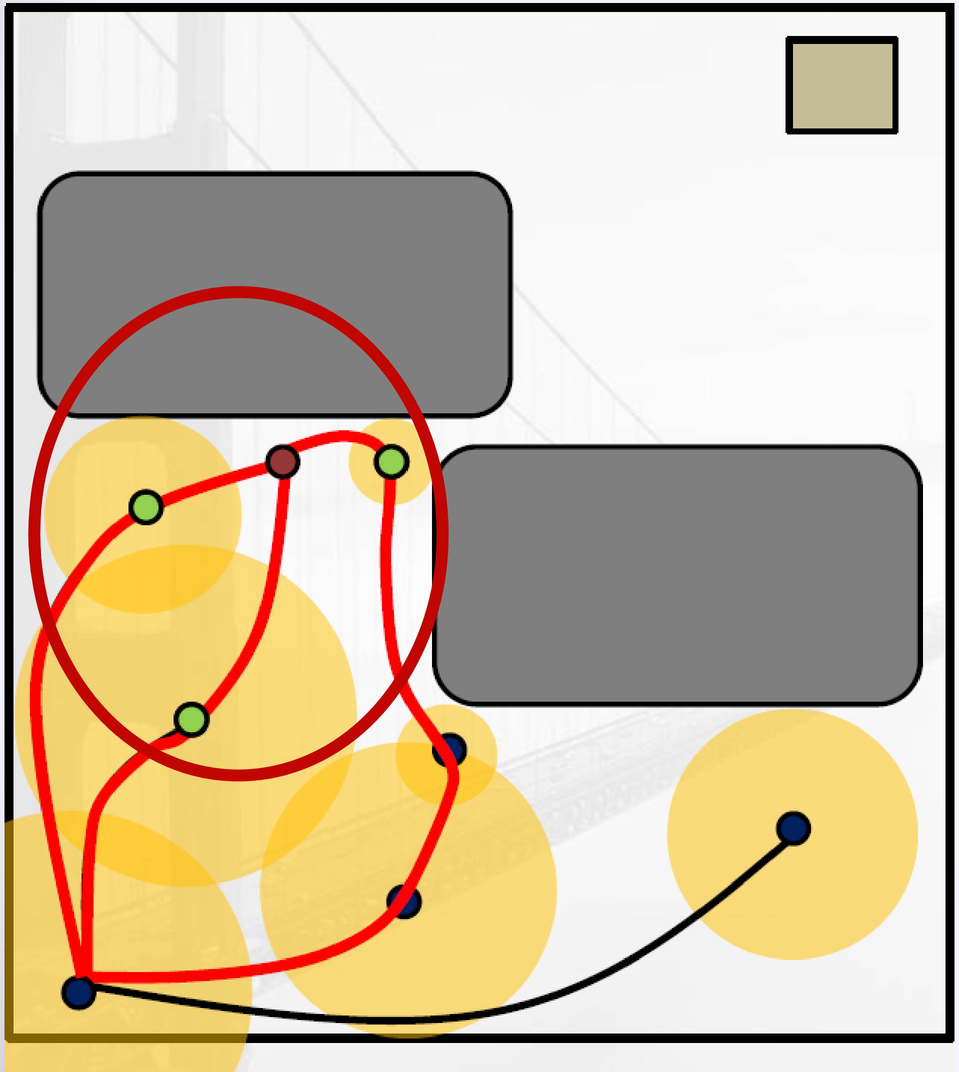
Example: Re-Wiring Operation



Identify nodes in a ball

From ball tree paper

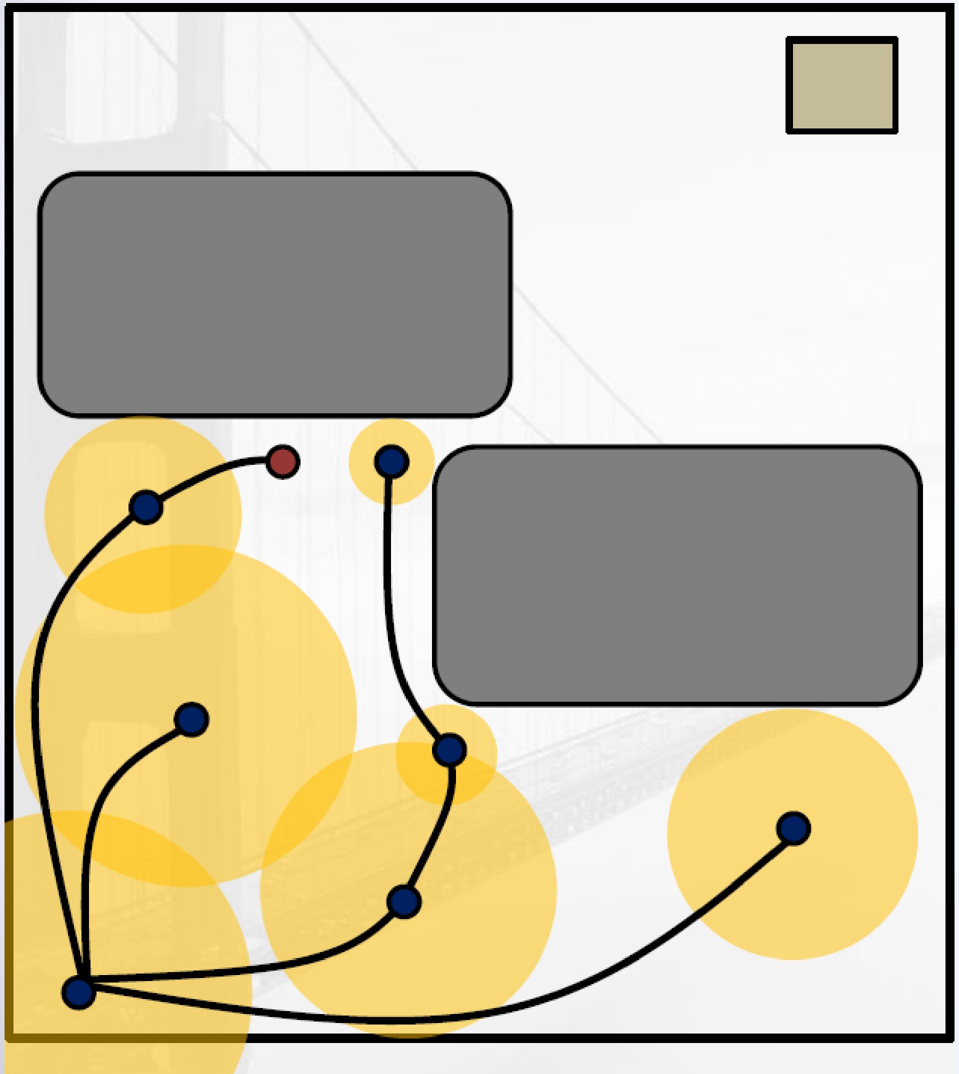
Example: Re-Wiring Operation



Identify which parent gives the lowest cost

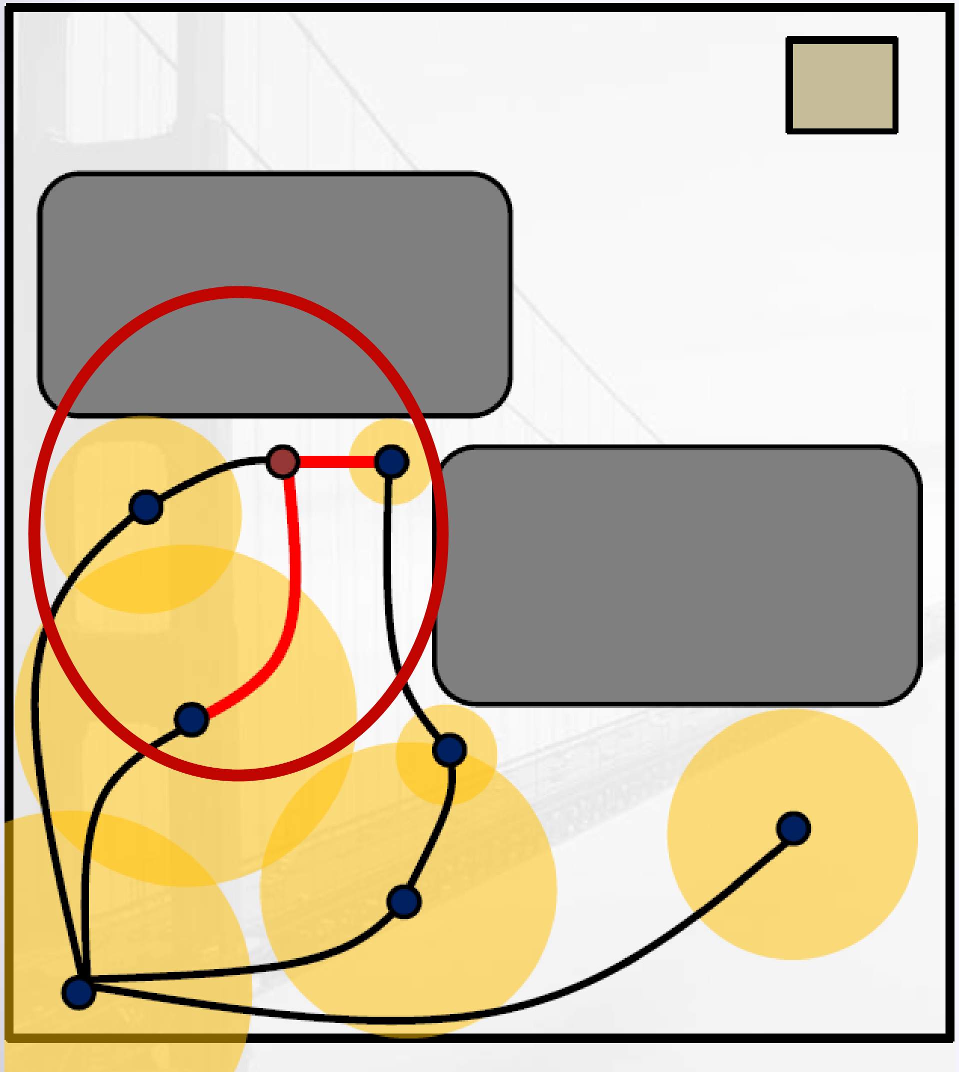
From ball tree paper

Example: Re-Wiring Operation



From ball tree paper

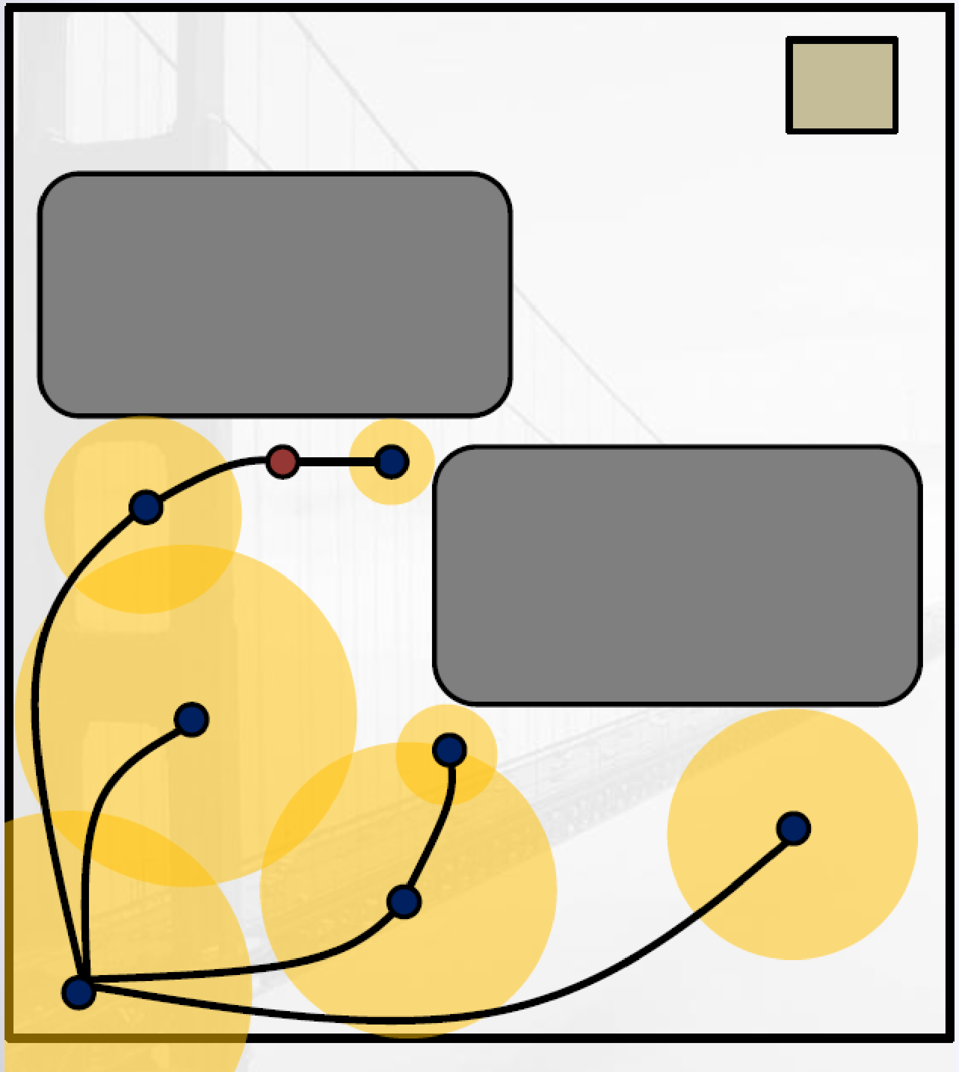
Example: Re-Wiring Operation



Identify which child gives the lowest cost

From ball tree paper

Example: Re-Wiring Operation



Video showing benefits with real robot

From ball tree paper

Two Recent Works of Our Group

- Handling narrow passages
- Improving low convergence to the optimal solution

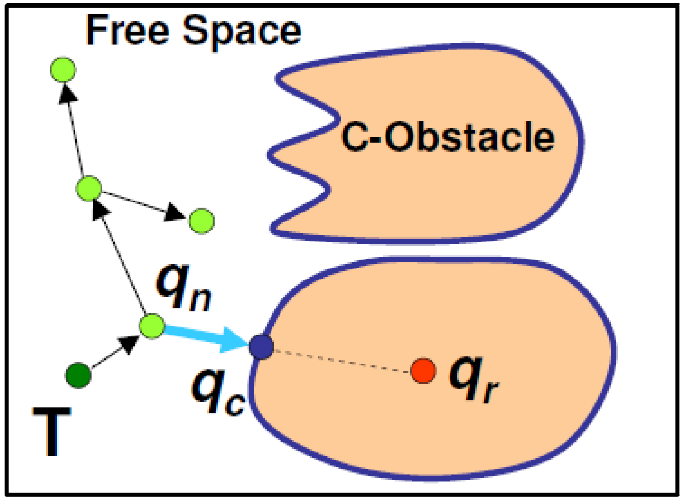
Two Recent Works of Our Group

- **Narrow passages**
  - **Identify narrow passage with a simple one-dimensional line test, and selectively explore such regions**
  - **Selective retraction-based RRT planner for various environments, Lee et al., T-RO 14**
  - **http://sglab.kaist.ac.kr/SRRRT/T-RO.html**

- **Low convergence to the optimal solution**

Retration-based RRT [Zhang & Manocha 08]

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



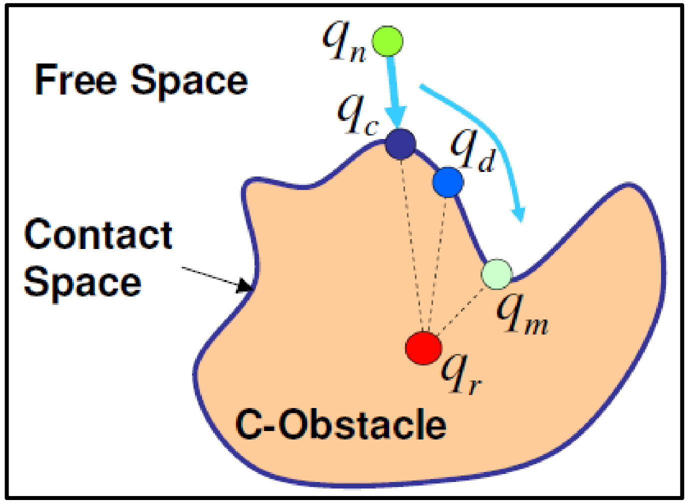
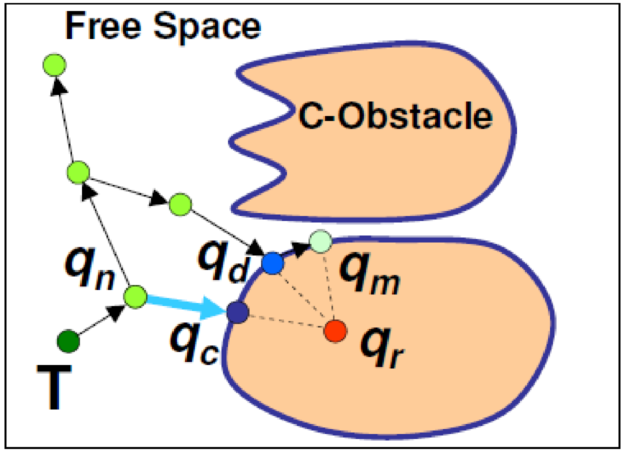


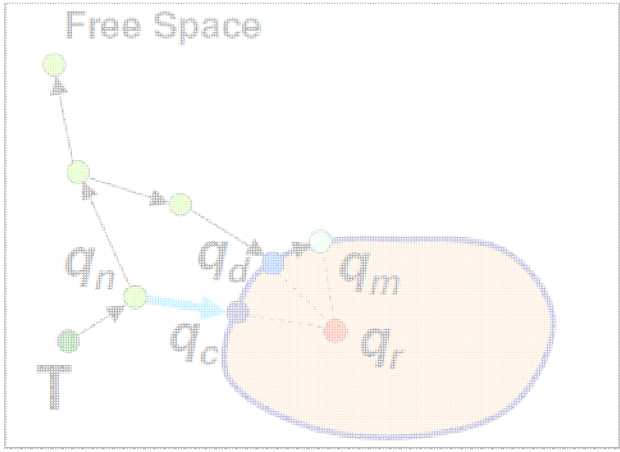
image from [Zhang & Manocha 08]

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

RRRT: Pros and Cons

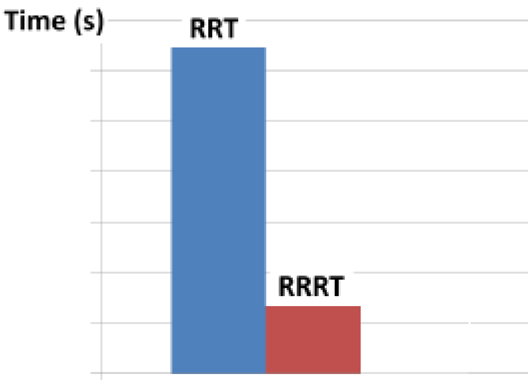


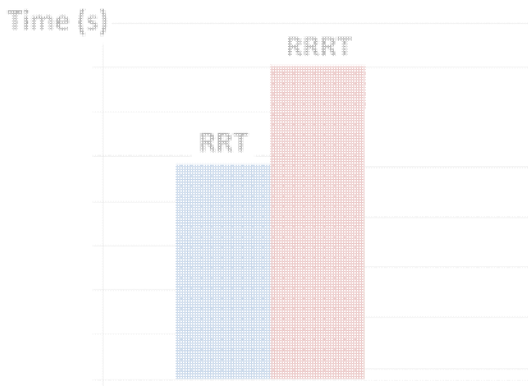
with narrow passages



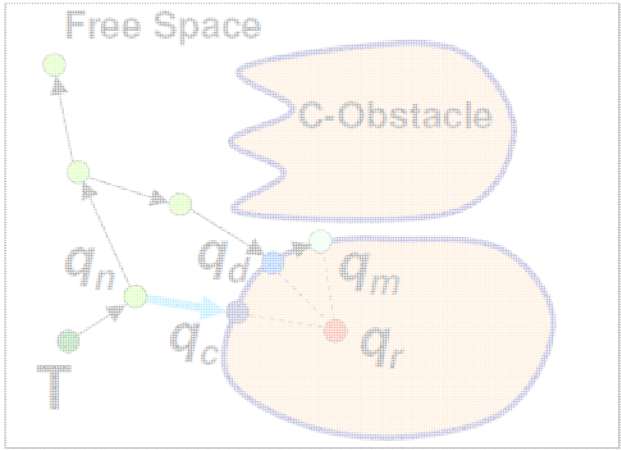
without narrow passages

images from [Zhang & Manocha 08]

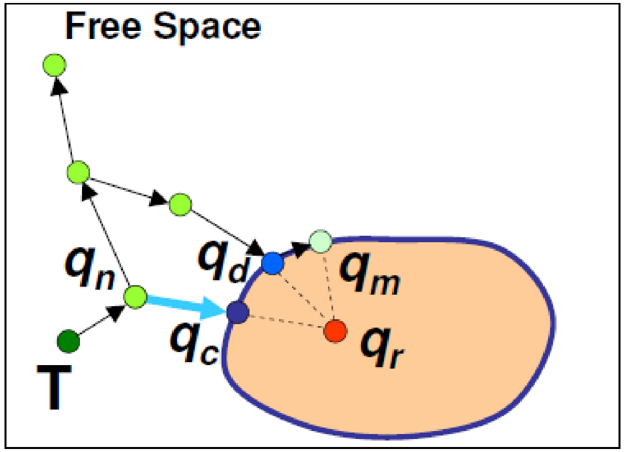




RRRT: Pros and Cons

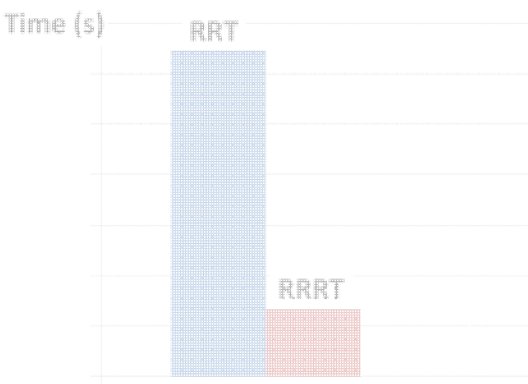


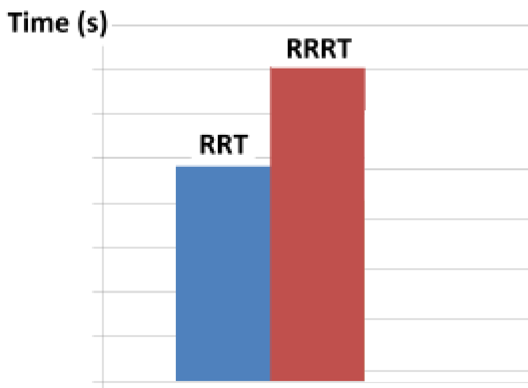
with narrow passages



without narrow passages

images from [Zhang & Manocha 08]

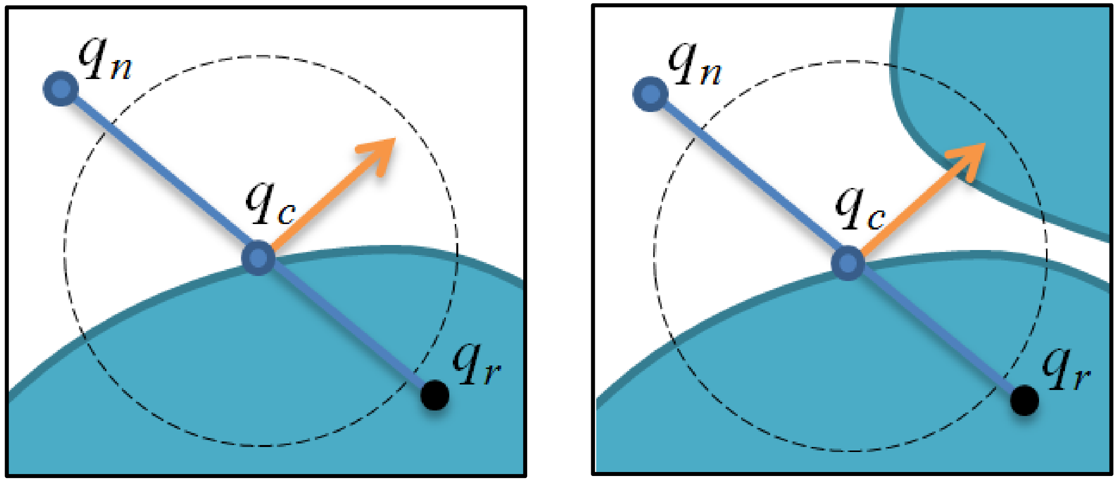




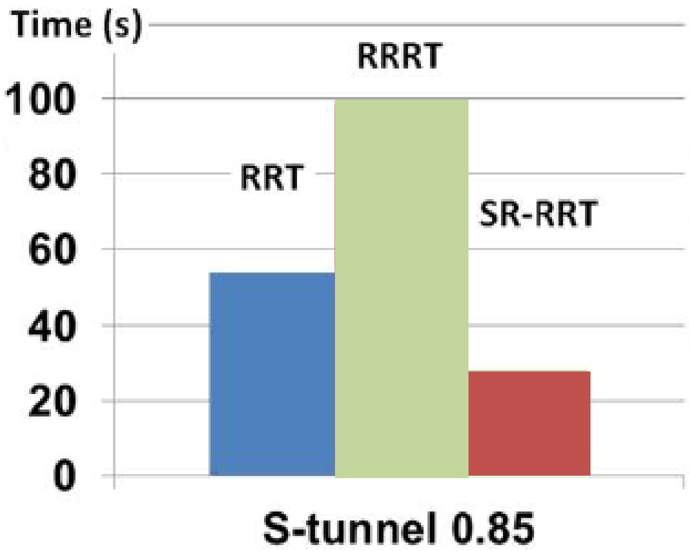
Bridge line-test [Lee et al., T-RO 14]

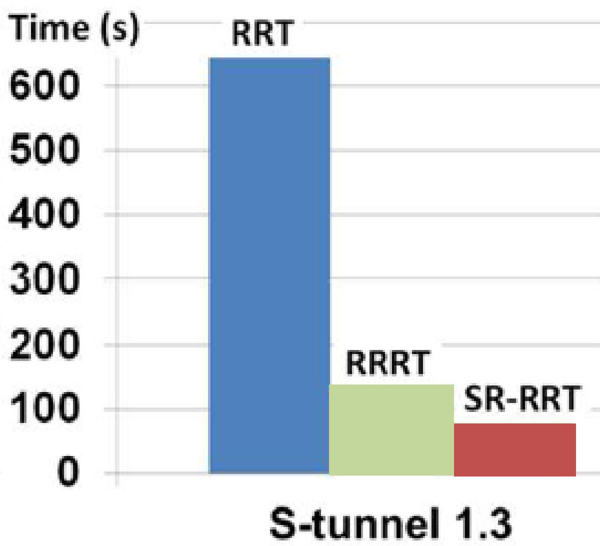
- To identify narrow passage regions

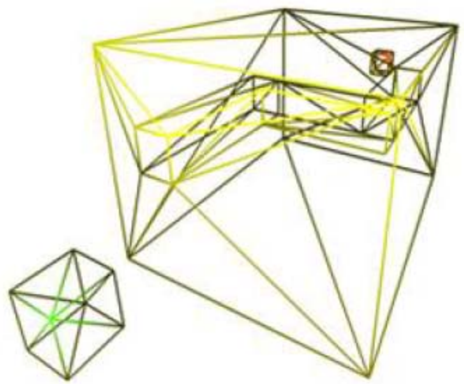
- Bridge line-test
  1. Generate a random line
  2. Check whether the line meets any obstacle



Results







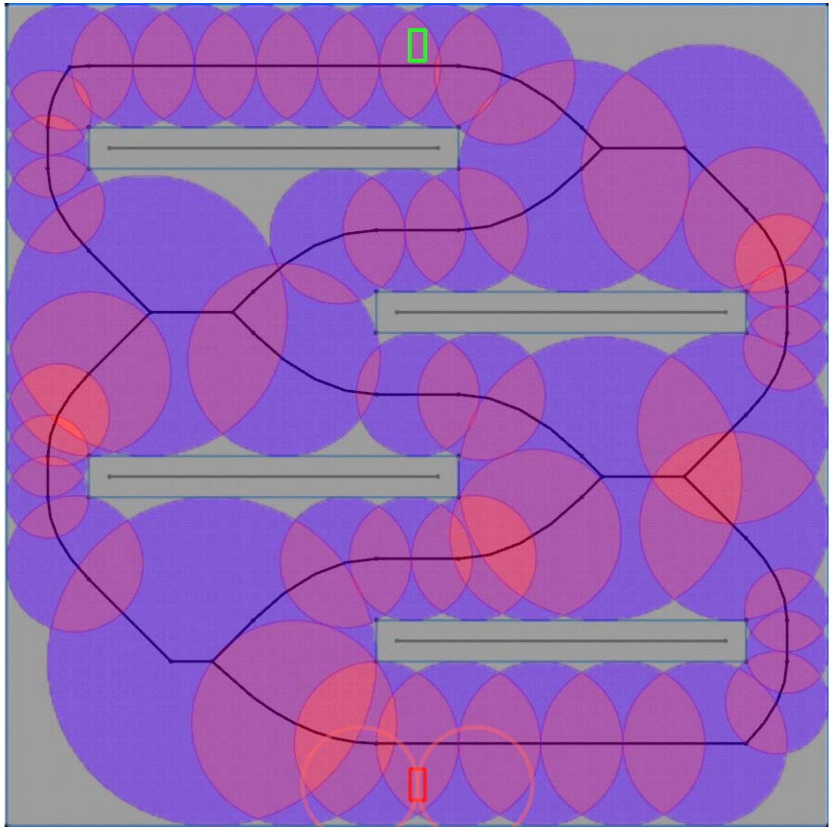
Video

Two Recent Works of Our Group

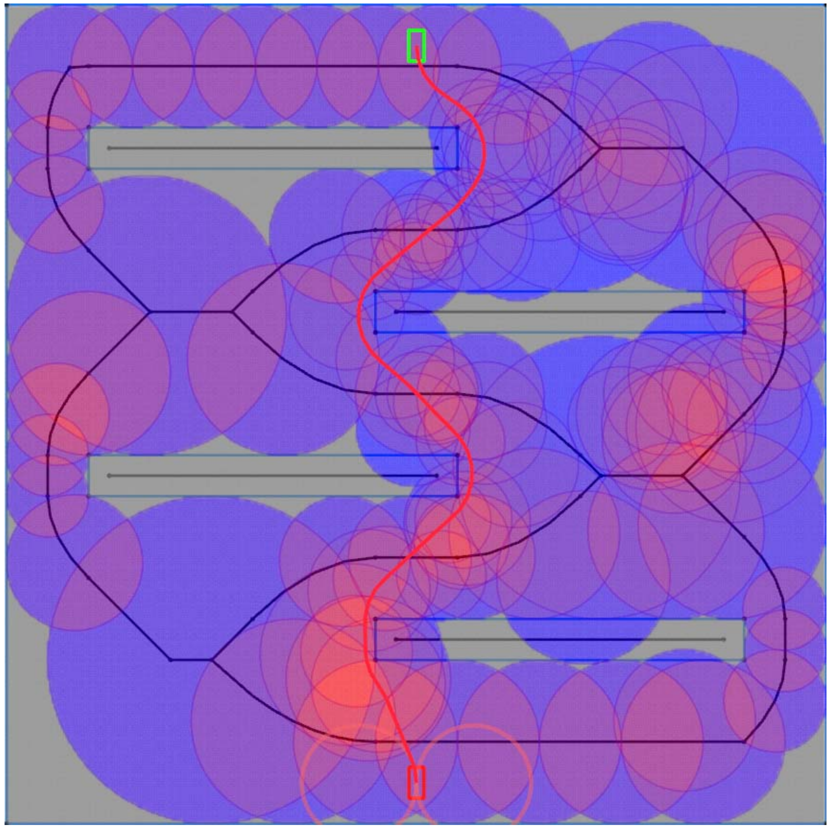
- **Handling narrow passages**

- **Improving low convergence to the optimal solution**
  - **Use the sampling cloud to indicate regions that lead to the optimal path**
  - **Cloud RRT* : Sampling Cloud based RRT*, Kim et al., ICRA 14**
  - **http://sglab.kaist.ac.kr/CloudRRT/**

Examples of Sampling Cloud [Kim et al., ICRA 14]



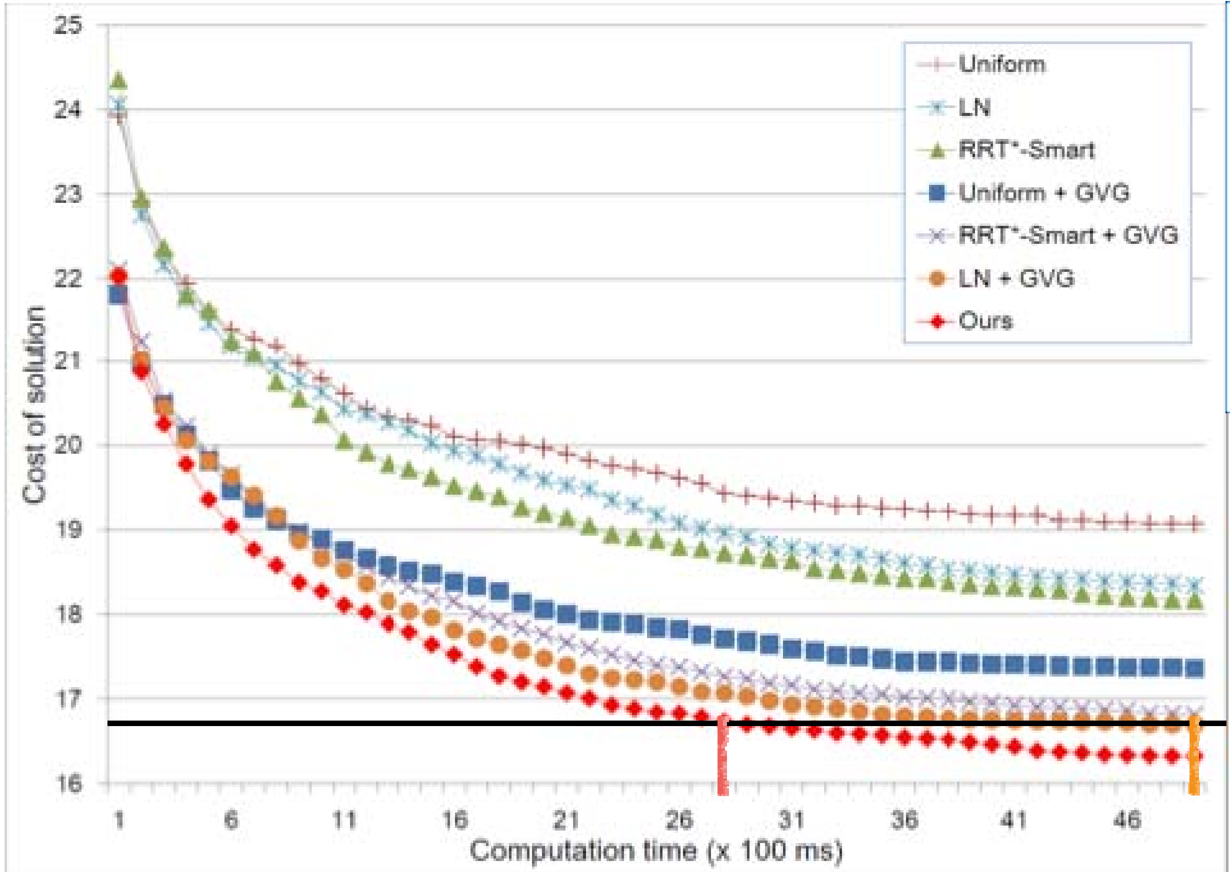
Initial state of sampling cloud

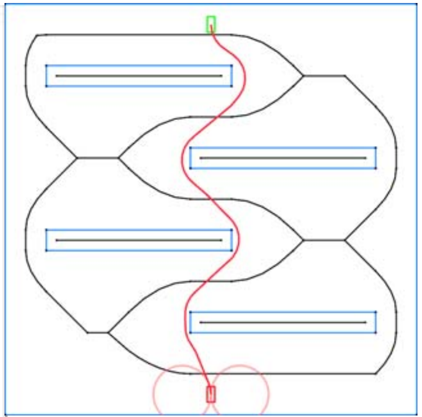


After updated several times

Video

Results: 4 squares





1.8X improvement

Conclusions

- Explained the basic motion planning problem and its goal
- Covered basic sampling based planners
  - RRT
- Discussed an optimal RRT: RRT*
- Briefly talked about two recent works
  - Handling narrow passages and low convergence

http://sglab.kaist.ac.kr

Acknowledgements

- Collaborators
  - My students, M. Gopi, Miguel Otaduy, George Drettakis, SeungYoung Lee, YuWing Tai, John Kim, Dinesh Manocha, Peter Lindstrom, Yong Joon Lee, Pierre-Yves Laffont, Jeong Mo Hong, Sun Xin, Nathan Carr, Zhe Lin
- Funding sources
  - Korea Research Foundation
  - Ministry of Knowledge Economy
  - Samsung, Microsoft Research Asia, Adobe, Boeing

KAIST