CS686: RRT

Sung-Eui Yoon
(윤성의)

Course URL:
http://sgvr.kaist.ac.kr/~sungeui/MPA

KAIST

Class Objectives

- **Understand the RRT technique and its recent advancements**
  - **RRT***
  - **Kinodynamic planning**

- **Last time**
  - **Probabilistic roadmap techniques**
  - **Sampling and re-sampling techniques**

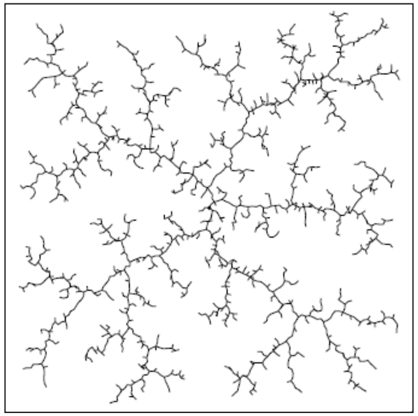
Question

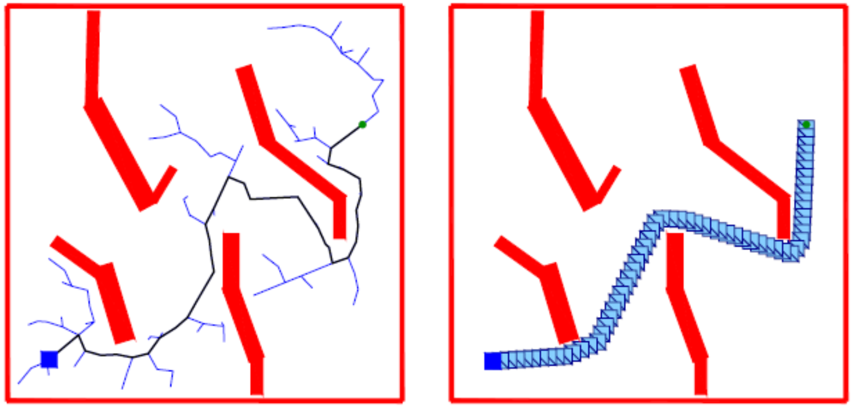
- **PRM assumes that we know the global map, but how can we handle the case where we have only a partial map due to the limited sensor range?**
  - **지난시간에 배운 PRM 기법들은 글로벌 맵을 알고 있어야 문제 해결이 가능한데, 전체 맵의 일부분(센서 탐지거리 제약 등으로)만을 알고 있는 상황에서 PRM알고리즘을 적용하려면 어떤 방식으로 해야 하는지요?**

KAIST

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**



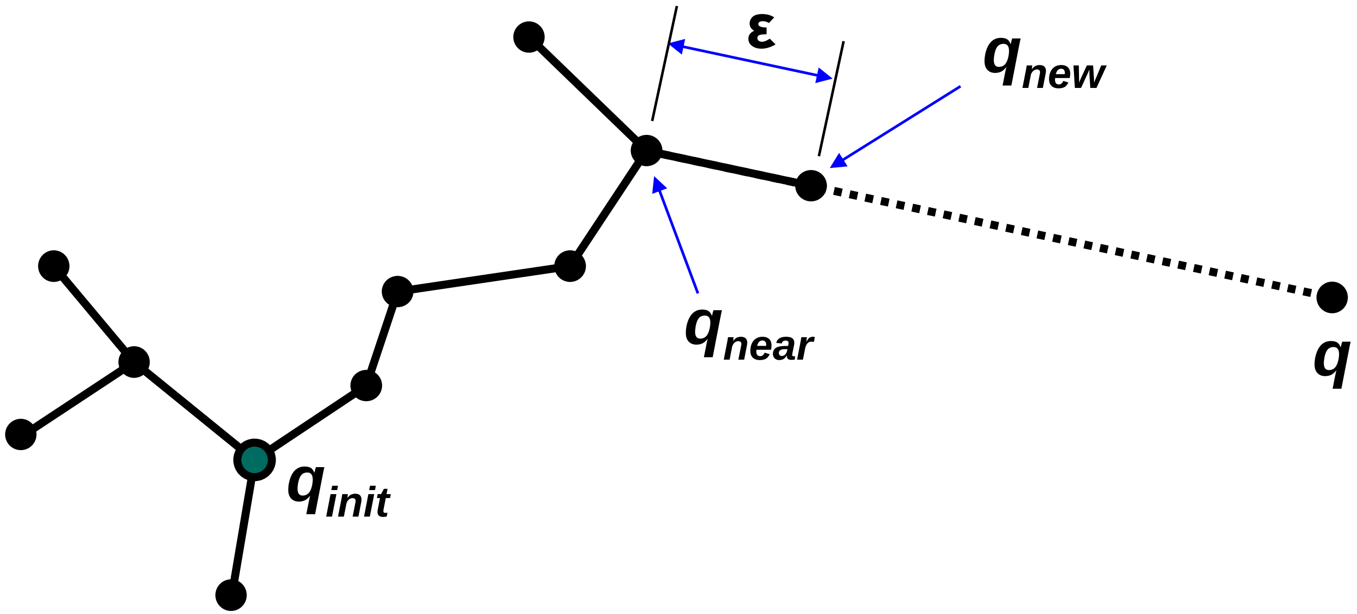


Rapidly-Exploring Random Tree

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

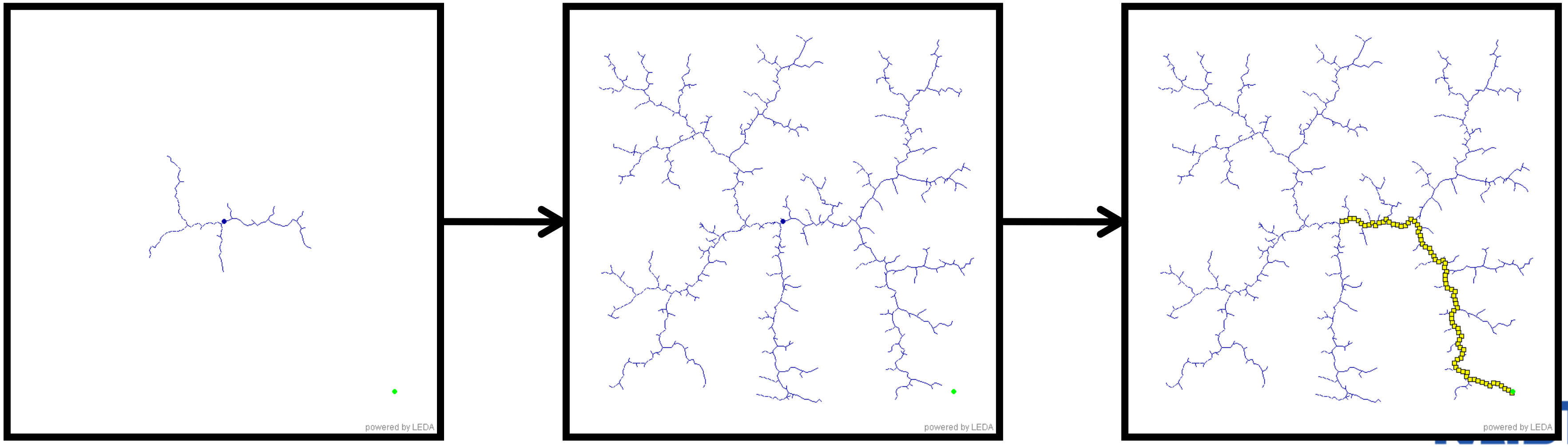
RRT Construction Algorithm

- **Extend a new vertex in each iteration**
  - **Alternatively, one can simply connect**



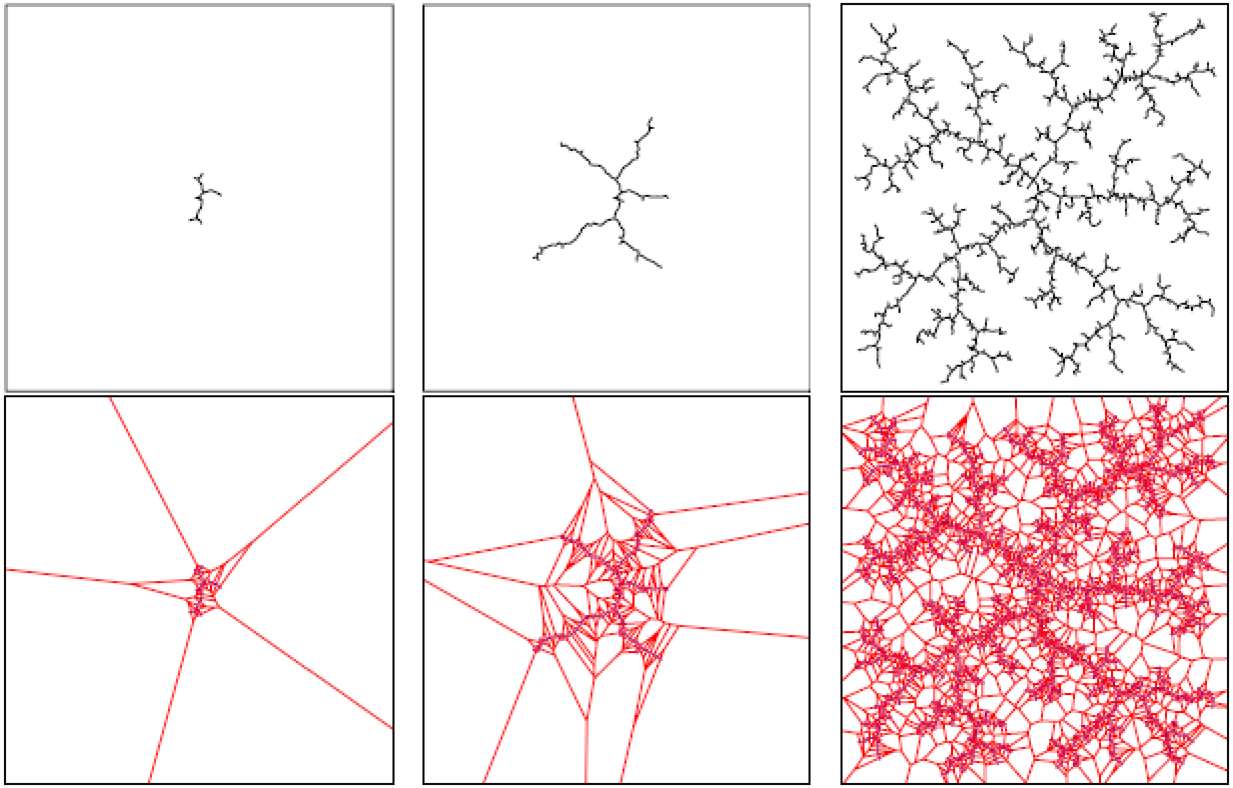
Overview – Planning with RRT

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



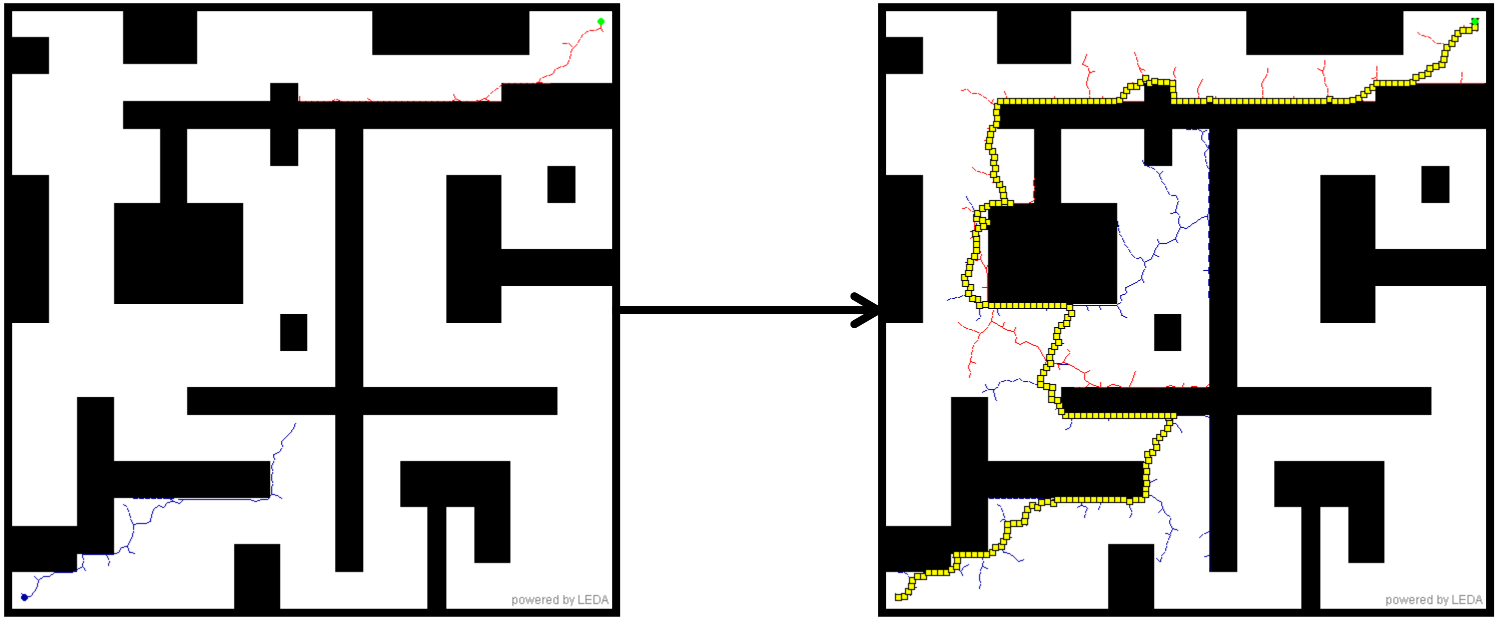
Voronoi Region

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



Overview – With Dual RRT

- Extend RRTs from both initial and goal states
- Find path much more quickly

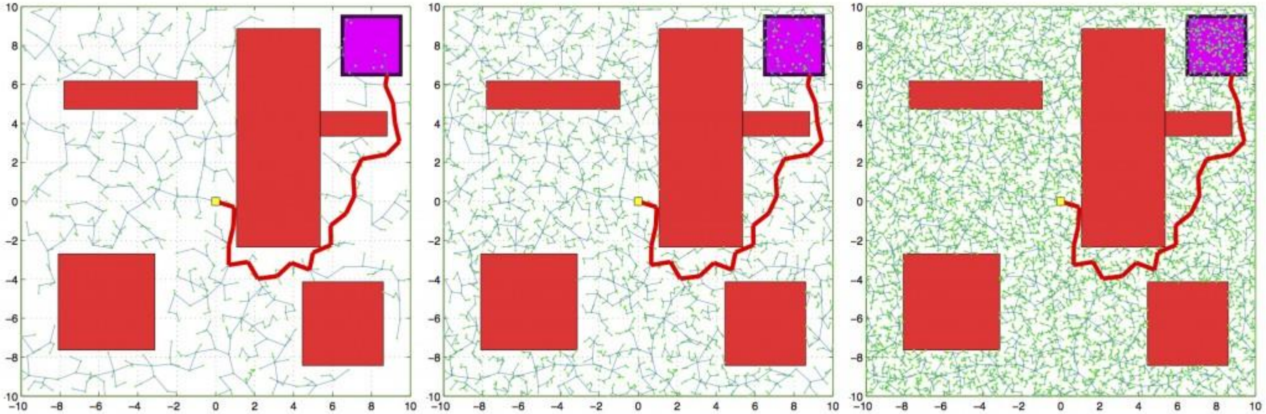


737 nodes are used

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*

From Sertac's homepage

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:

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

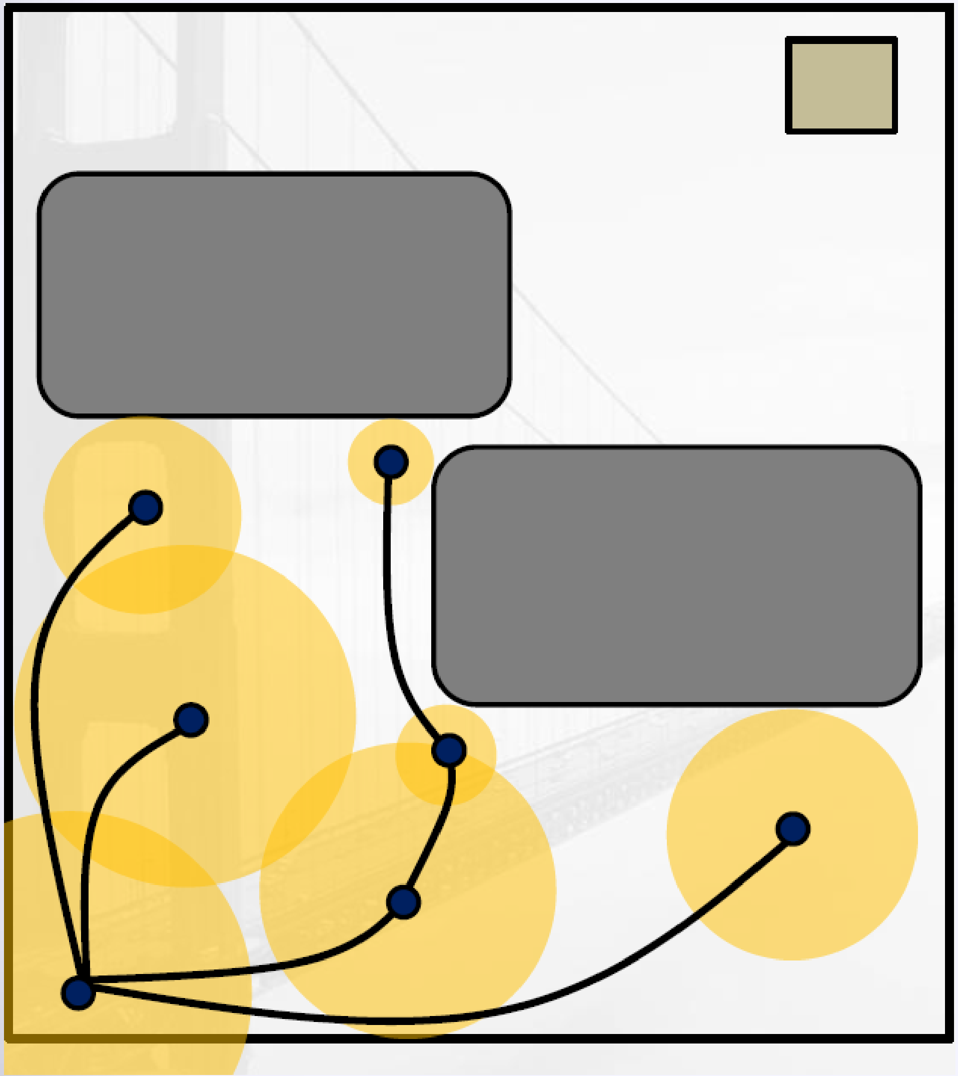
- **Y_n^{RRT*} : cost of the best path in the RRT***
- **c^* : cost of an optimal solution**
- **M_n^{RRT} : # of steps executed by RRT at iteration n**
- **M_n^{RRT*}: # of steps executed by RRT* at iteration n**

From DH's homepage

Key Operation of RRT*

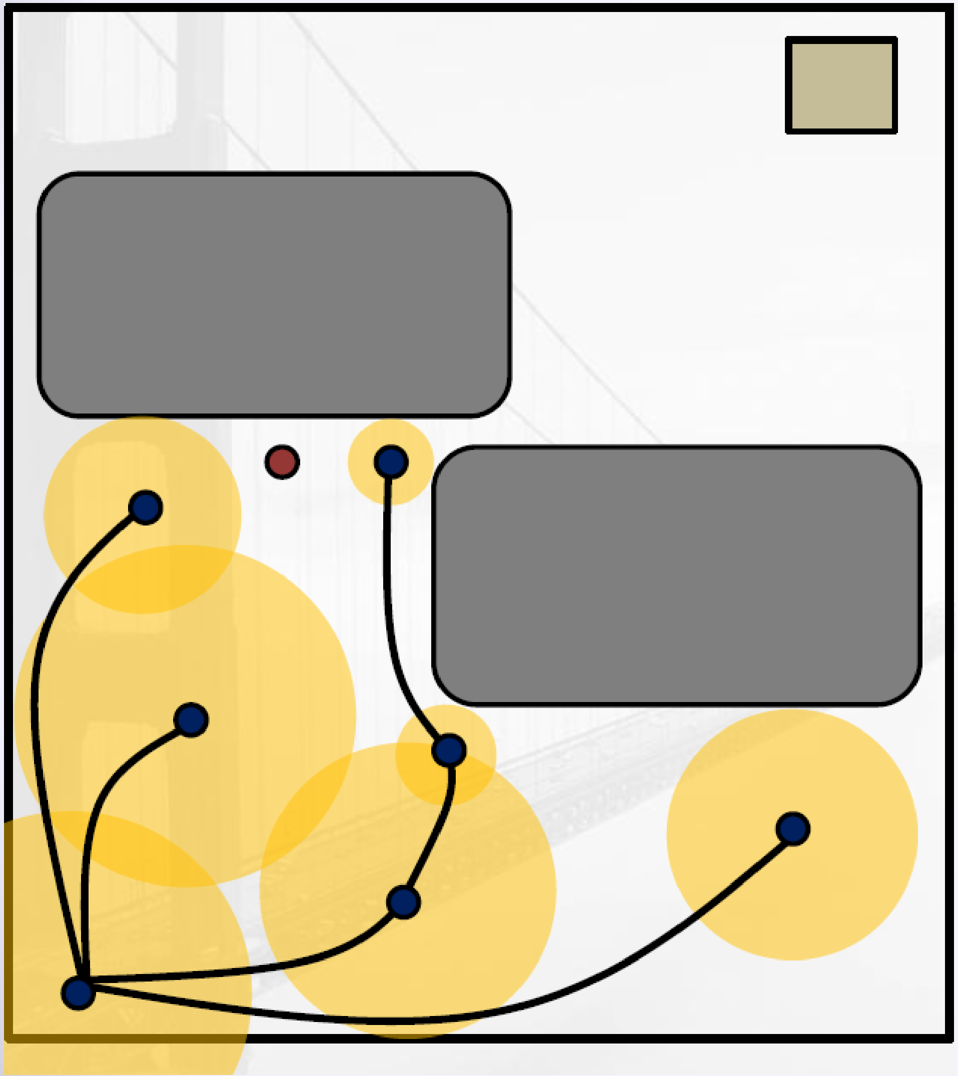
- RRT
  - Just connect a new node to its nearest neighbor node
- RRT*: refine the connection with re-wiring operation
  - Given a ball, identify neighbor nodes to the new node
  - Refine the connection to have a lower cost

Example: Re-Wiring Operation



From ball tree paper

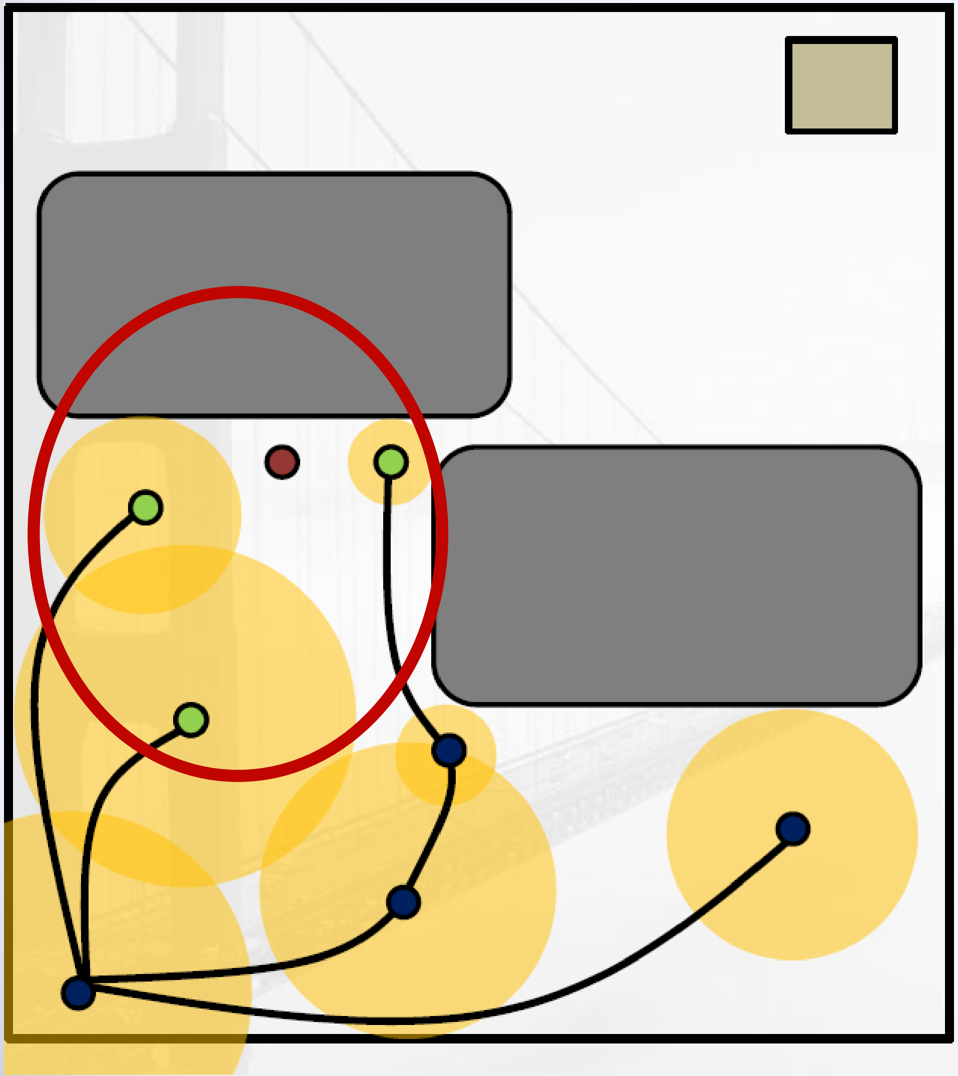
Example: Re-Wiring Operation



Generate a new sample

From ball tree paper

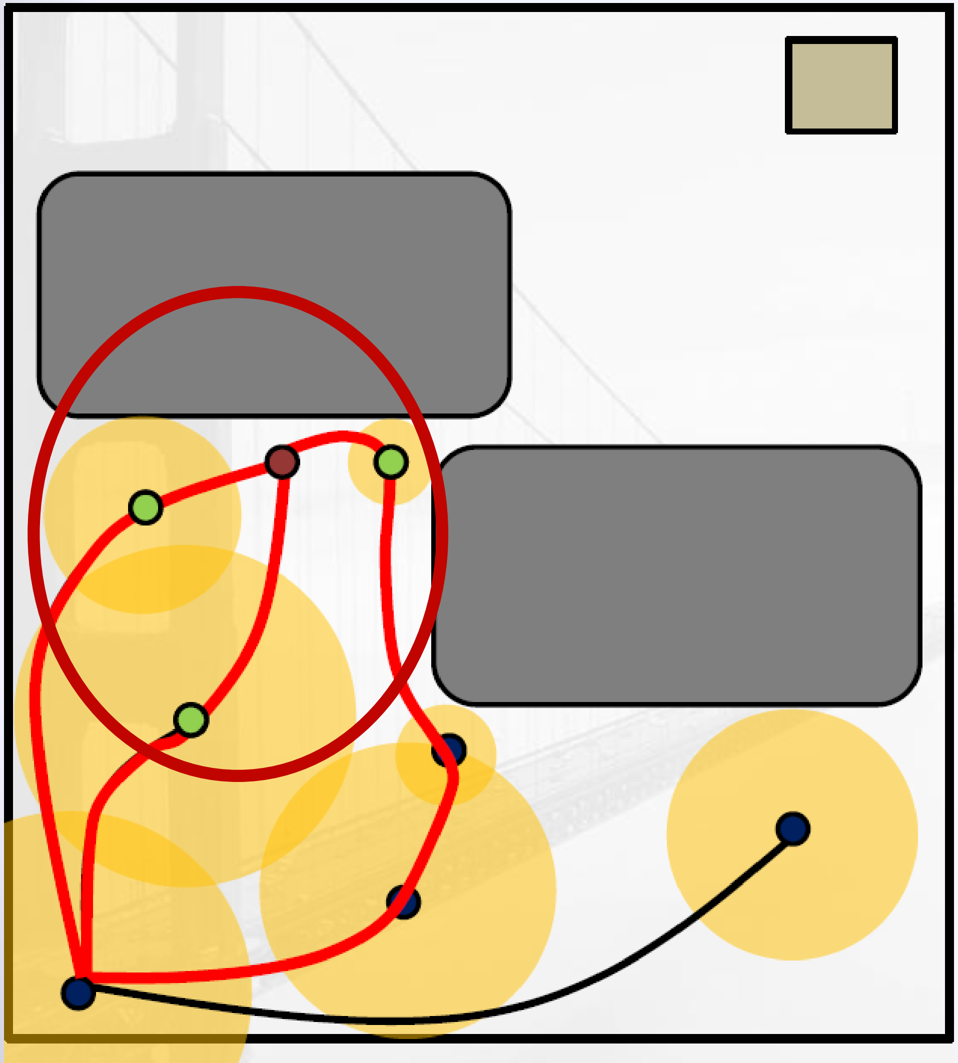
Example: Re-Wiring Operation



Identify nodes in a ball

From ball tree paper

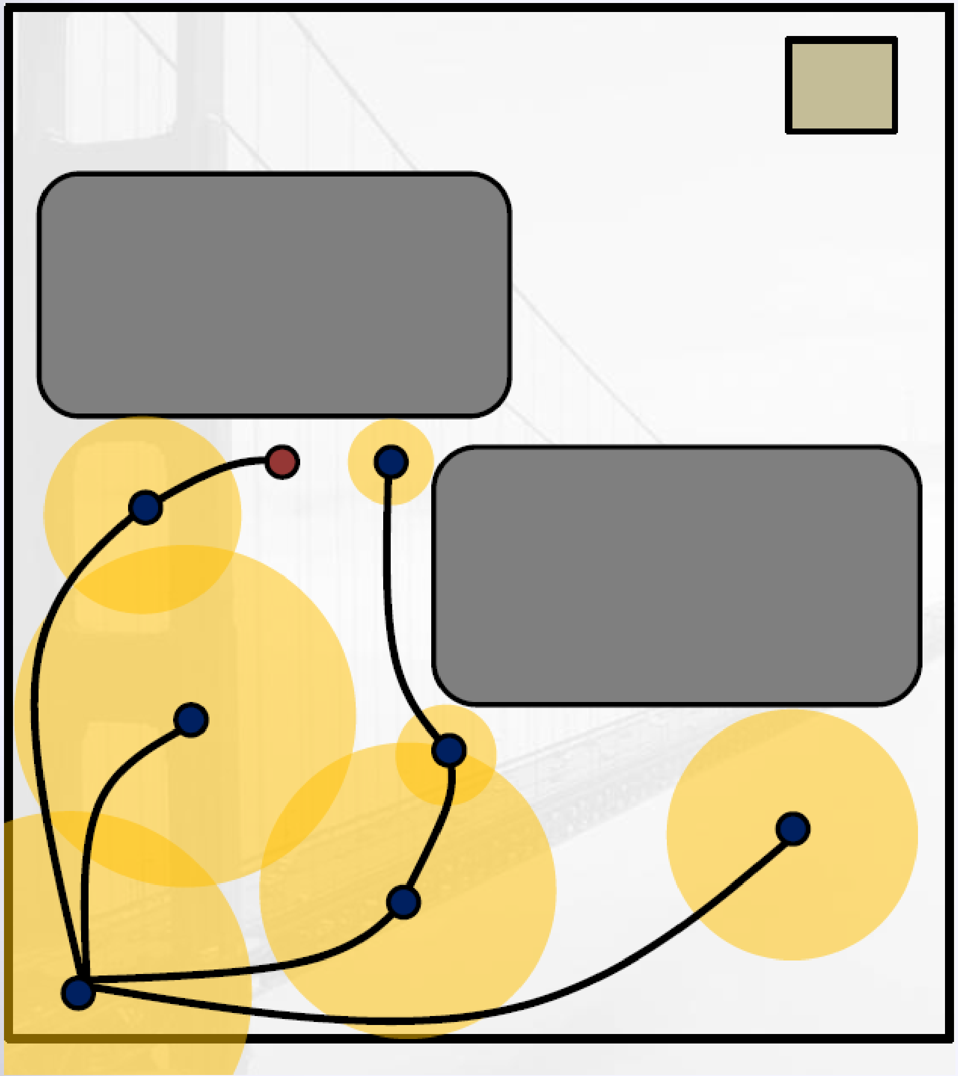
Example: Re-Wiring Operation



Identify which parent gives the lowest cost

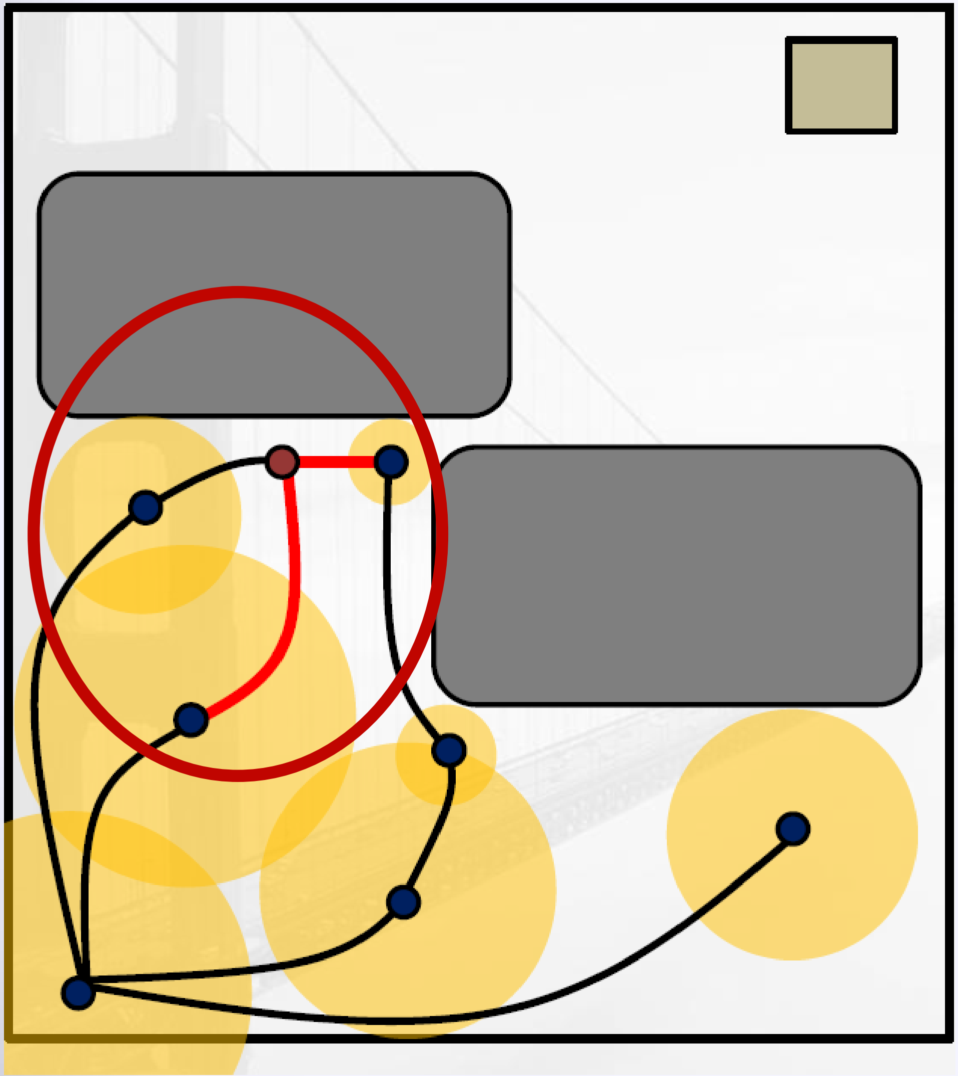
From ball tree paper

Example: Re-Wiring Operation



From ball tree paper

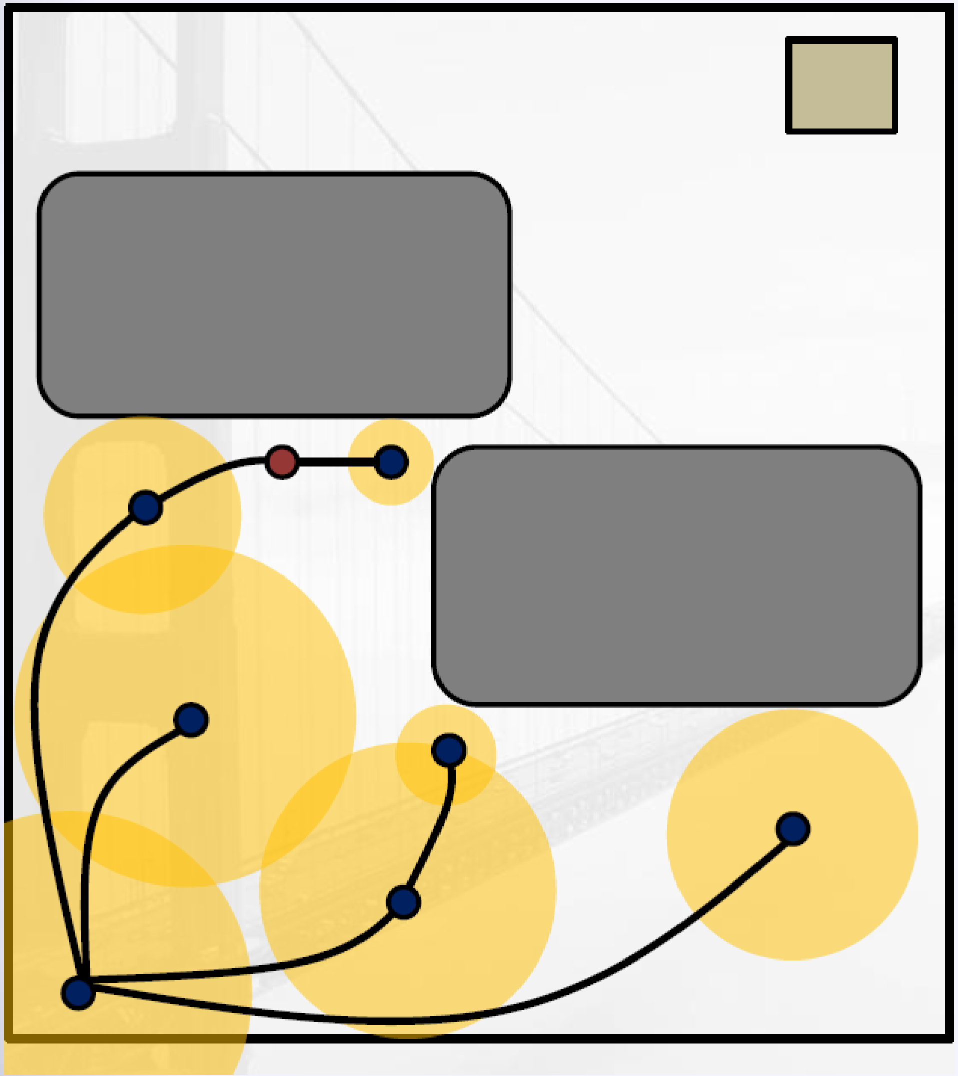
Example: Re-Wiring Operation



Identify which child gives the lowest cost

From ball tree paper

Example: Re-Wiring Operation



Video showing benefits with real robot

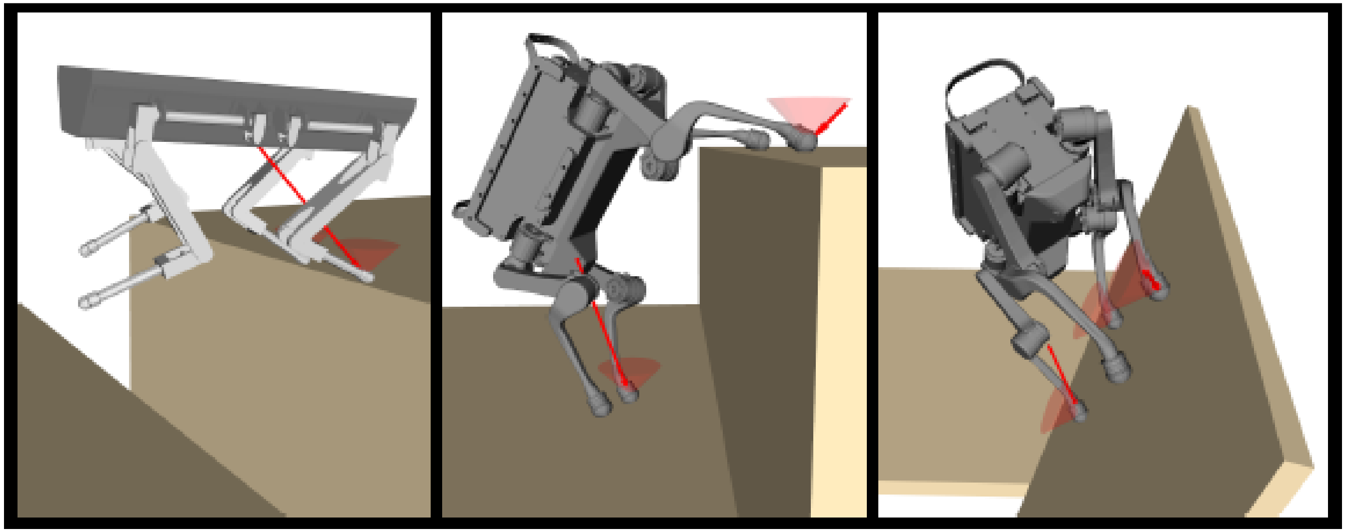
From ball tree paper

Kinodynamic Path Planning

$$\text{ALSO GIVEN:} \quad h_i(q,\dot{q},\ddot{q}) \le 0,\ h_i(q,\dot{q},\ddot{q}) = 0,\ \ldots$$

$$\text{FIND:} \quad \tau \text{ that satisfies } f_i(q),\ g_i(q,\dot{q}),\ h_i(q,\dot{q},\ddot{q})$$

- **Consider kinematic + dynamic constraints**



Gait and Trajectory Optimization for Legged Systems through Phase-based End-Effector Parameterization

State Space Formulation

- **Kinodynamic planning → 2n-dimensional state space**

C denote the C-space

X denote the state space

$$x = (q, \dot{q}), \text{ for } q \in C, x \in X$$

$$x = [q_1 \;\; q_2 \;\; \dots \;\; q_n \;\; \frac{dq_1}{dt} \;\; \frac{dq_2}{dt} \;\; \dots \;\; \frac{dq_n}{dt}]$$

Constraints in State Space

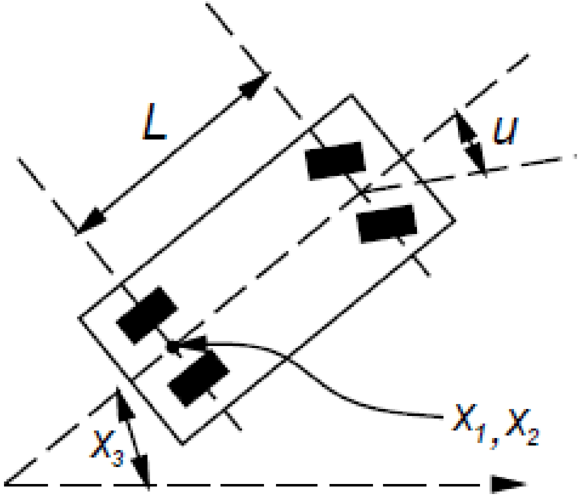
$$h_i(q, \dot{q}, \ddot{q}) = 0 \text{ becomes } G_i(x, \dot{x}) = 0,$$

$$\text{for } i = 1, \ldots, m \text{ and } m < 2n$$

- **Constraints can be written in:**

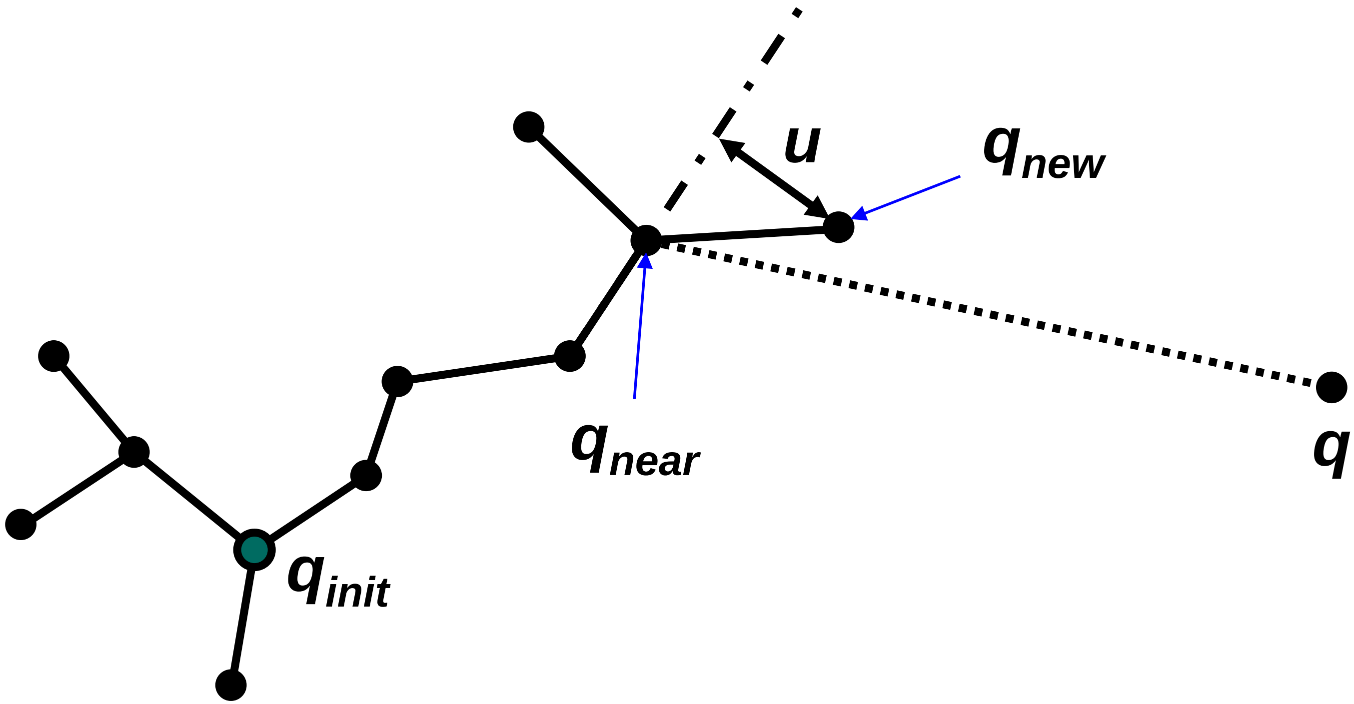
$$\dot{x} = f(x, u)$$

$$u \in U, \quad U: \text{Set of allowable controls or inputs}$$



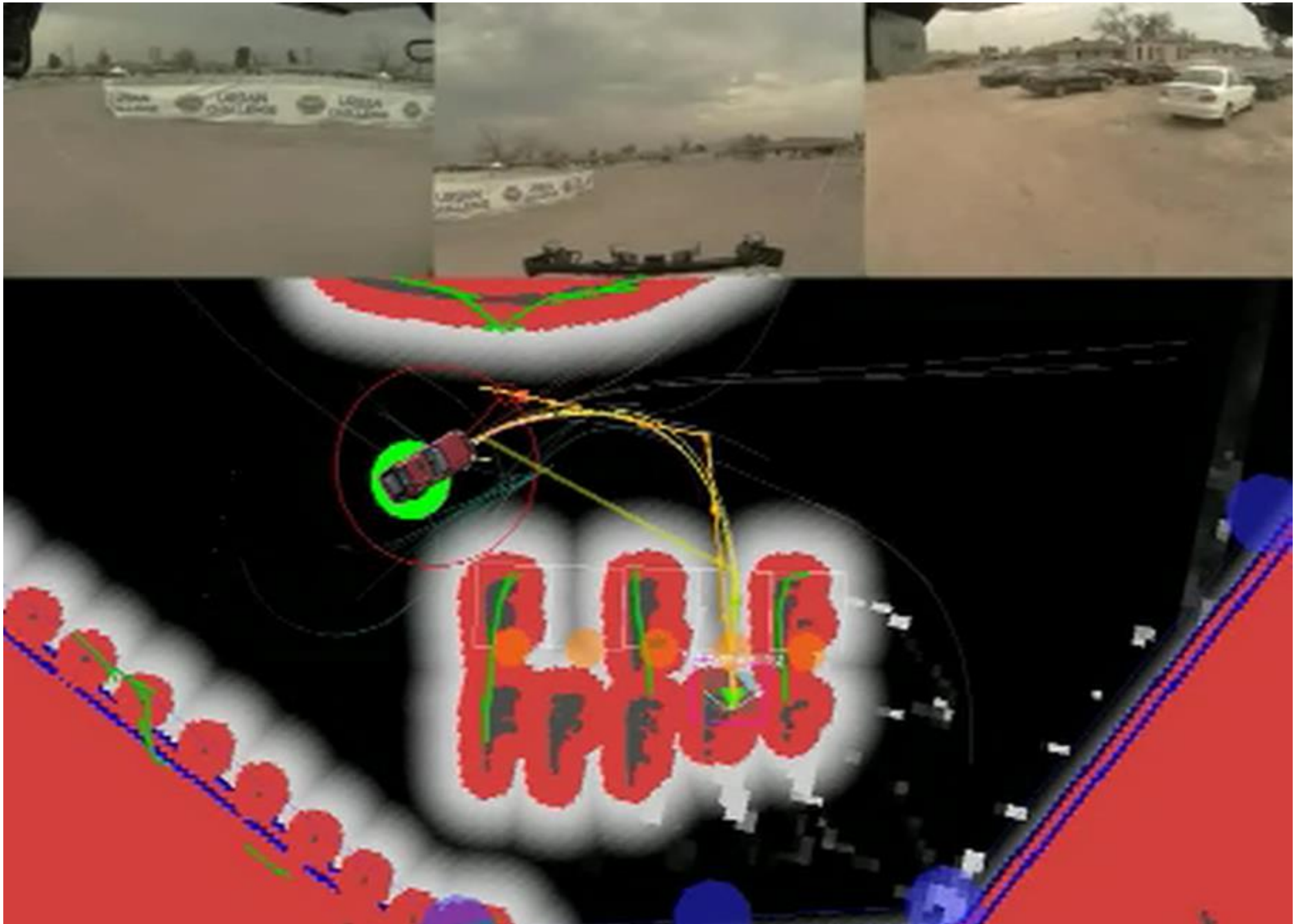
Rapidly-Exploring Random Tree

- **Extend a new vertex in each iteration**



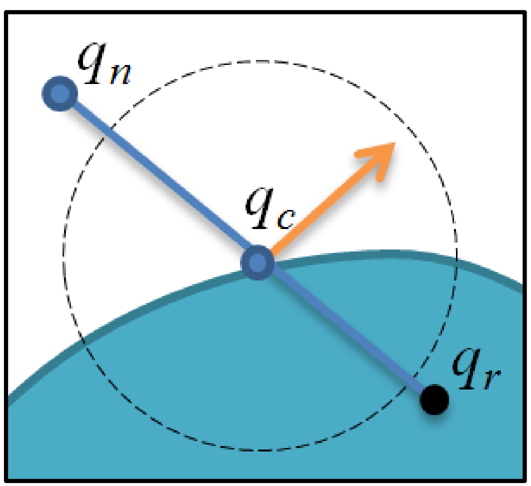
KAIST

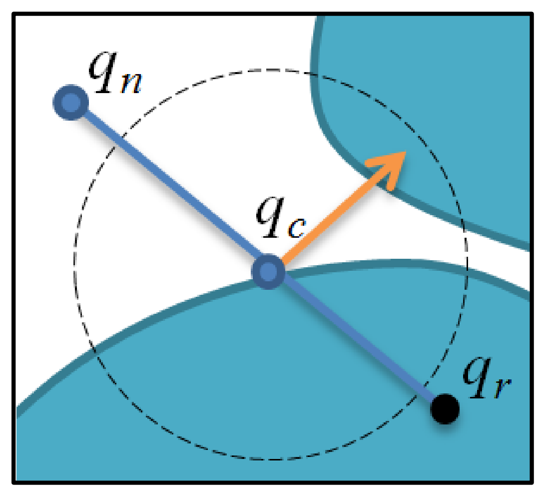
RRT at work: Successful Parking Maneuver



Some 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**

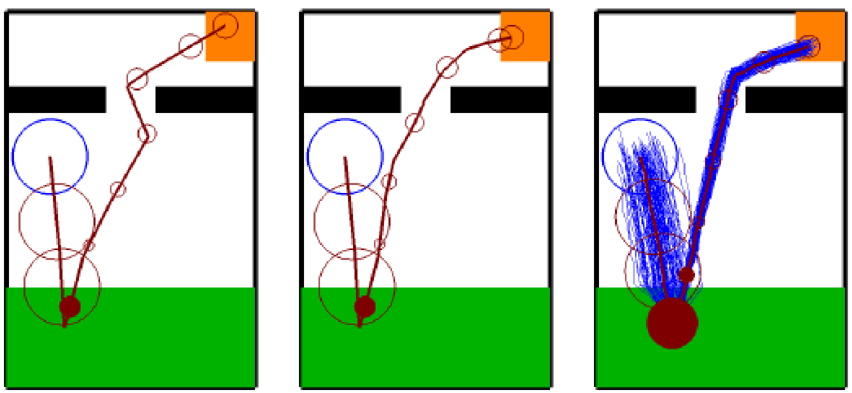




KAIST

Handling uncertainty and dynamic objects

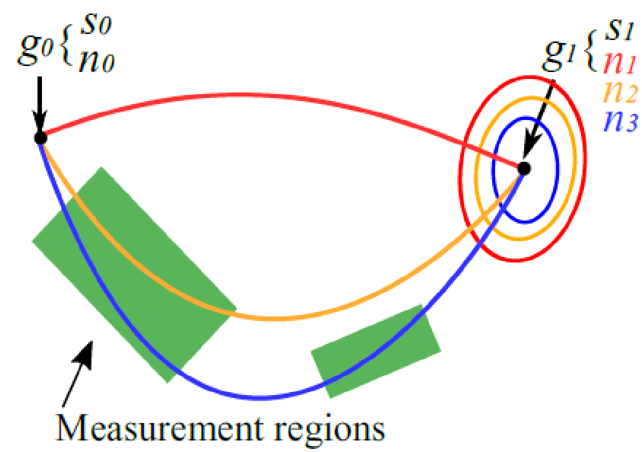
- **Anytime RRBT for handling uncertainty and dynamic objects, IROS 16**



500 1000 1500

Number of iteration

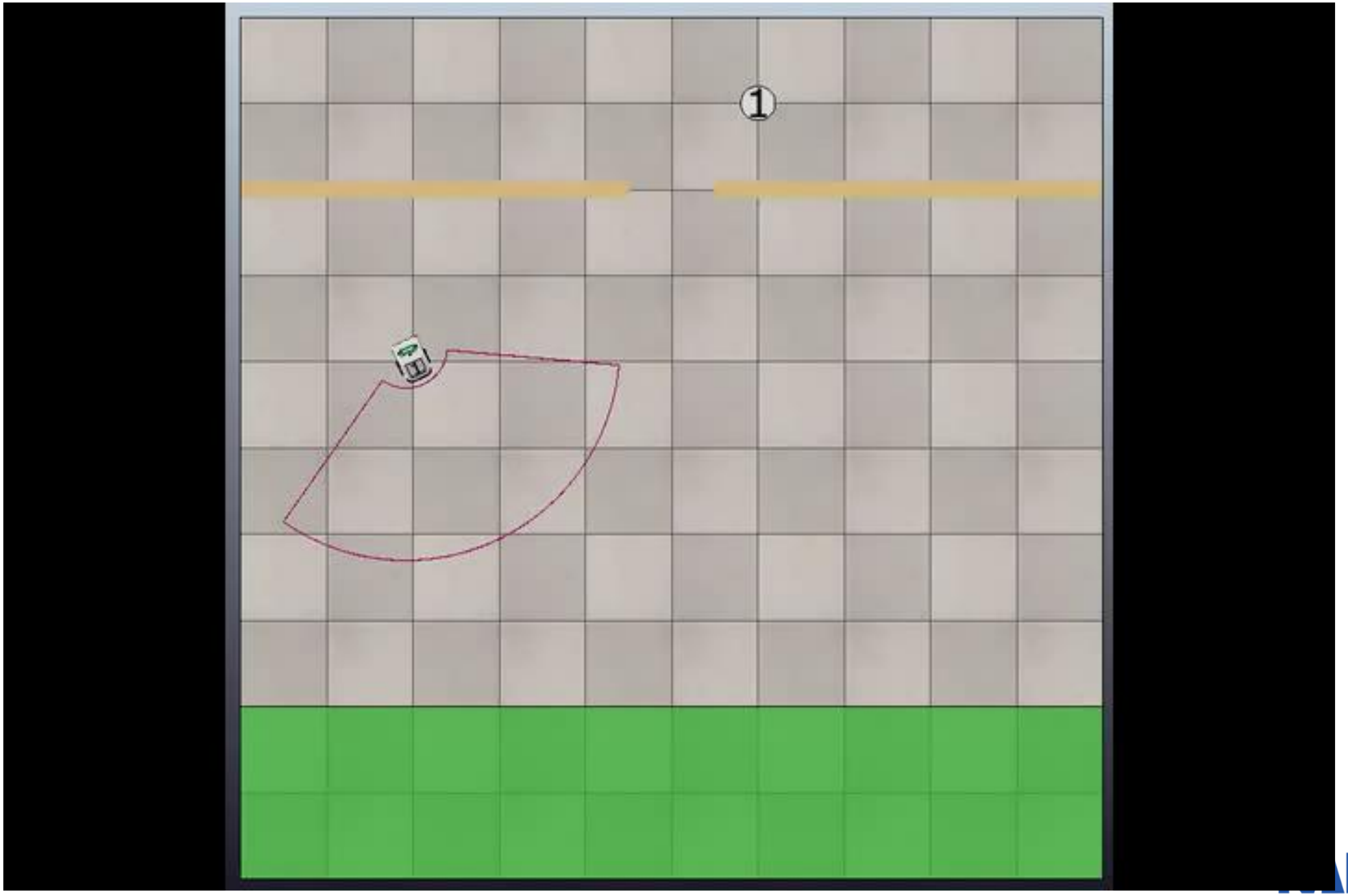
Multiple belief nodes in the same vertex



Preserve optimal path

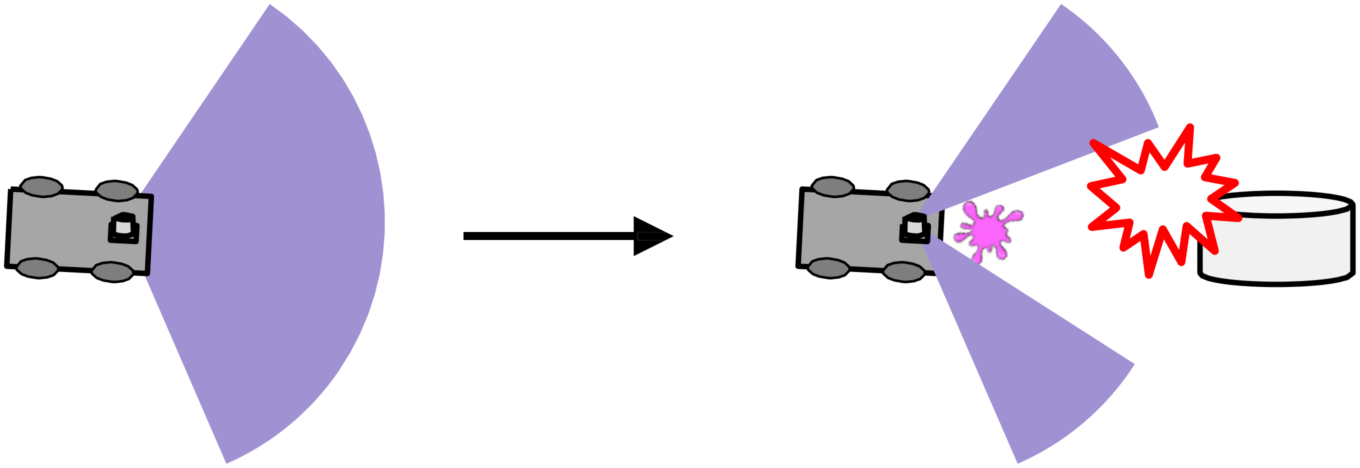
KAIST

Main Contribution: Anytime Extension



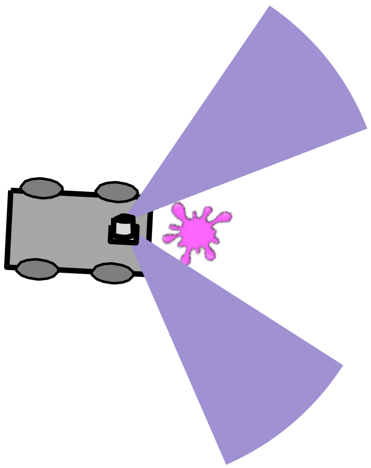
Confidence-based Robot Navigation under Sensor Occlusion w/ Deep Reinforcement Learning, ICRA 22

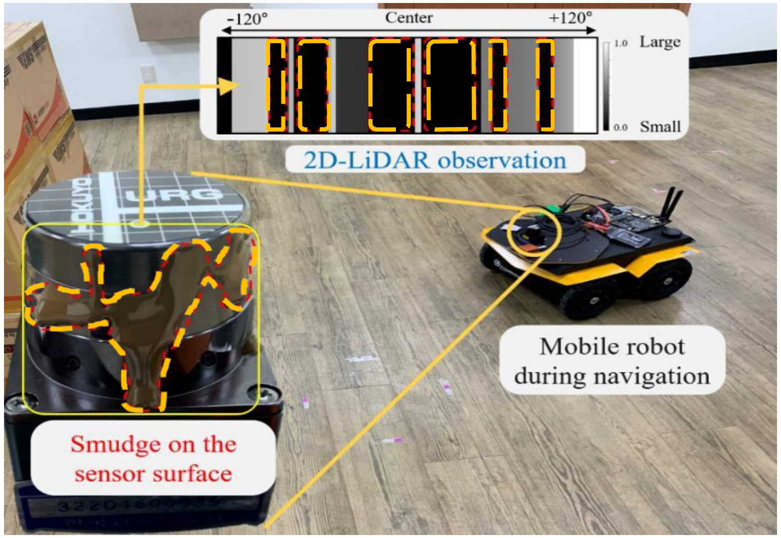
- **Robot navigation under sensor occlusion**
  - **LiDAR based navigation often suffer from unexpected occlusion on (e.g., dust, water, or smudge) sensor surface**
  - **Such occlusion lowers the visibility of the sensor and might cause potential collisions.**



Confidence-based Robot Navigation under Sensor Occlusion w/ Deep Reinforcement Learning, ICRA 22

- **Our goal**
  - **Build a robot navigation policy robust to such sensor occlusion**





Occlusions on the real sensor surface

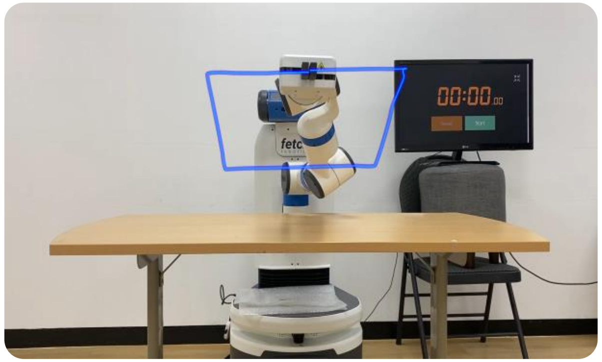
Received Outstanding Navigation Award Finalist

Hybrid Planning Techniques

- **Traditional methods have been carefully designed and worked quite well in many cases**
- **Learning approaches are showing interesting success, yet have limitations such as data hungry, high computation, and handling global information**
- **Interesting to combine those two orthogonal approaches together!**

Learning-based Initialization of Trajectory Optimization for Path-following Problems of Redundant Manipulators

- Problem Statement of Path-following Problems
- Generate a joint trajectory precisely following a given 6-dimensional Cartesian path (i.e., target path) with an end-effector.



Target path: ‘Hello’

Target path: ‘Square’

Target path: ‘Zigzag’

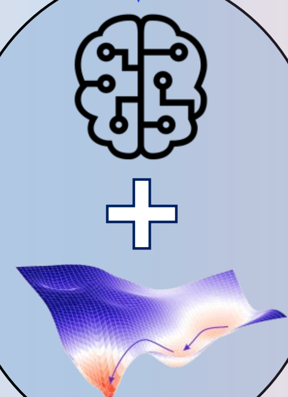
Take-home Message

- **Integrating learning and planning** is an important strategy that works in a complementary manner.
  → Improves accuracy and efficiency by combining the two approaches.

Learning-based methods

→ may not guarantee optimality

→ but offer a good starting point for optimization quickly.



Optimization-based methods

→ may struggle with high-dimensional and non-convex problems

→ but find optimal solutions around starting point by iterative refinement.

IEEE

IEEE ROBOTICS AND AUTOMATION SOCIETY

Outstanding Planning Paper Award

IEEE International Conference on Robotics and Automation – ICRA 2023

For the paper by

Minsung Yoon, Mincheul Kang, Daehyung Park, and Sung-Eui Yoon

"Learning-based Initialization of Trajectory Optimization for Path-following Problems of Redundant Manipulators"

Frank Park
Society President

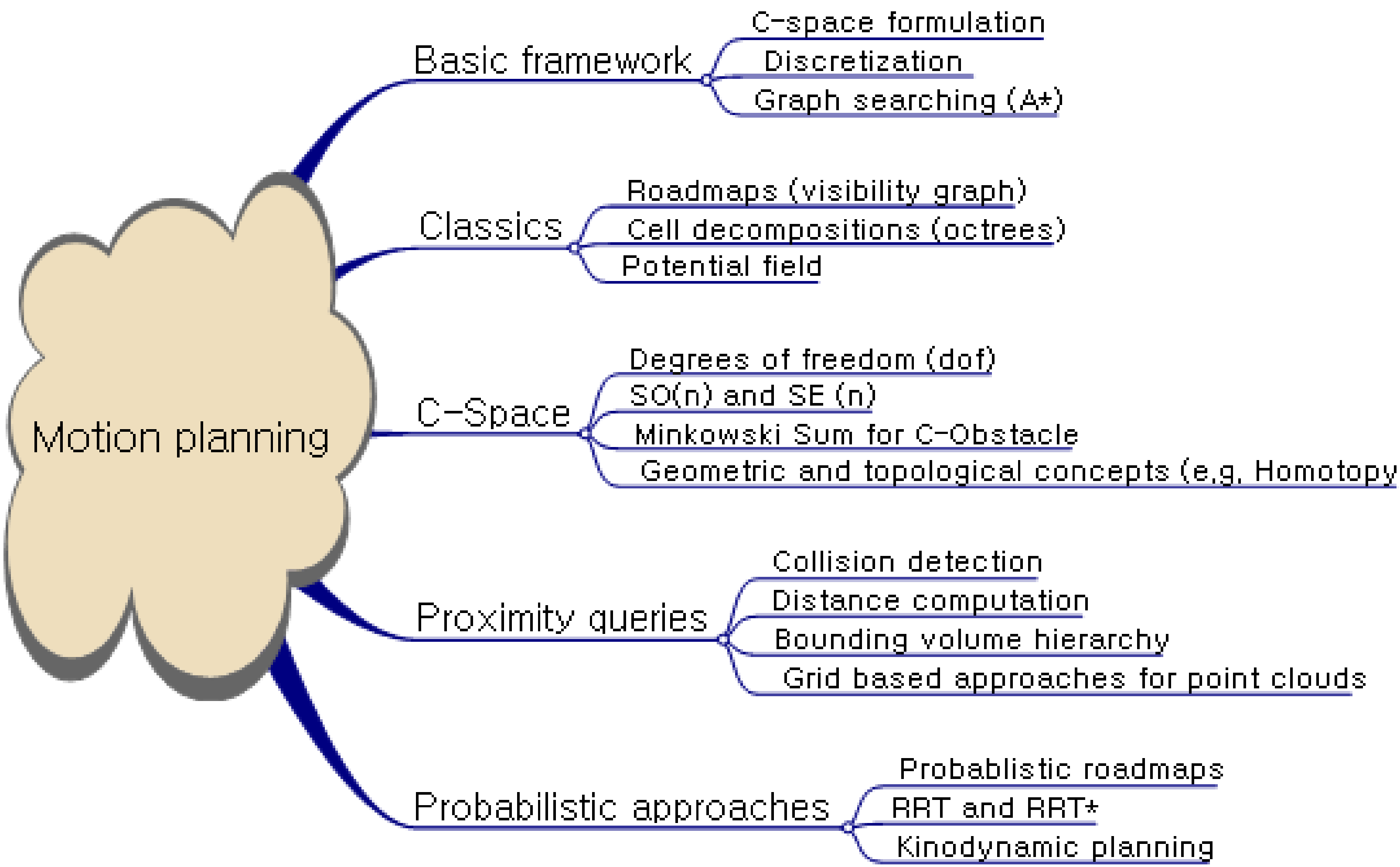
Maren Bennewitz
ICRA 2023 Award Chair

June 2023

Class Objectives were:

- **Understand the RRT technique and its recent advancements**
  - **RRT* for optimal path planning**
  - **Kinodynamic planning**
  - **Some related techniques to RRT**

Summary



Motion planning
Basic framework
C-space formulation
Discretization
Graph searching (A*)
Classics
Roadmaps (visibility graph)
Cell decompositions (octrees)
Potential field
C-Space
Degrees of freedom (dof)
SO(n) and SE (n)
Minkowski Sum for C-Obstacle
Geometric and topological concepts (e.g. Homotopy)
Proximity queries
Collision detection
Distance computation
Bounding volume hierarchy
Grid based approaches for point clouds
Probabilistic approaches
Probablistic roadmaps
RRT and RRT*
Kinodynamic planning

Next Time..

- **Basic concepts of reinforcement learning**

Homework for Every Class

- **Submit summaries of 2 ICRA/IROS/RSS/CoRL/TRO/IJRR papers**
- **Go over the next lecture slides**
- **Come up with two question submissions before the mid-term exam**