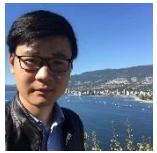
深蓝学院
shenlanxueyuan.com

Motion Planning for Mobile Robots



主讲人 Fei Gao

Ph.D. in Robotics
Hong Kong University of Science and Technology
Tenured Associate Professor，Zhejiang University

Introduction

About this Course

- This course is about:
  - Academism (old school) planning pipeline
  - Path finding algorithm
  - Trajectory generation/optimization
  - Real-time software implementation
- This course is NOT about:
  - Dynamics Modelling ☹
  - Robot design ☹
  - End-to-end navigation ☹
  - Learning-based method ☹
  - Mathematical/theoretical proof ☹
- This course may also cover:
  - Autonomy for mobile robots
  - Paper recommendations
  - Cutting-edge research direction/ethics

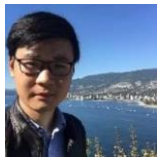
Basic Expectation

- Discipline:
  - Every researcher has his own taste/style. Since you choose this course, please follow my style.
  - Q&A only at appointed office time.
  - Finish your homework by yourself.
- Project:
  - Basic algorithm validation (MATLAB)
  - Sophisticated engineering implementation (ROS/C++)

每位研究者都有自己的风格和品味，如果你们选择了这门课，那请你们在这门课的范围内，尽量follow我们的风格。我相信，如果大家跟随老师的风格，无论如何，大家最后都会收获很多很多。

我简单说一下我的风格：**我的风格一定是实用第一**，任何算法如果没有很好的实用价值，那我就认为它在实际的机器人系统上没有太大实际意义。我教给大家的算法，希望不要仅停留在Matlab的数值验证层面，更要在C++或者ROS里面跑起来，在嵌入式系统中跑起来。这就是我的风格！

Teaching Team



Instructor 1: Fei Gao

Ph.D. in Robotics
Hong Kong University of Science and Technology
Assistant Professor，Zhejiang University

Instructor 2: Hongkai Ye

D. Eng. in Robotics
Zhejiang University

Instructor 3: Zhepei Wang

Ph.D. in Robotics
Zhejiang University

Instructor 4: Jialin Ji

D. Eng. Candidate in Robotics
Zhejiang University

Instructor 5: Xin Zhou

D. Eng. Candidate in Robotics
Zhejiang University

Instructor 6: Qianhao Wang

Ph.D. Candidate in Robotics
Zhejiang University

Workload

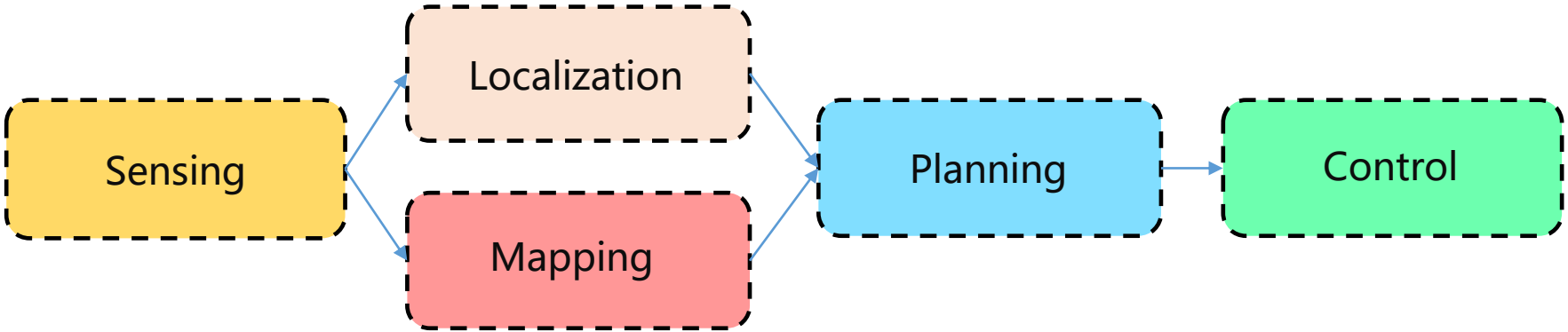
- Expected Student Background:
  - Linear algebra
  - Probability
  - MATLAB programming skills
  - C++ programming skills (VERY IMPORTANT)
  - Linux
  - Love robots ☺
- Workload:
  - Attend lectures
  - Lots of project work
  - Have fun with robots ☺

What is autonomous robot

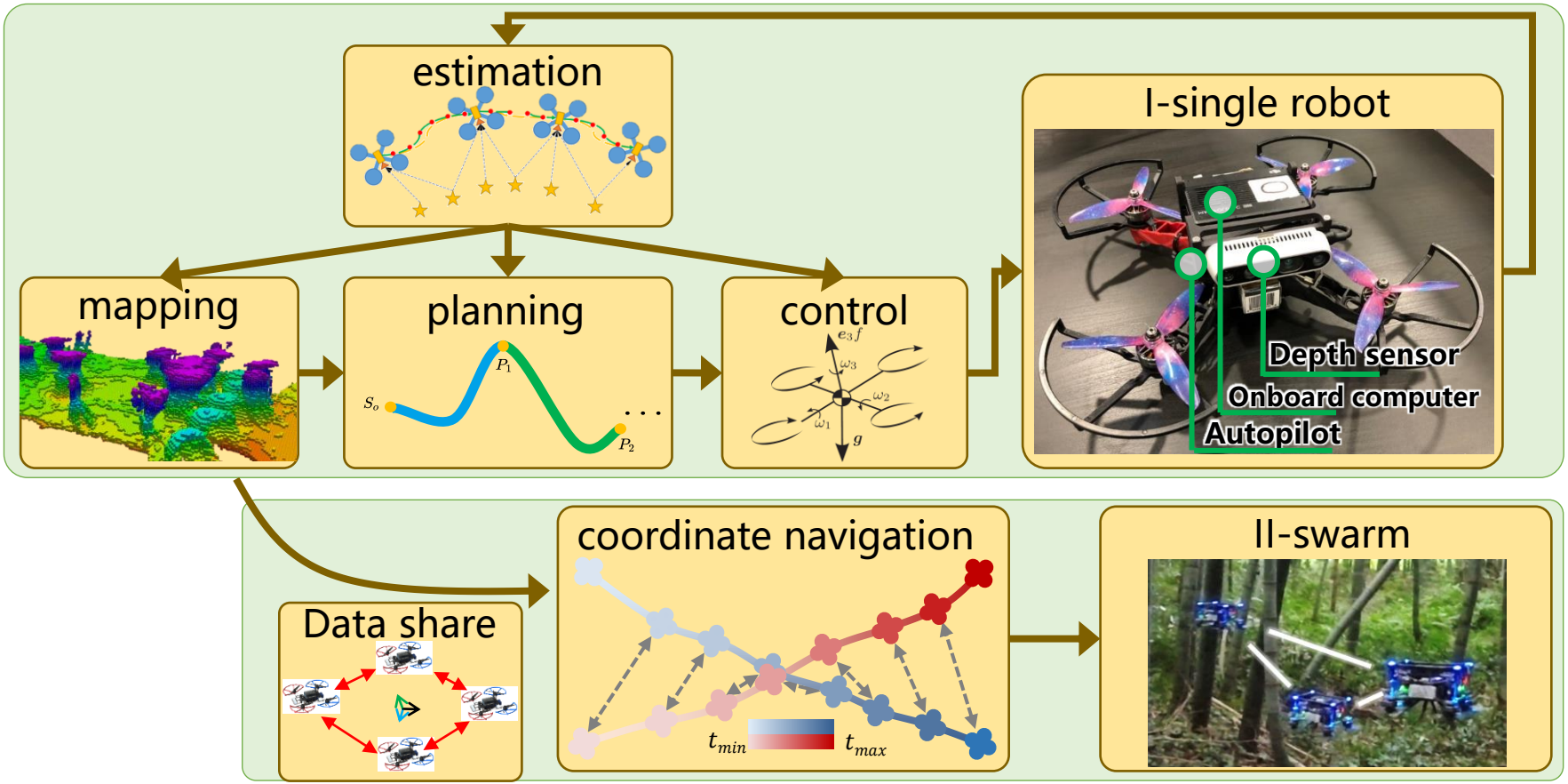
Definition: an autonomous robot is a robot that performs behaviors or tasks with a high degree of autonomy (without external influence).

Subjects: Computer Science, Automation, Mechanism, Electronics …

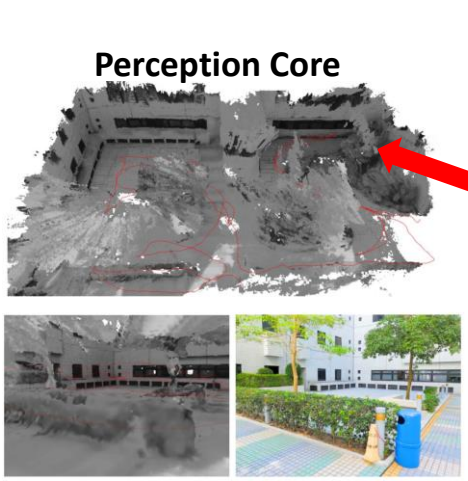
- Perception-Planning-Control action loop

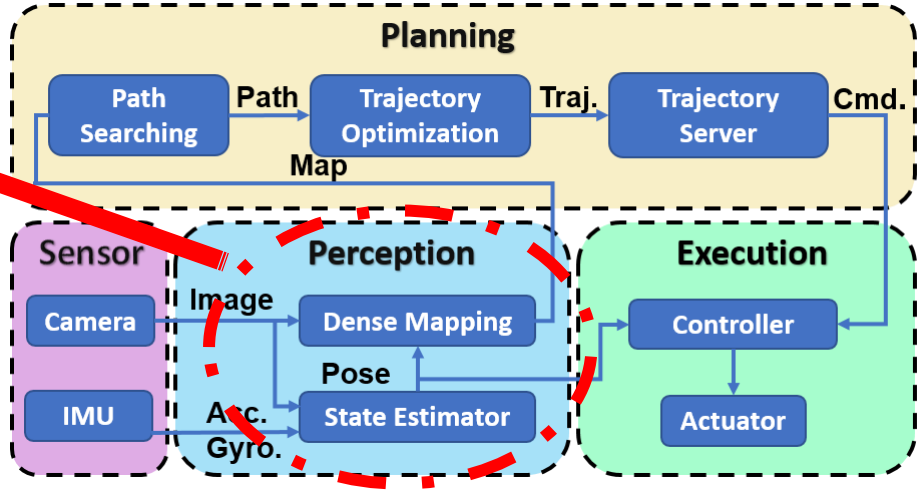


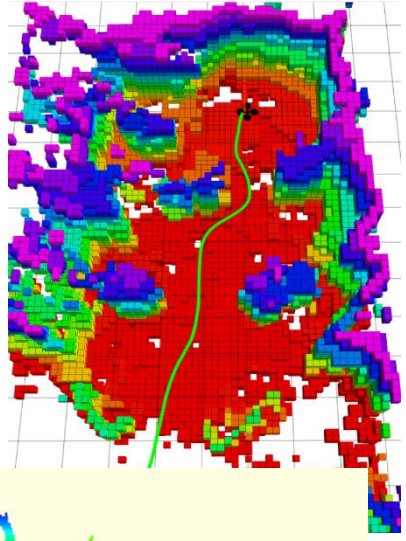
What is autonomous robot

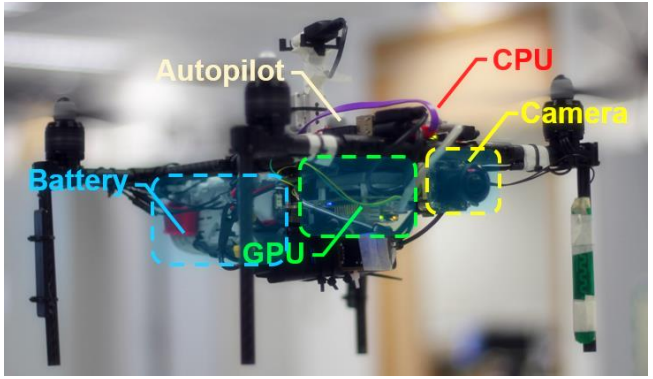


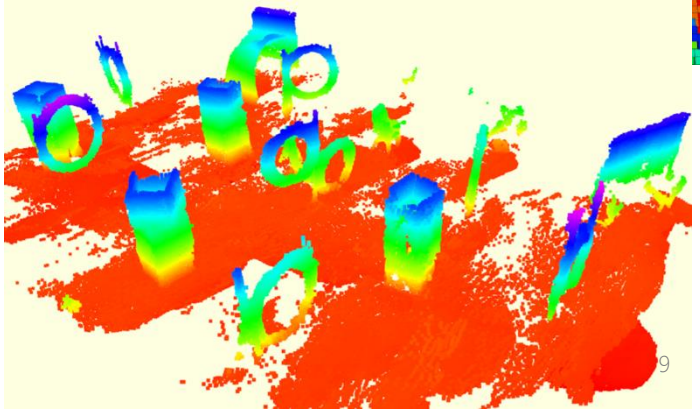
Vision-based Autonomous Drone





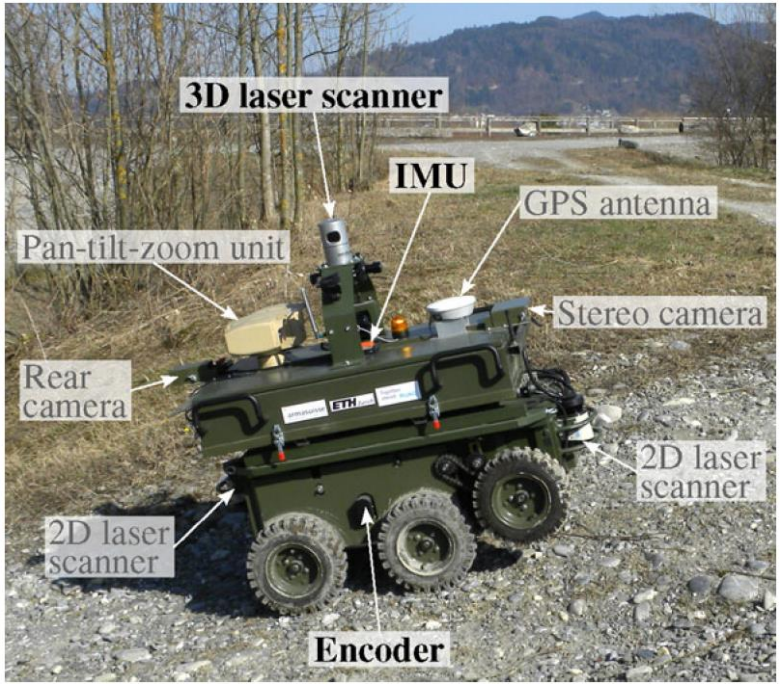






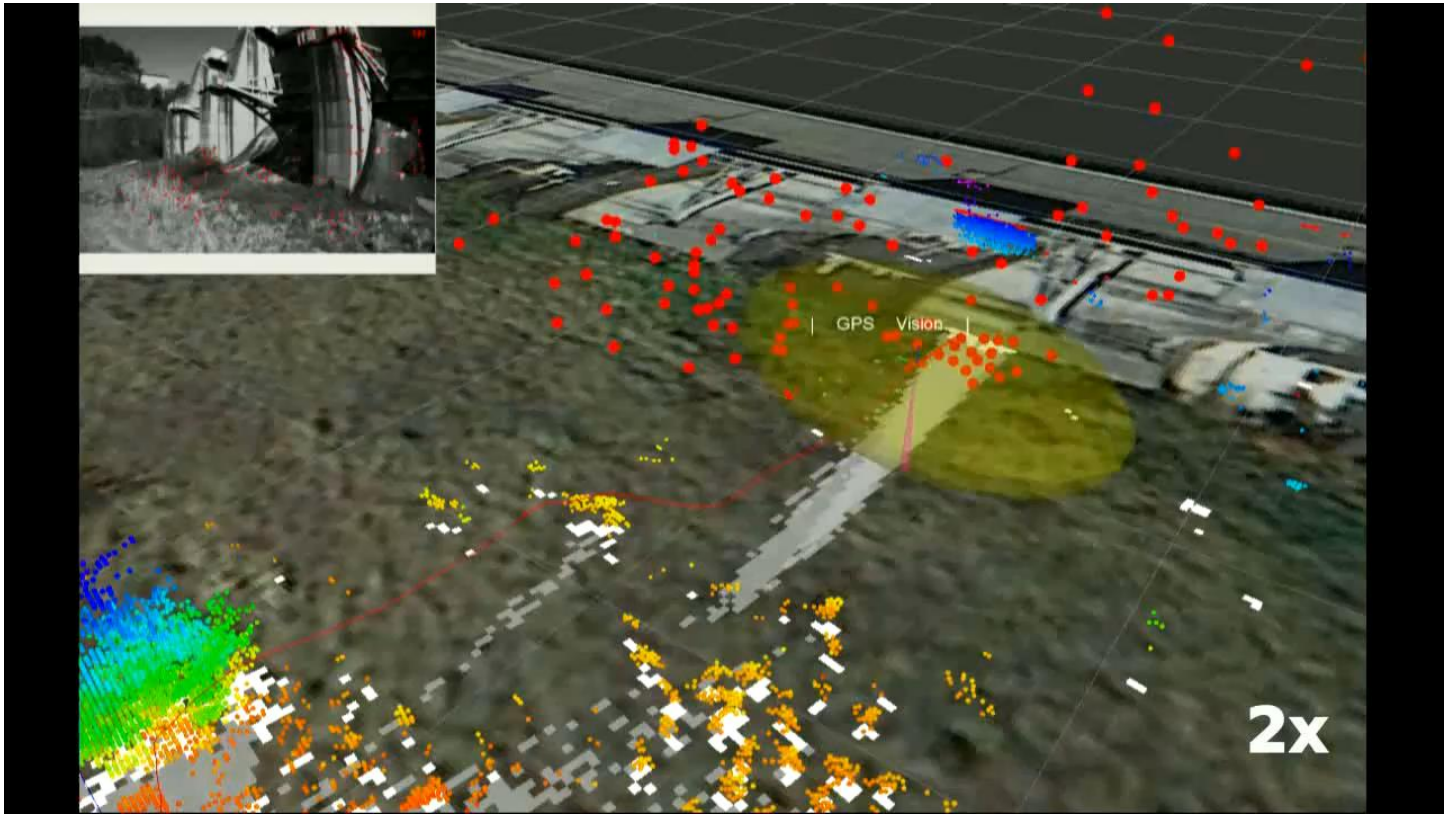
Real-time monocular dense mapping on aerial robots using visual-inertial fusion, **ICRA** 2017

Autonomous robot: applications



Aerial Robots

Dam Inspection



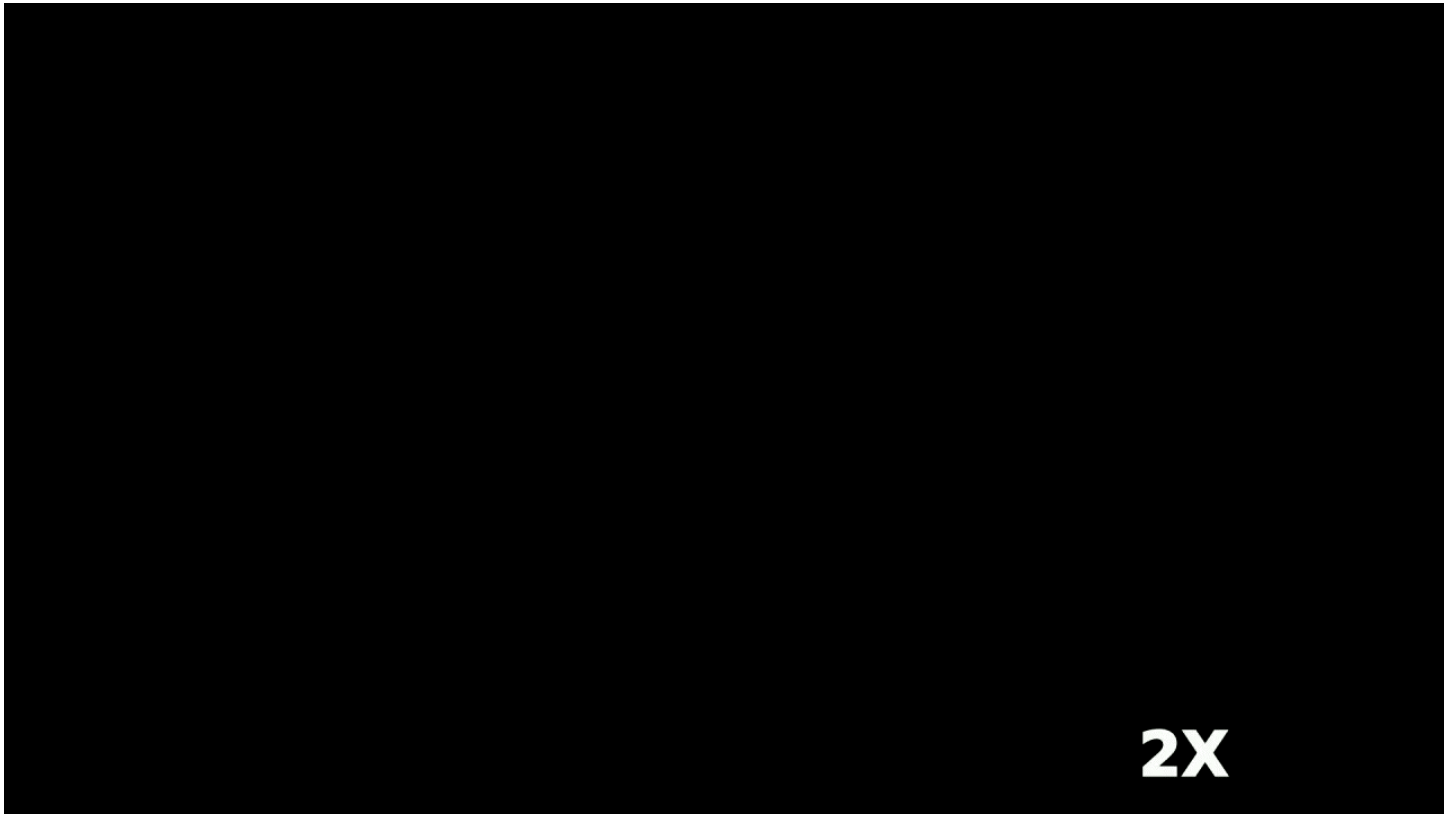
Shen, et al, 2014

Dam Inspection

Ozaslan, et al, 2014

Cellular Tower Inspection

Precision Farming



Shen, et al, 2014

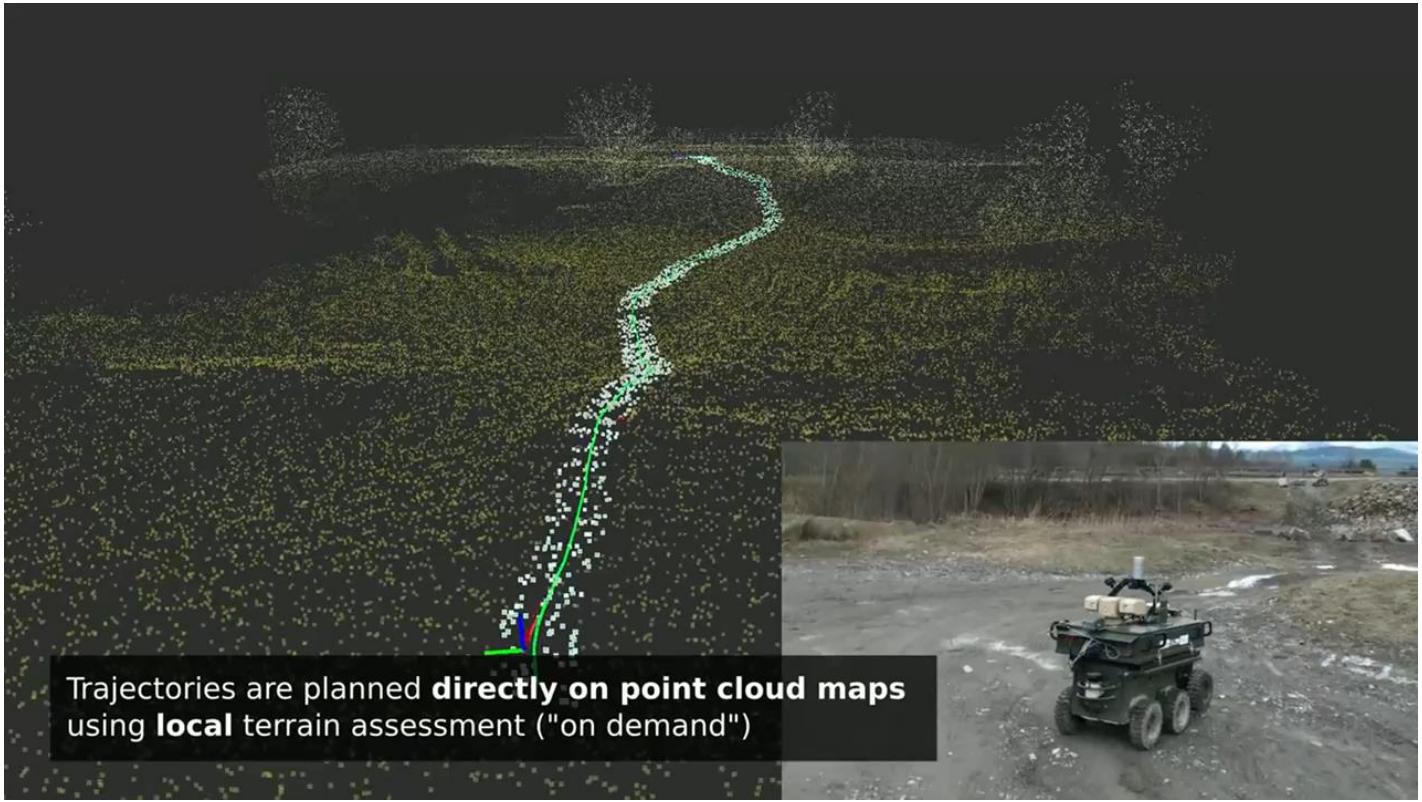
Delivery

Ground Robots

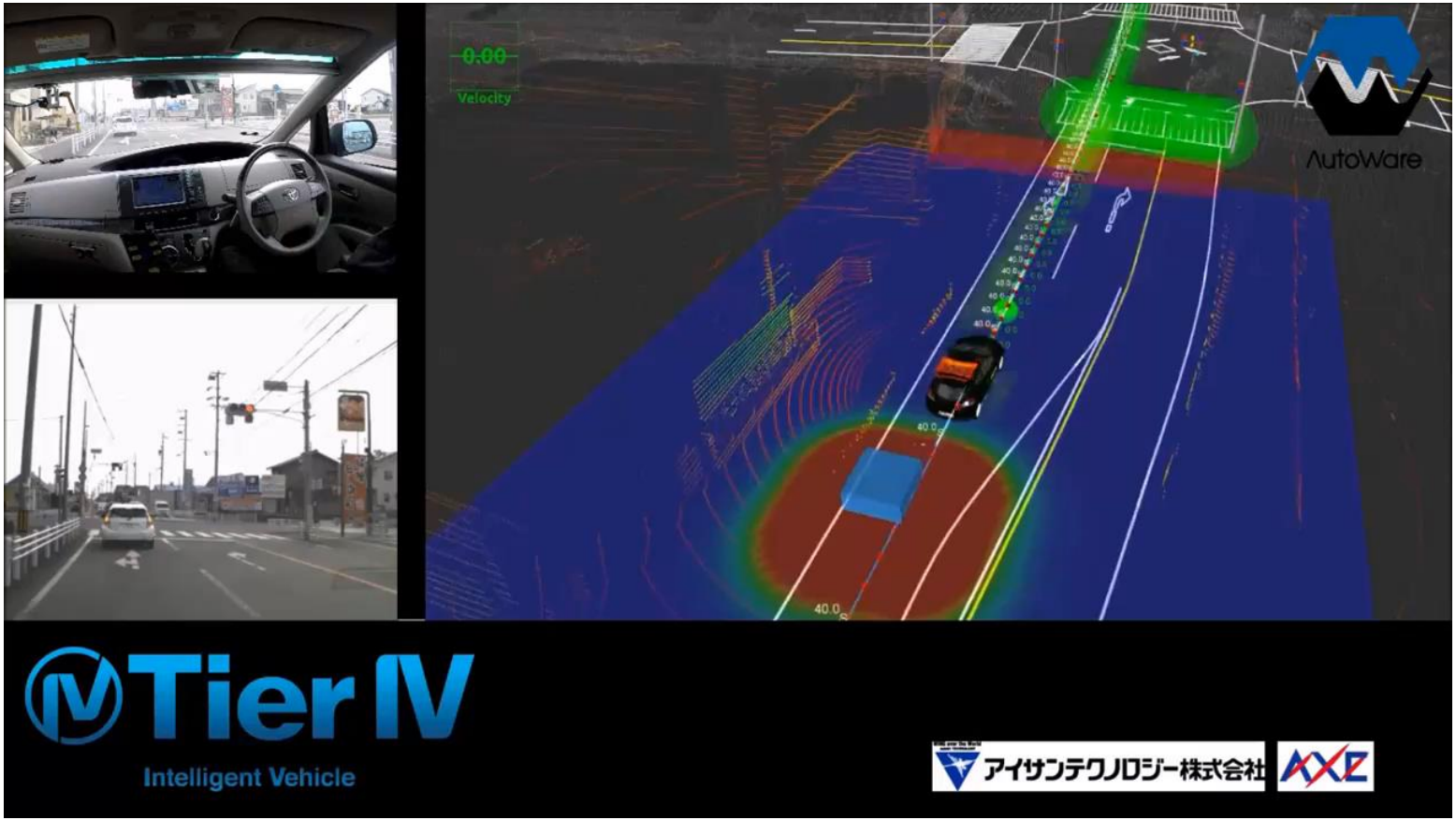
Search and Rescue

Housekeeping

Hills Driving



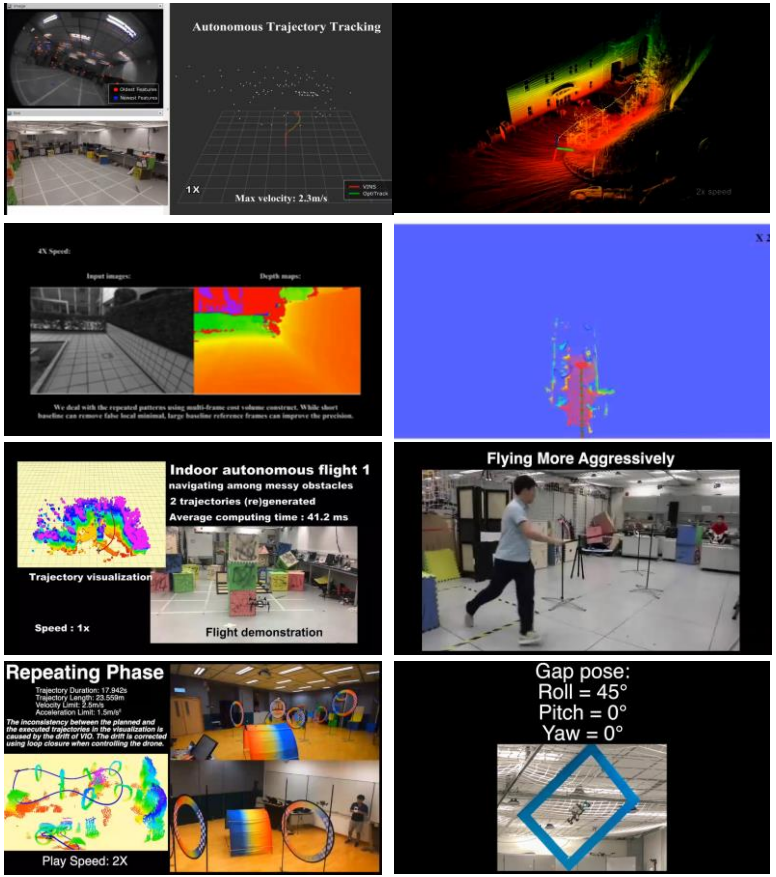
Urban Driving



Crowd Driving

What is autonomous robot

- Estimation
  - Low latency
  - High accuracy & consistency
- Perception
  - 3D sensing & dense perception
  - Map fusion & integration for planning
- Planning
  - Complex & unknown environments
  - Safety & dynamical feasibility
  - Limited sensing & computation
- Control
  - Aggressive maneuvers
  - Smooth trajectory tracking



What is motion planning

- Basic requirements
  - Safety: collision avoidance
  - Smoothness: energy saving, comfort
  - Kinodynamic feasibility: executable, controllable
- Old-school pipeline
  - Front-end path finding
    - Search for an initial safe path
    - Low dimensional
    - Discrete space
  - Back-end trajectory generation
    - Search for an executable trajectory
    - High dimensional
    - Continuous space

How to do robotics research

- Find a problem
  - ✓ A problem only in your imagination is not a problem at all.
  - ✓ A robotician must be an engineer first.
  - ✓ Hot topic is just hot.
  - ✓ Be honest.
- Solve a problem
  - ✓ Don’t intentionally making things complicated.
  - ✓ Simple but effective solution is always preferable.
  - ✓ Simulation tells nothing, show everyone real robots.
  - ✓ Solve real problems.
  - ✓ Solve it 100%, or not.

How to do motion planning

- Overall knowledge of planning
  - ✓ Choose suitable methods for different scenarios.
  - ✓ Design customized strategy.
- Dirty hands
  - ✓ Don’t wait, don’t just read papers. Do it yourself.
  - ✓ A lot of coding work.
  - ✓ A lot of field work.
- Know the whole system well
  - ✓ Take care every component in your robot.

Course Outline

Front-end: Path finding

- SEARCH-BASED PATH FINDING
  - Graph Search Basis
  - Dijkstra and A*
  - Jump Point Search
  - Homework
- SAMPLING-BASED PATH FINDING
  - Probabilistic Road Map
  - Rapidly-exploring Random Tree (RRT)
  - Optimal Sampling-based Methods
  - Advanced Sampling-based Methods
  - Homework
- KINODYNAMIC PATH FINDING
  - Introduction
  - State-state Boundary Value Optimal Control Problem
  - State Lattice Search
  - Kinodynamic RRT*
  - Hybrid A*
  - Homework

Back-end: Trajectory Generation

- MINIMUM SNAP TRAJECTORY GENERATION
  - Differential Flatness
  - Minimum Snap Optimization
  - Closed-form Solution to Minimum Snap
  - Time Allocation
  - Implementation in Practice
  - Homework

MPC

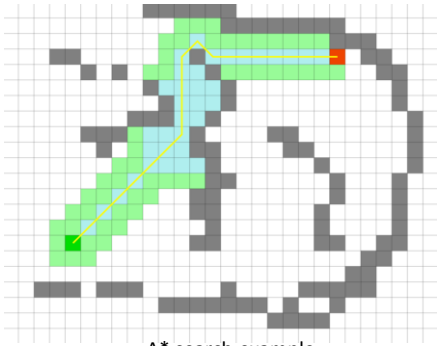
- MODEL PREDICTIVE CONTROL FOR ROBOTICS PLANNING
  - Introduction
  - Linear MPC
  - Non-linear MPC
  - Homework

Overview

Front-end: Path Finding

- Sampling-based methods
  - Probabilistic roadmap (PRM)
  - Rapidly exploring random tree (RRT)
  - RRT*, informed RRT*
- Search-based methods
  - General graph search: DFS, BFS
  - Dijkstra and A* search
  - Jump point search

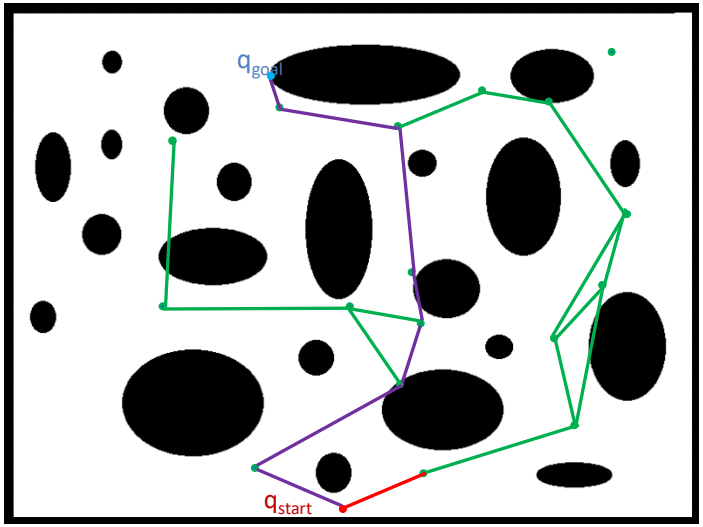
RRT example



A* search example

Sampling-based Method

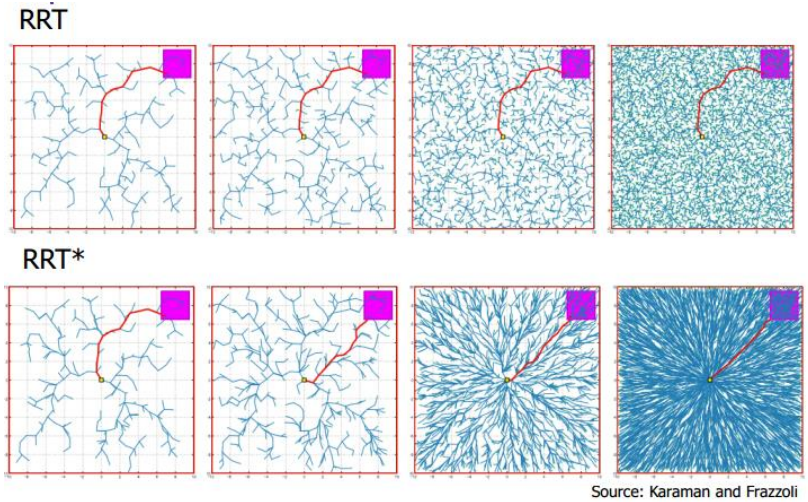
Probabilistic Roadmap (PRM)



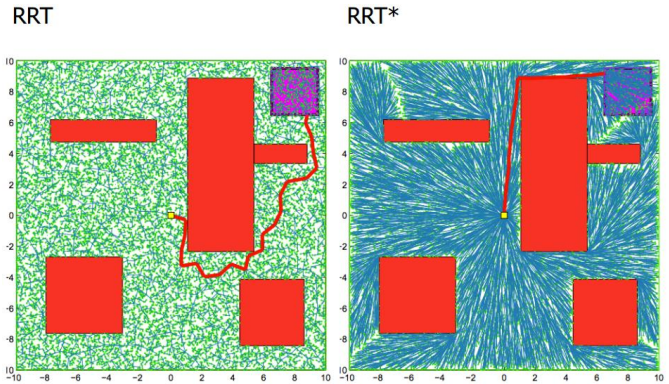
C-space

Sampling-based Method

RRT* vs RRT

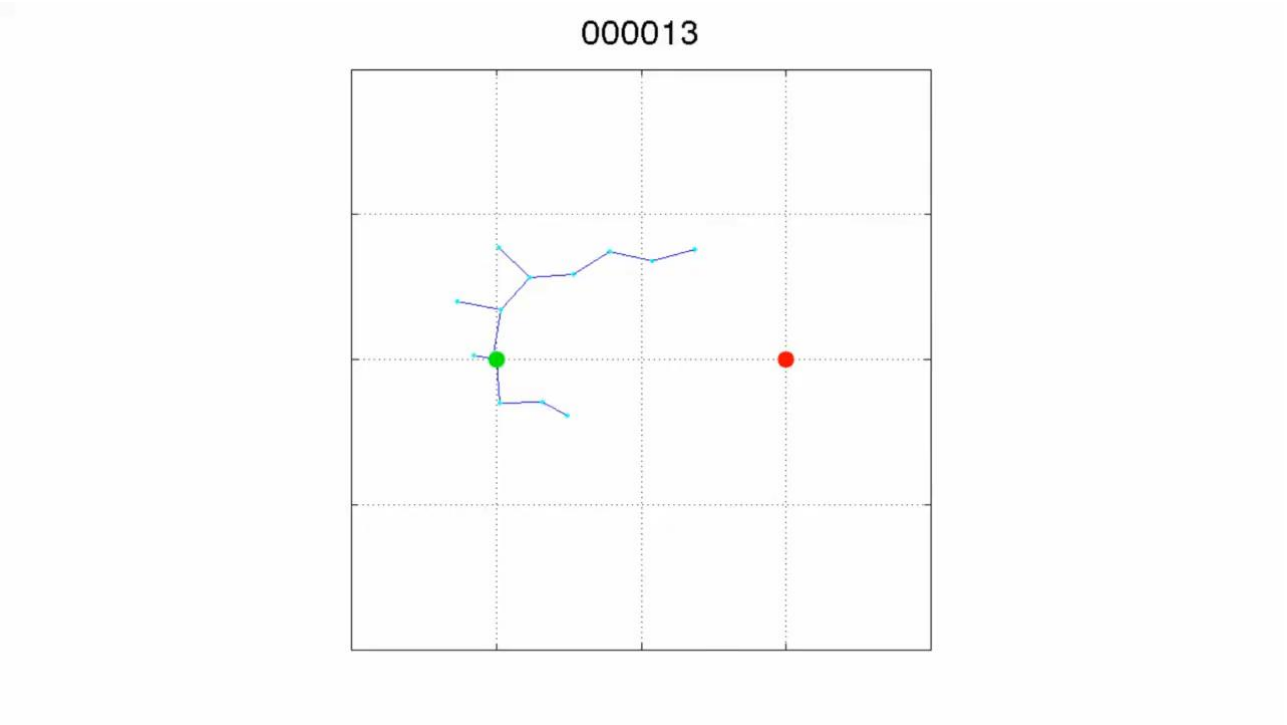


Source: Karaman and Frazzoli



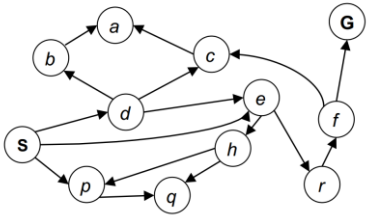
Sampling-based Method

Informed RRT*

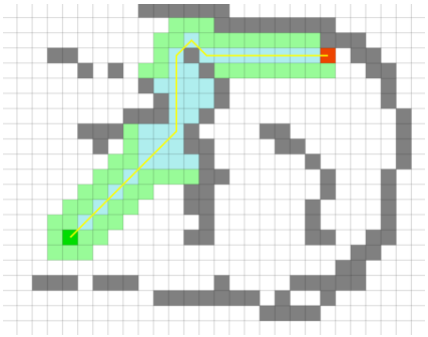


Search-based Method

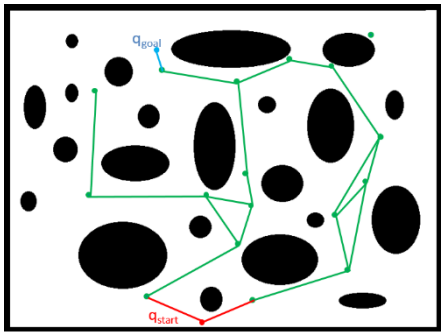
- For every search problem, there's a corresponding state space graph
- Connectivity between nodes in the graph is represented by (directed or undirected) edges



Ridiculously tiny search graph for a tiny search problem



Grid-based graph: use grid as vertices and grid connections as edges

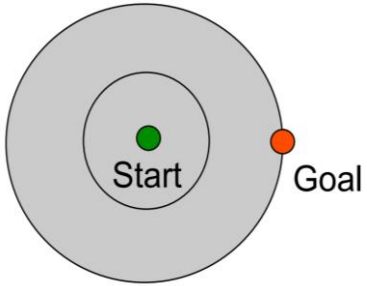


The graph generated by probabilistic roadmap (PRM)

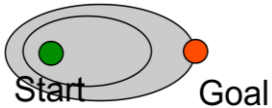
Search-based Method

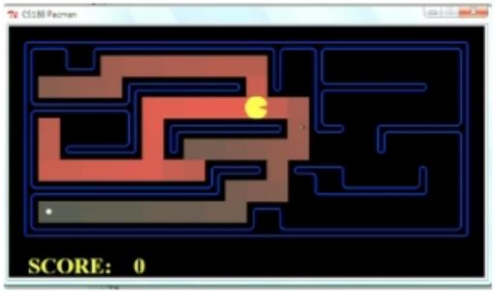
Dijkstra’ s vs. A*

- Dijkstra’s algorithm expanded in all directions



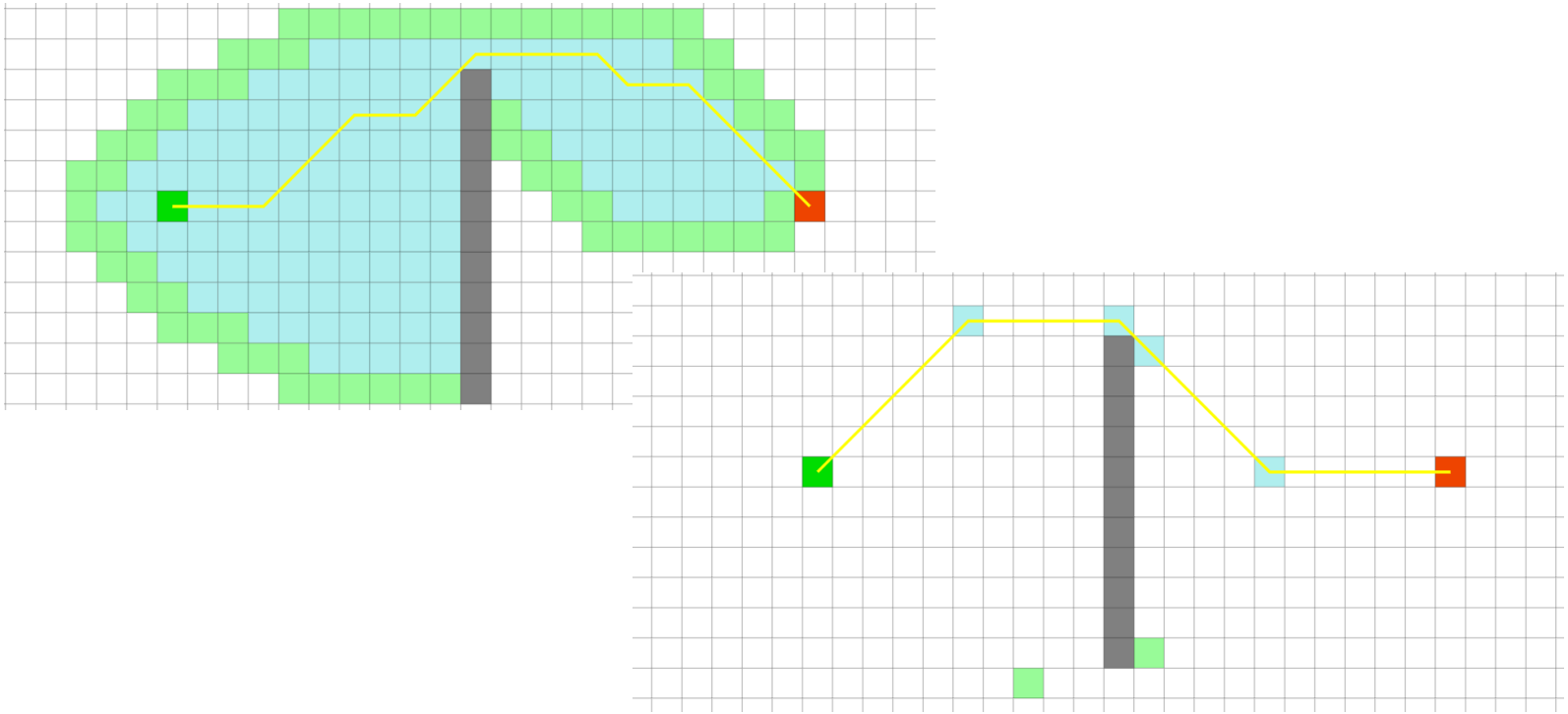
- A* expands mainly towards the goal, but does not hedge its bets to ensure optimality





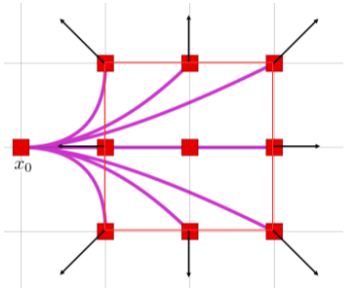
Search-based Method

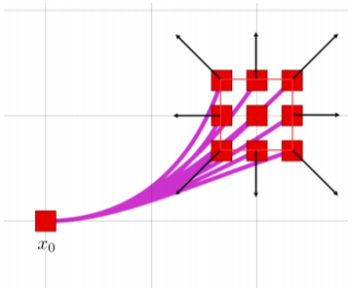
A* vs. JPS

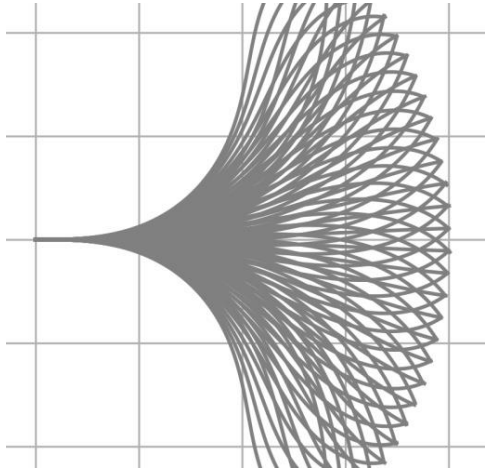


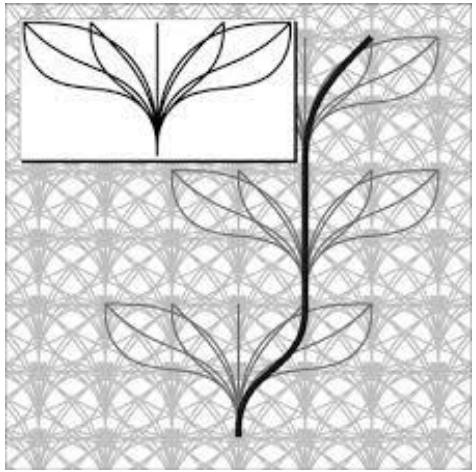
Kinodynamic Path Finding

State Lattice Search



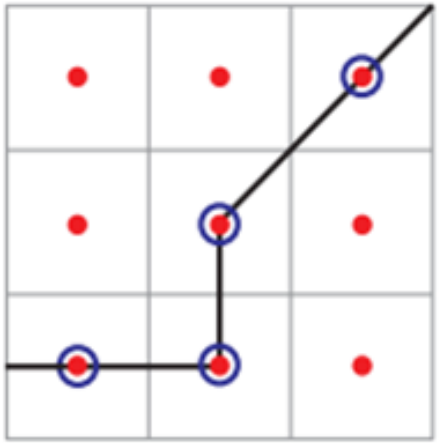


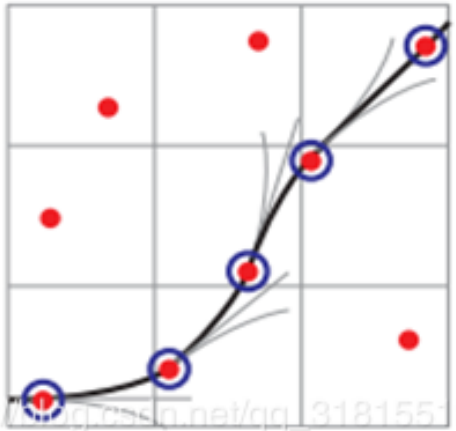




Kinodynamic Path Finding

Hybrid A*





1.Follow A* algorithm

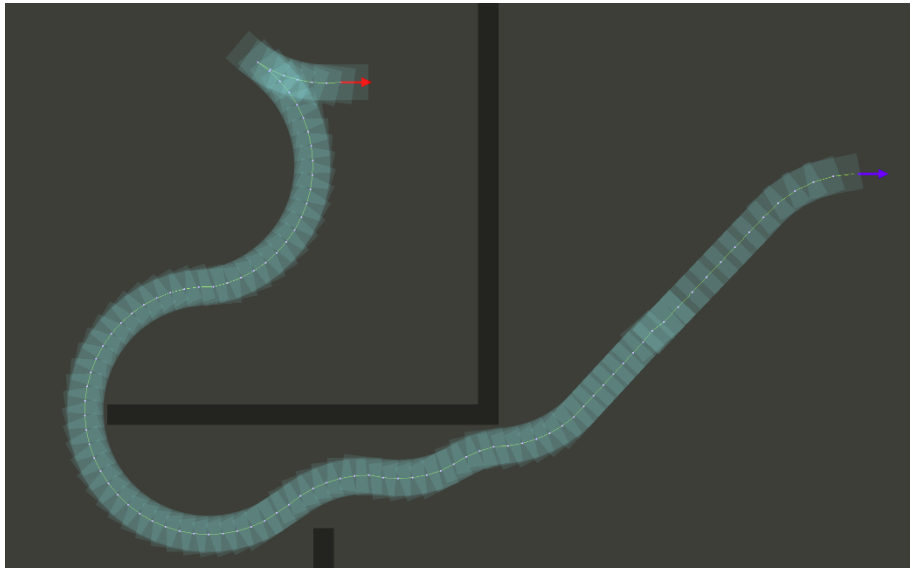
2. Forward simulate states with different discrete control inputs

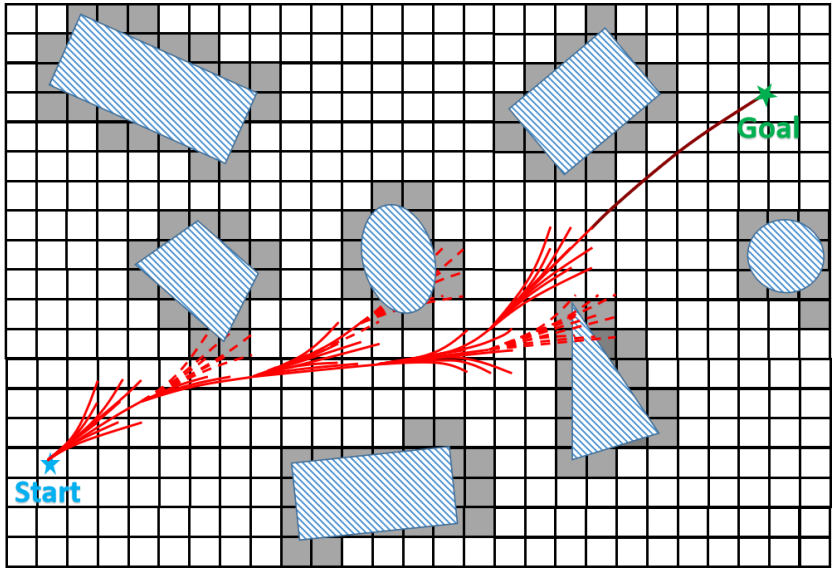
3. Keep only 1 state in each grid

Discrete control

Kinodynamic Path Finding

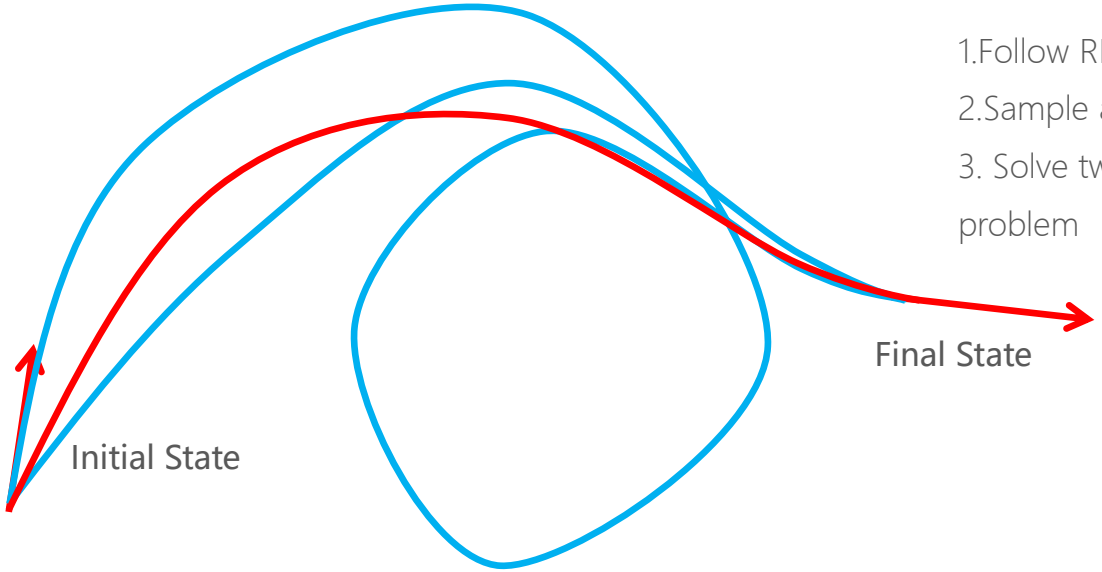
Hybrid A*





Kinodynamic Path Finding

Kinodynamic RRT*



1.Follow RRT* algorithm

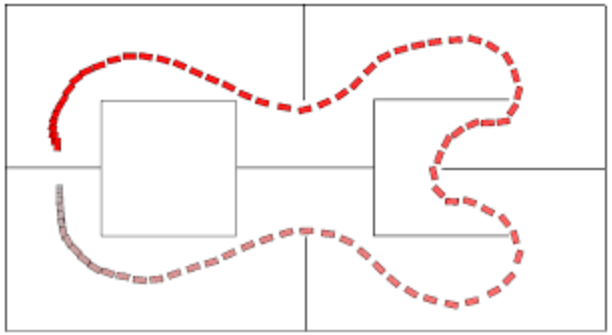
2.Sample a random state

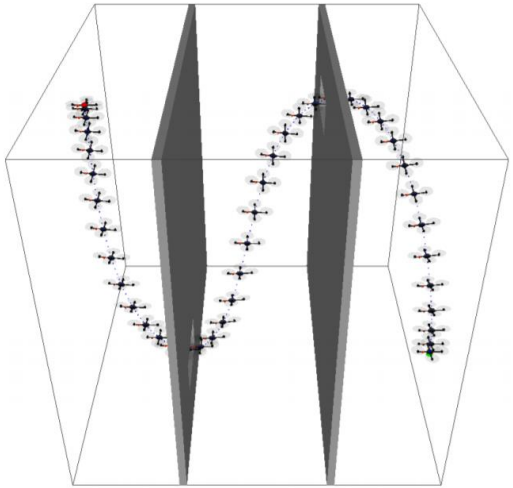
3. Solve two state boundary optimal control problem

Discrete state

Kinodynamic Path Finding

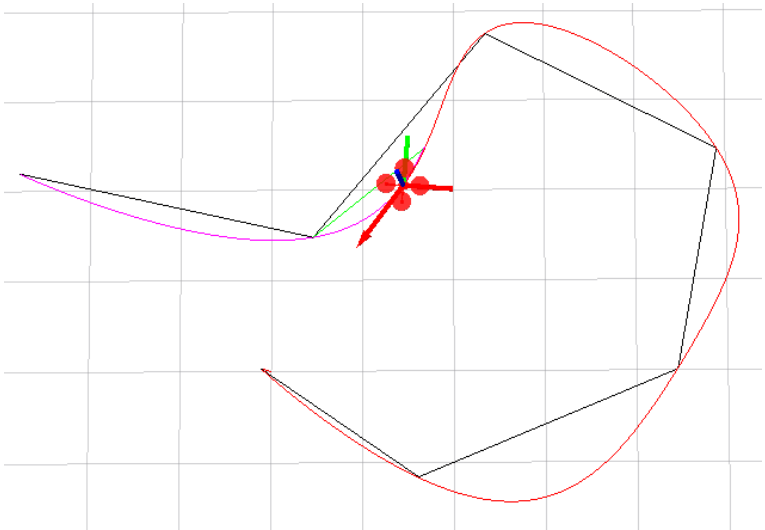
Kinodynamic RRT*

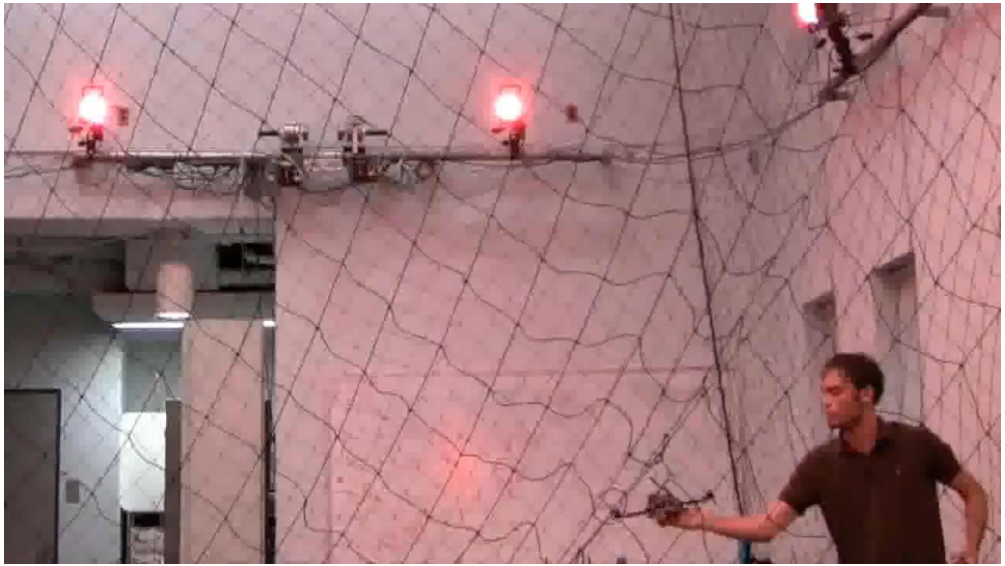




Back-end: Trajectory Optimization

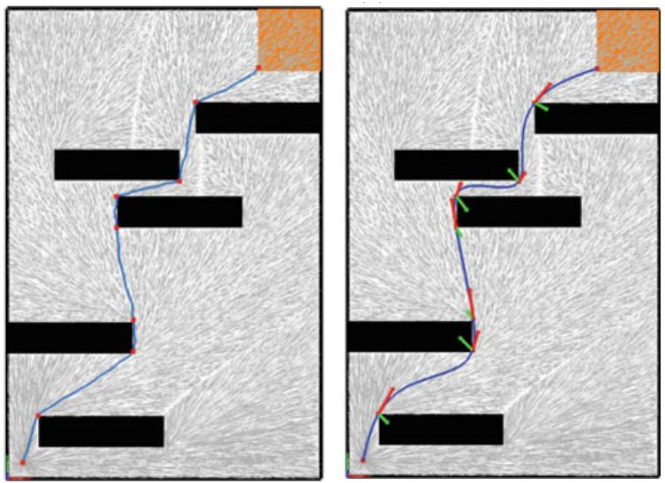
Basic Minimum-snap

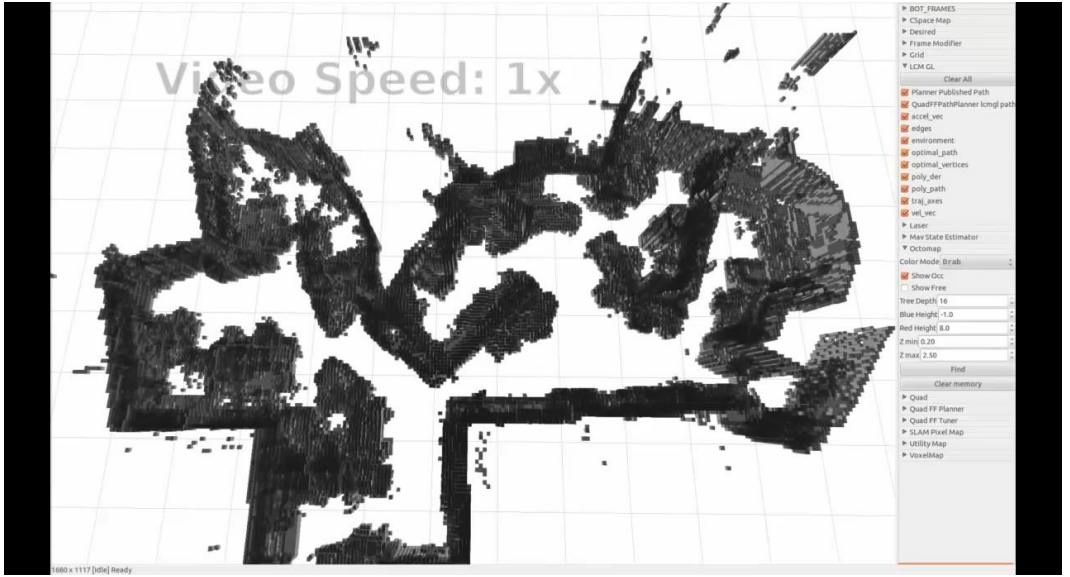




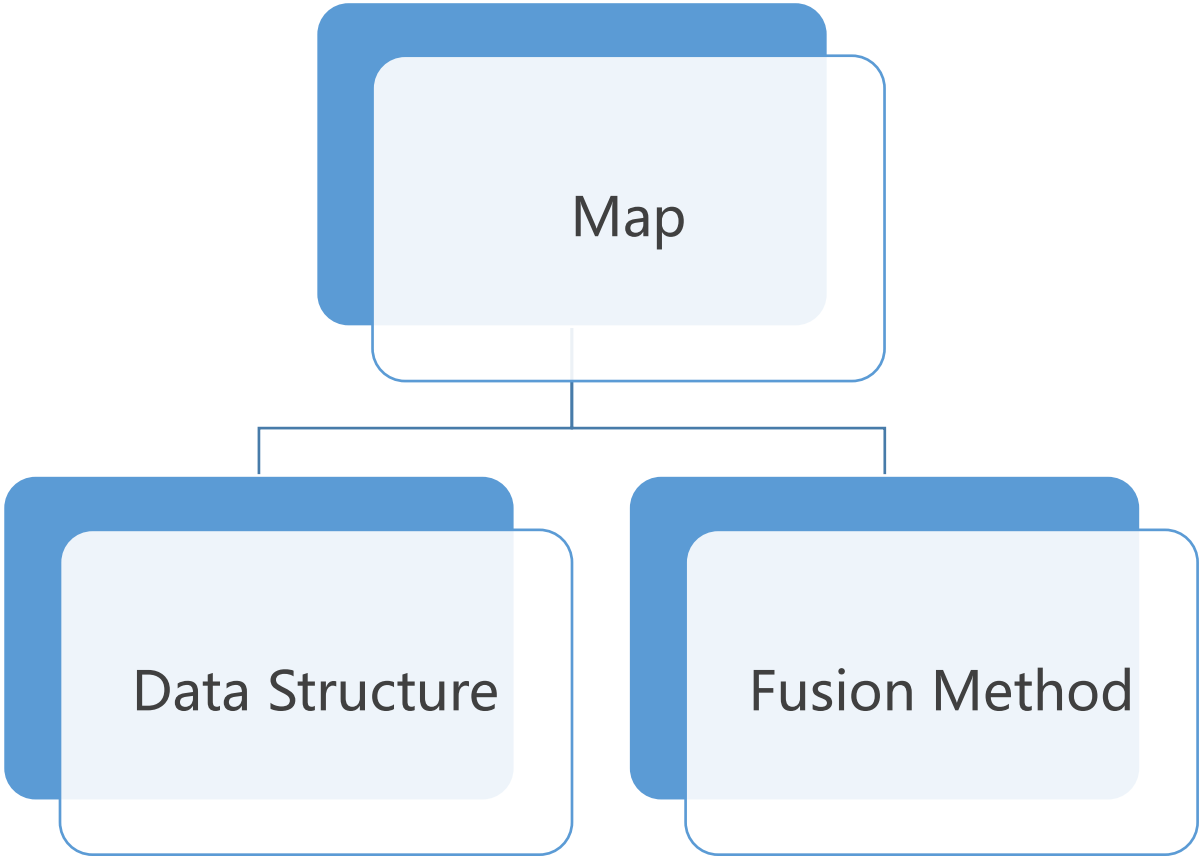
Back-end: Trajectory Optimization

Basic Minimum-snap

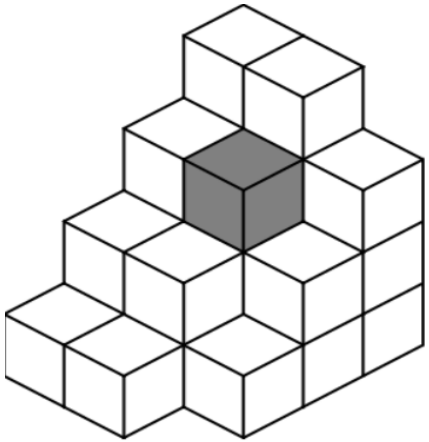


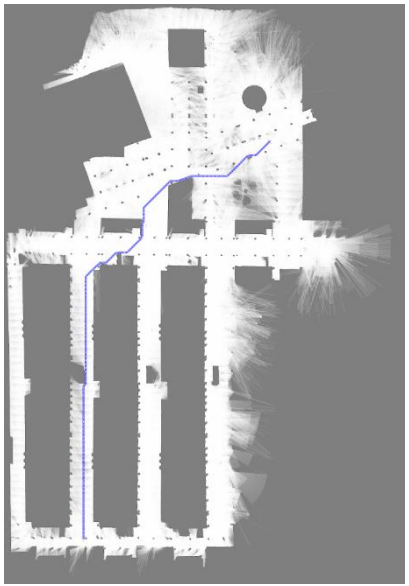


Map Representation

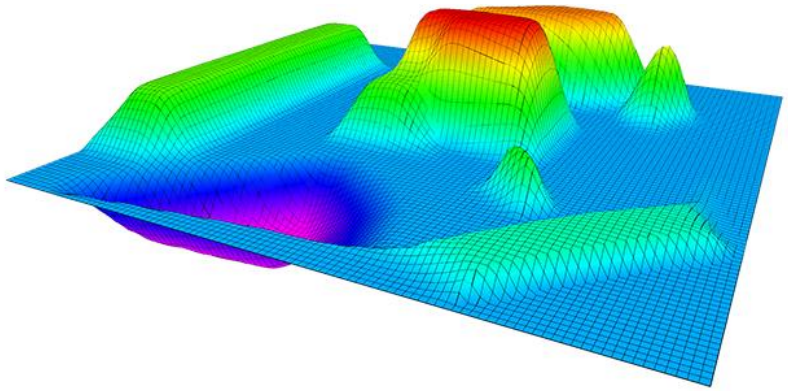

Map
Data Structure
Fusion Method

Occupancy grid map



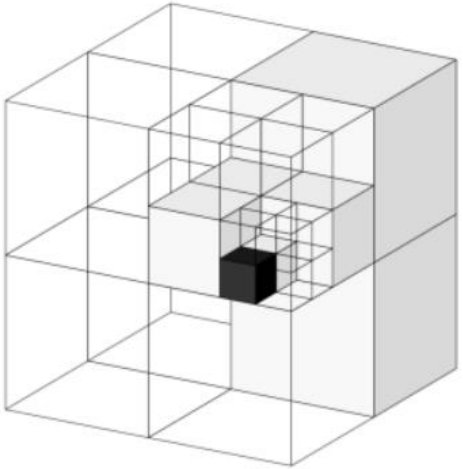


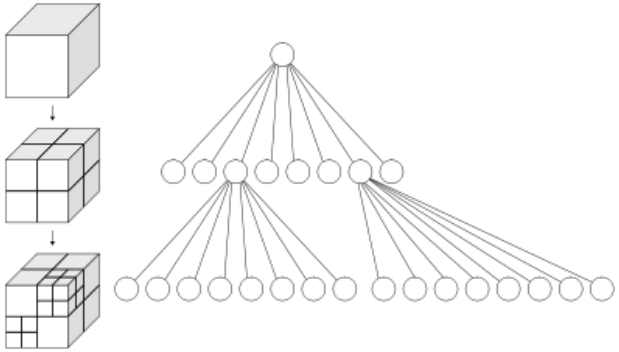
- Most Dense
- Structural
- Direct Index Query



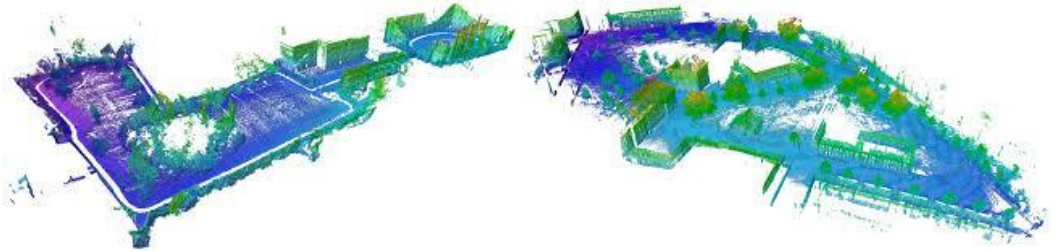
https://github.com/ANYbotics/grid_map

Octo-map



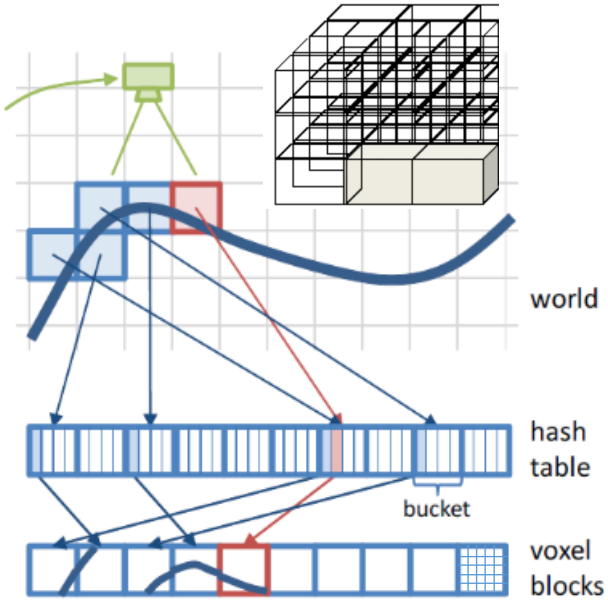


- Sparse
- Structural
- Indirect Index Query

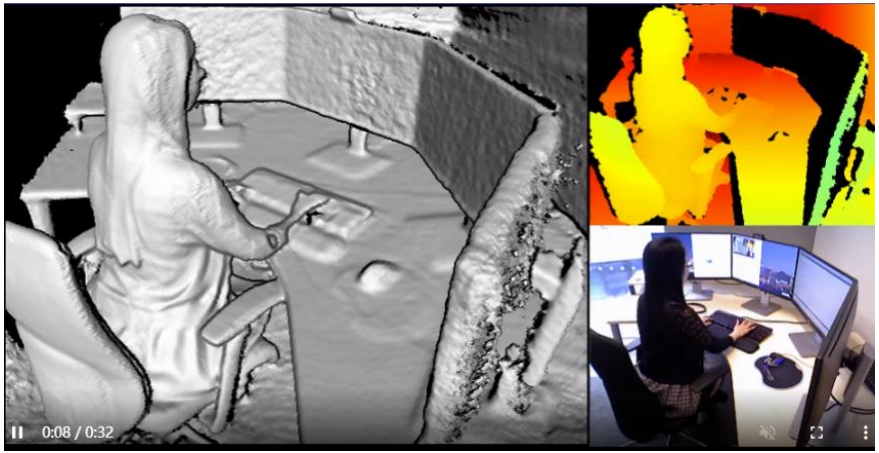


https://octomap.github.io/

Voxel hashing



- Most Sparse
- Structural
- Indirect Index Query



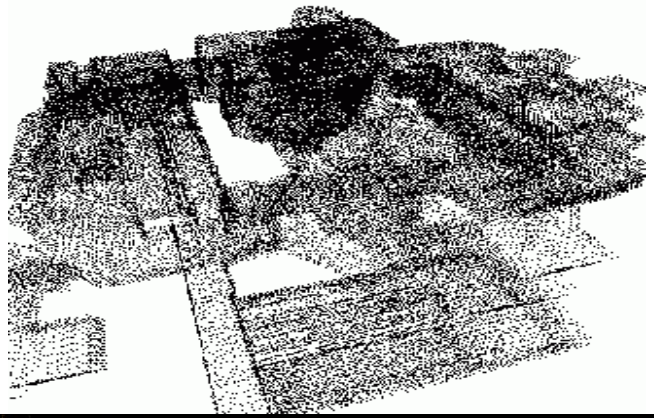
Voxel Hashing:

https://github.com/niessner/VoxelHashing

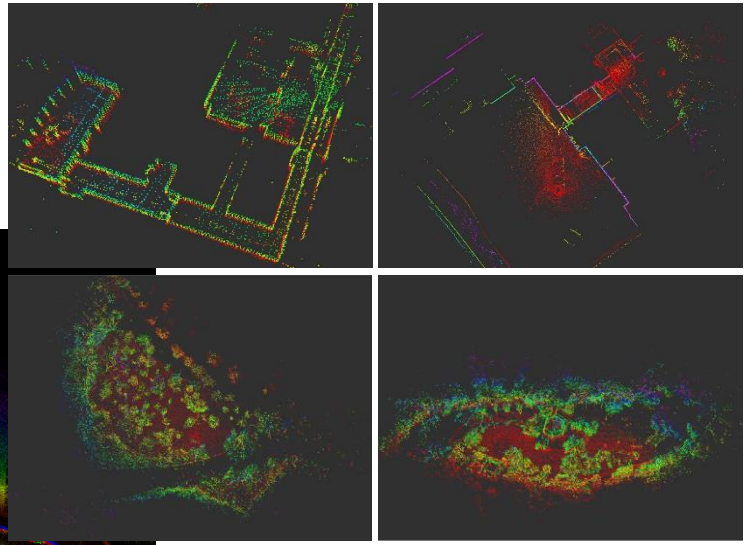
InfiniTAM:

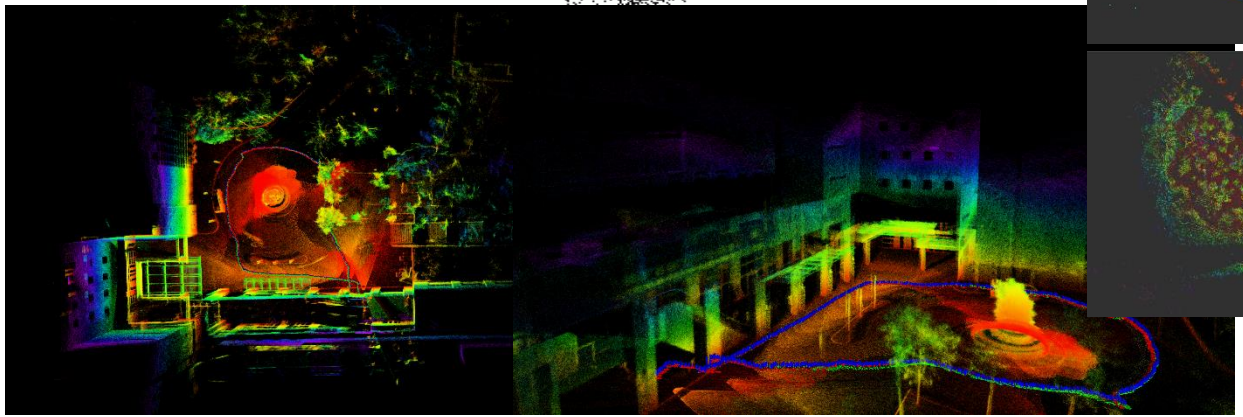
http://www.robots.ox.ac.uk/~victor/infinitam/

Point cloud map



- Un-ordered
- No Index Query

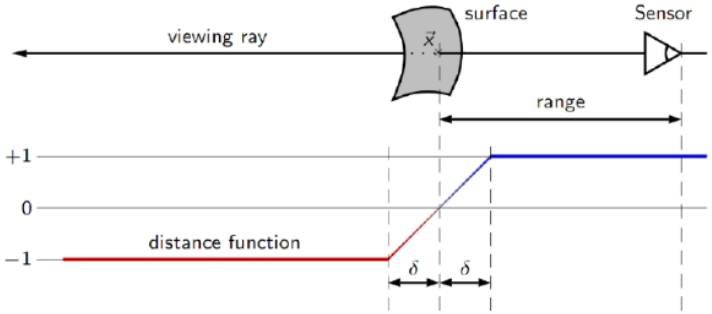


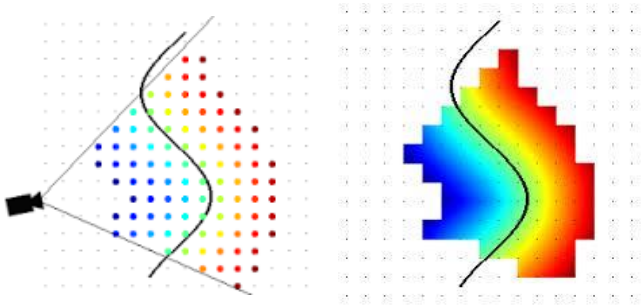


PCL
http://pointclouds.org/

TSDF map

Truncated Signed Distance Functions





OpenChisel

https://github.com/personalrobotics/OpenChisel

ESDF map

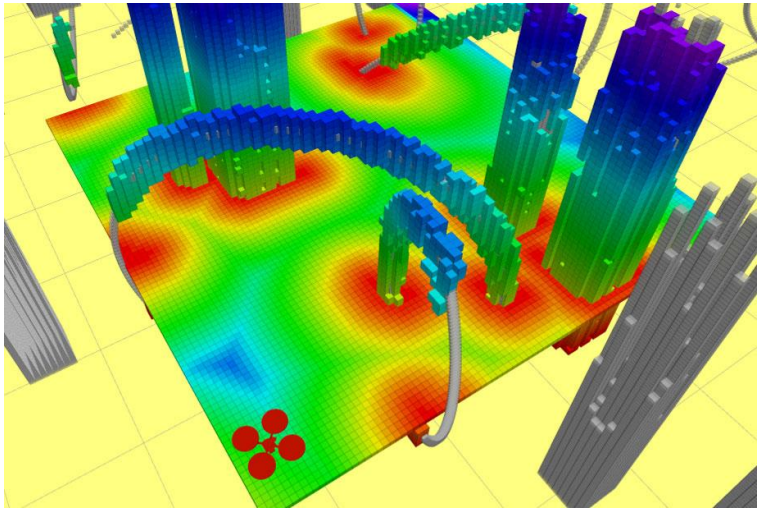
Euclidean Signed Distance Functions

Incremental Update, Global Map

Running the Cow_and_Lady Dataset[1]
Compare with Voxblox[2]

[1] https://projects.asl.ethz.ch/datasets/doku.php?id=iros2017/
[2] Helen Oleynikova, Zachary Taylor, Marius Fehr, Juan Nieto, and Roland Siegwart, "Voxblox: Building 3D Signed Distance Fields for Planning", In IEEE Int. Conf. on Intelligent Robots and Systems (IROS), October 2017.

Batch Update, Local Map



Distance Transforms of Sampled Functions, PF Felzenszwalb

VoxBlox

https://github.com/ethz-asl/voxblox

FIESTA

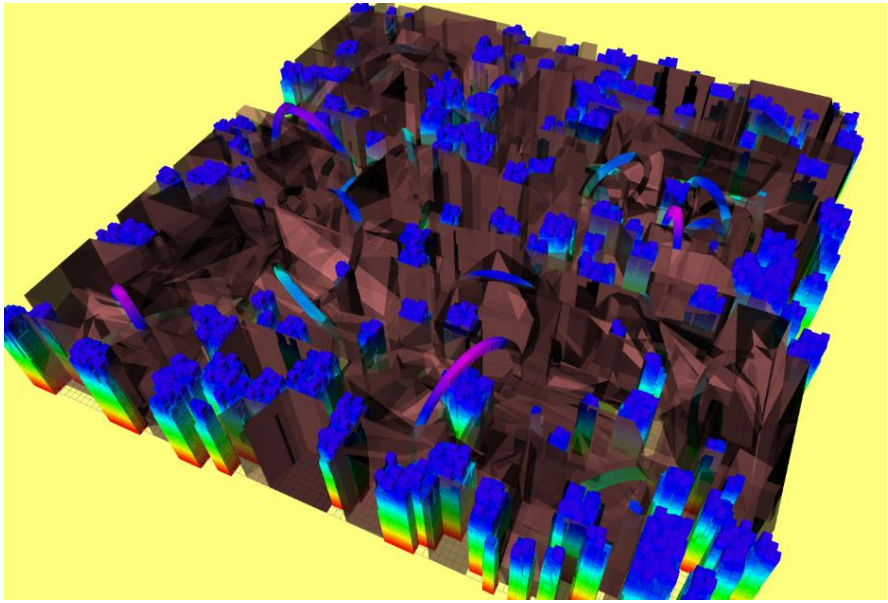
https://github.com/HKUST-Aerial-Robotics/FIESTA

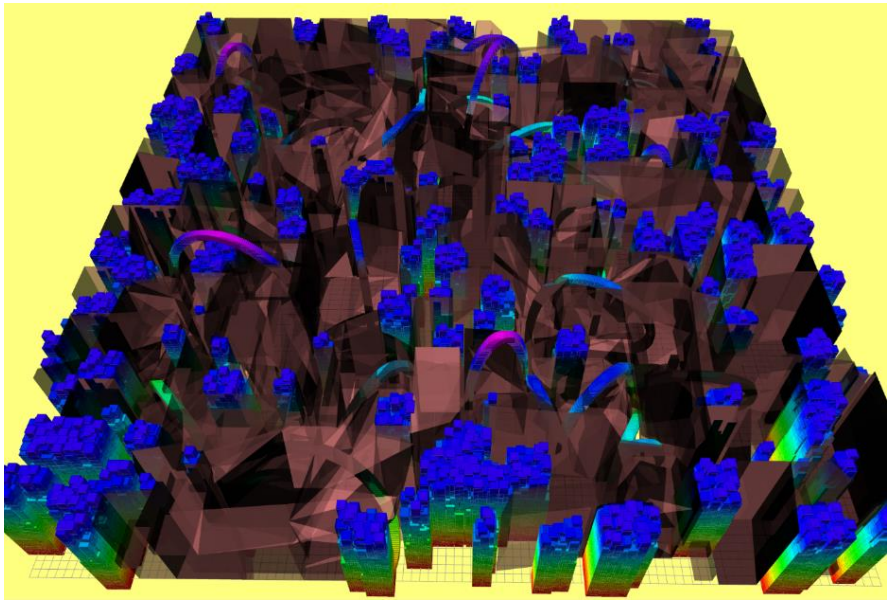
TRR's Local Map

https://github.com/HKUST-Aerial-Robotics/Teach-Repeat-Replan

More ?

Free-space Roadmap

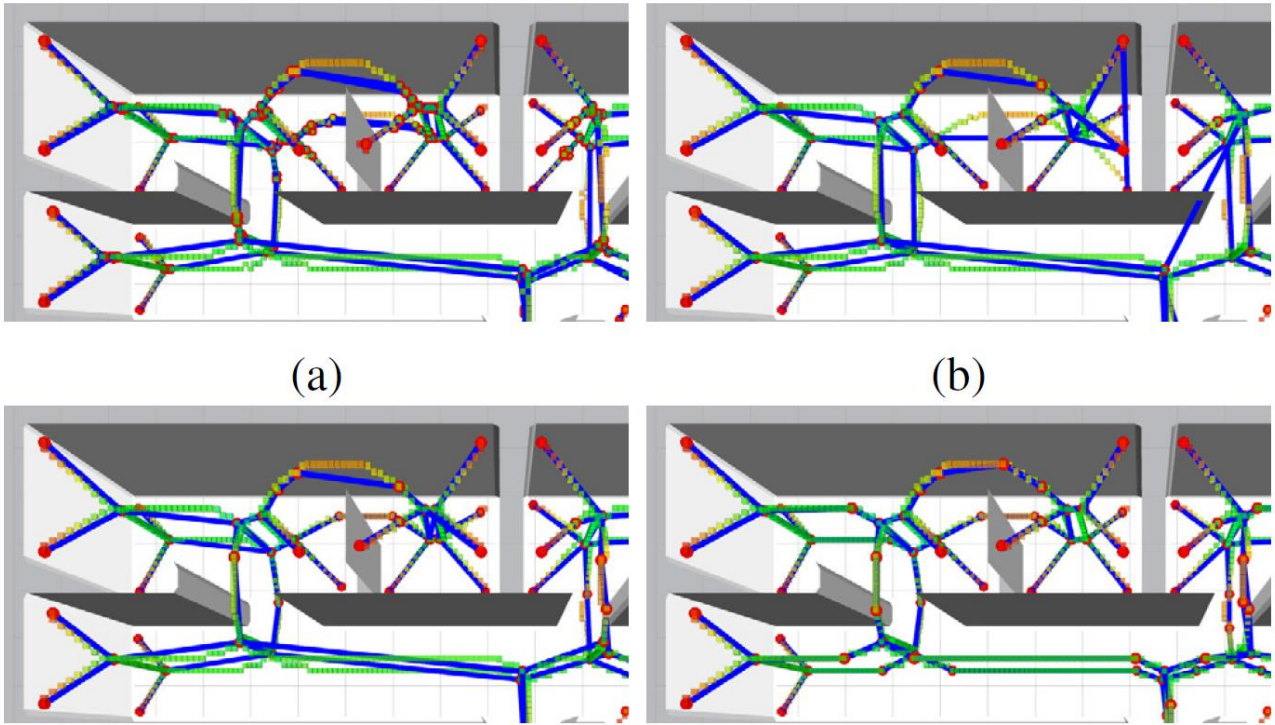




https://github.com/HKUST-Aerial-Robotics/Teach-Repeat-Replan

More ?

Voronoi Diagram Map



https://github.com/ethz-asl/mav_voxblox_planning

Thanks for Listening!