Motion Planning for Mobile Robots
机器人路径规划

主讲人 常 晶

Email: jchang@xidian.edu.cn

目录

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 may also cover:
  - Autonomy for mobile robots
  - Paper recommendations
  - Cutting-edge research direction/ethics

- 理解什么是路径规划和搜索
- 基于搜索、采样等的路径规划
- 轨迹生成/优化、MPC
- 实时软件实现

Basic Expectation

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

基 础

线性代数、概率论
Matlab /C++编程

- Love robots ☺

教 材

《机器人学、机器视觉与控制——MATLAB算法基础》, Peter Corke著，刘荣等译，电子工业出版社，2016.

参考书目

1. 《Robotics,Vision and Control. Fundamental Algorithms in MATLAB》Peter Corke, Springer, 2017.
2. 《Principles of Robot Motion: Theory, Algorithms, and Implementations》, Howie Choset et. al. The MIT Press, 2005.

考核方式

- 平时作业成绩：40%
- 随堂测试成绩：10%
- 课程设计成绩：50%。主要考核算法实践动手能力及知识点的理解程度，文字和口头表达能力，分工协作能力。每组选一个代表进行5分钟汇报并接受5分钟提问，个人提交一份实验报告。在50%中，分组口头报告占比10%，源代码正确性验证占比20%，个人实验报告20%。

- Lots of project work

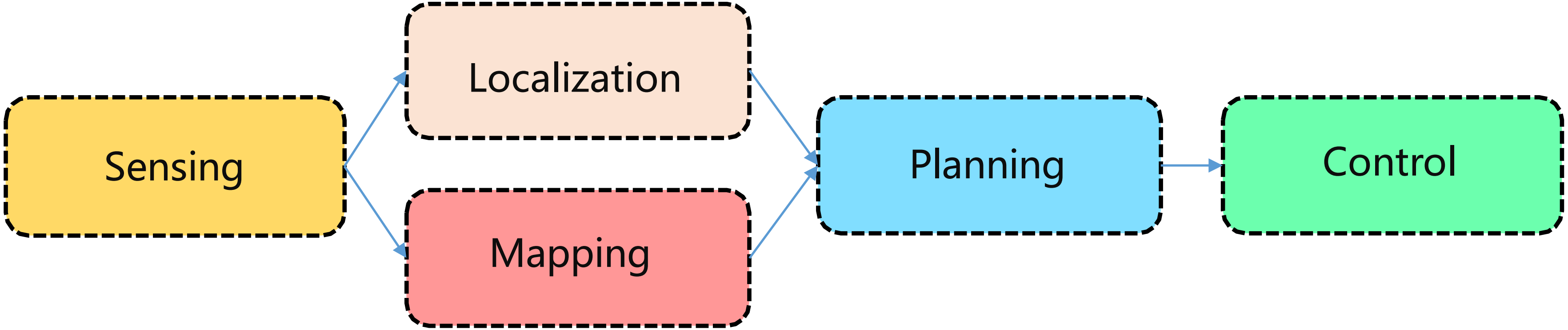
- Have fun with robots ☺

What is autonomous robot

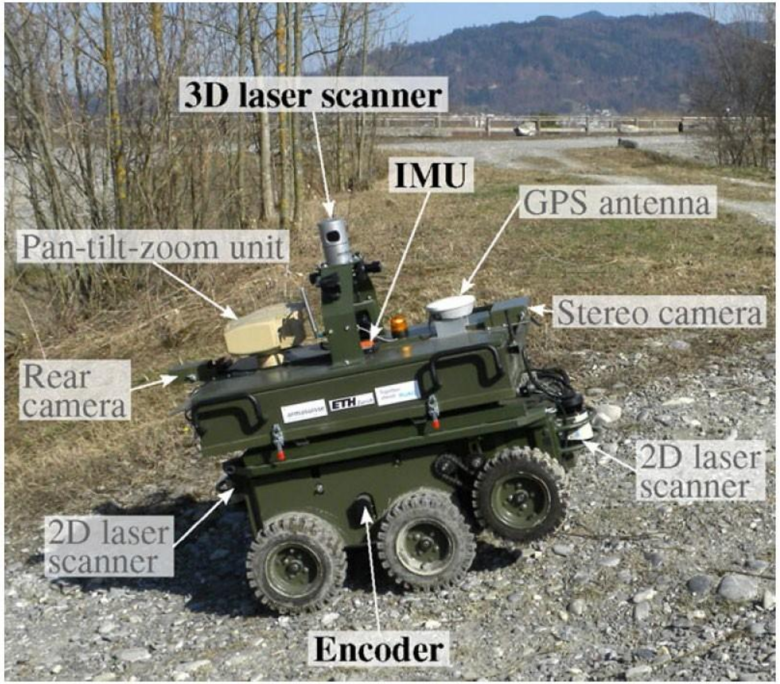
Definition: an autonomous robot is a robot that performs behaviors or tasks with a high degree of autonomy (without external influence).

Subjects: Computer Science, Automation, Mechanism, Electronics ...

- Perception-Planning-Control action loop



Autonomous robot: applications



Autonomous robot: applications

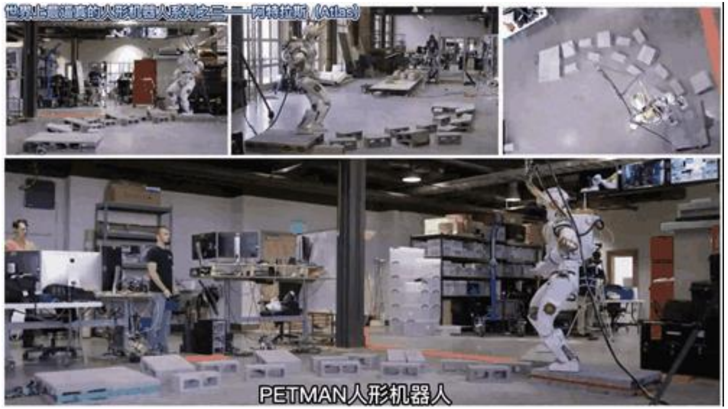
仿生机器

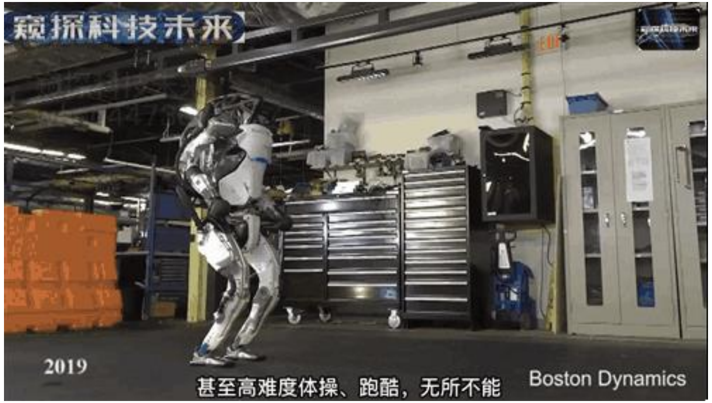
物流机器

智能家居

服务型机器人（情感交互）

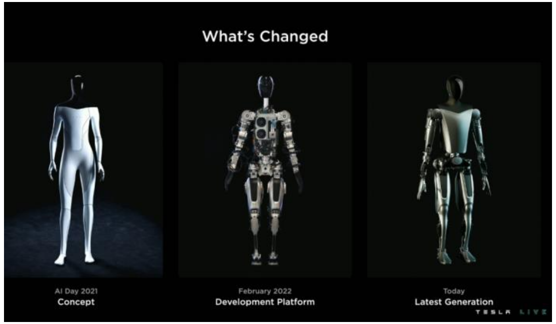
Autonomous robot: applications





2017年11月16日，波士顿动力发布了自己的Atlas 机器人，可以灵活跳跃、实现后空翻的高难度动作。这一突破性成果适用了基于智能控制论的技术，实现了类似人体应激反应的平衡控制效果。

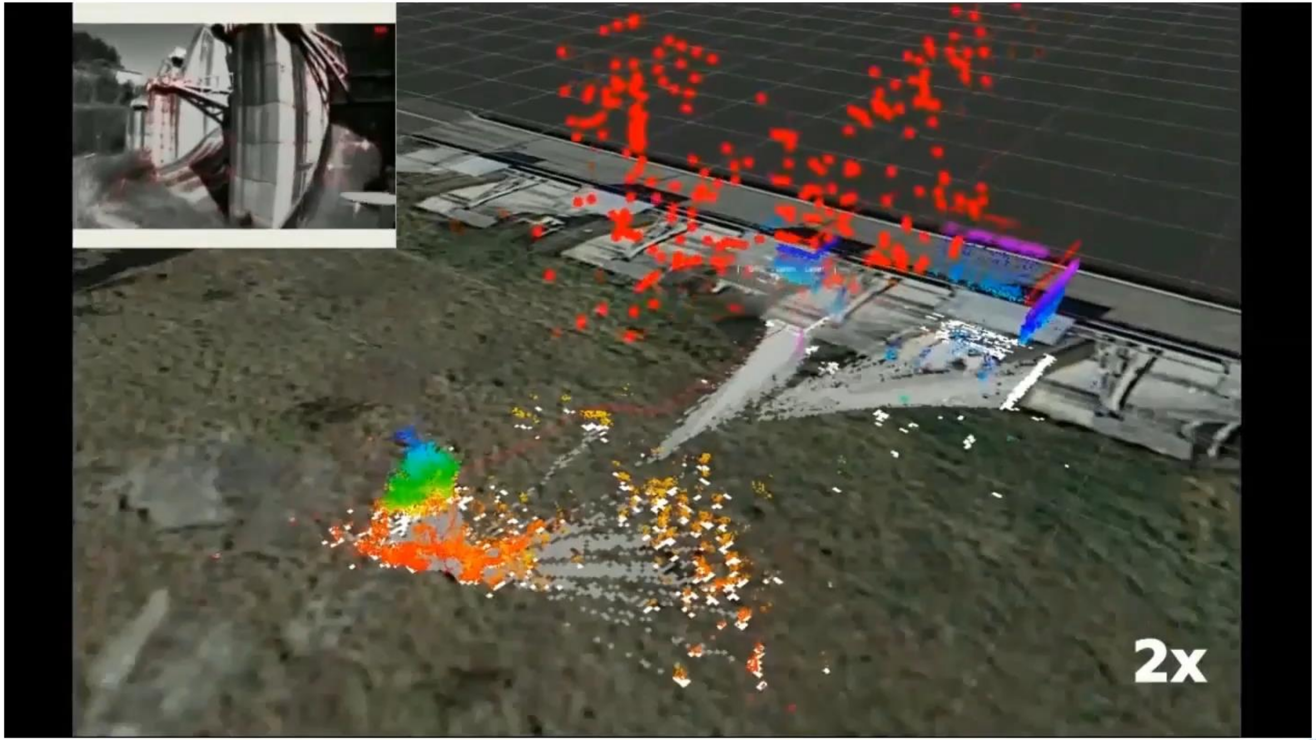
Autonomous robot: applications



2022年9月30日,马斯克展示了人形机器人“擎天柱”(Optimus)。

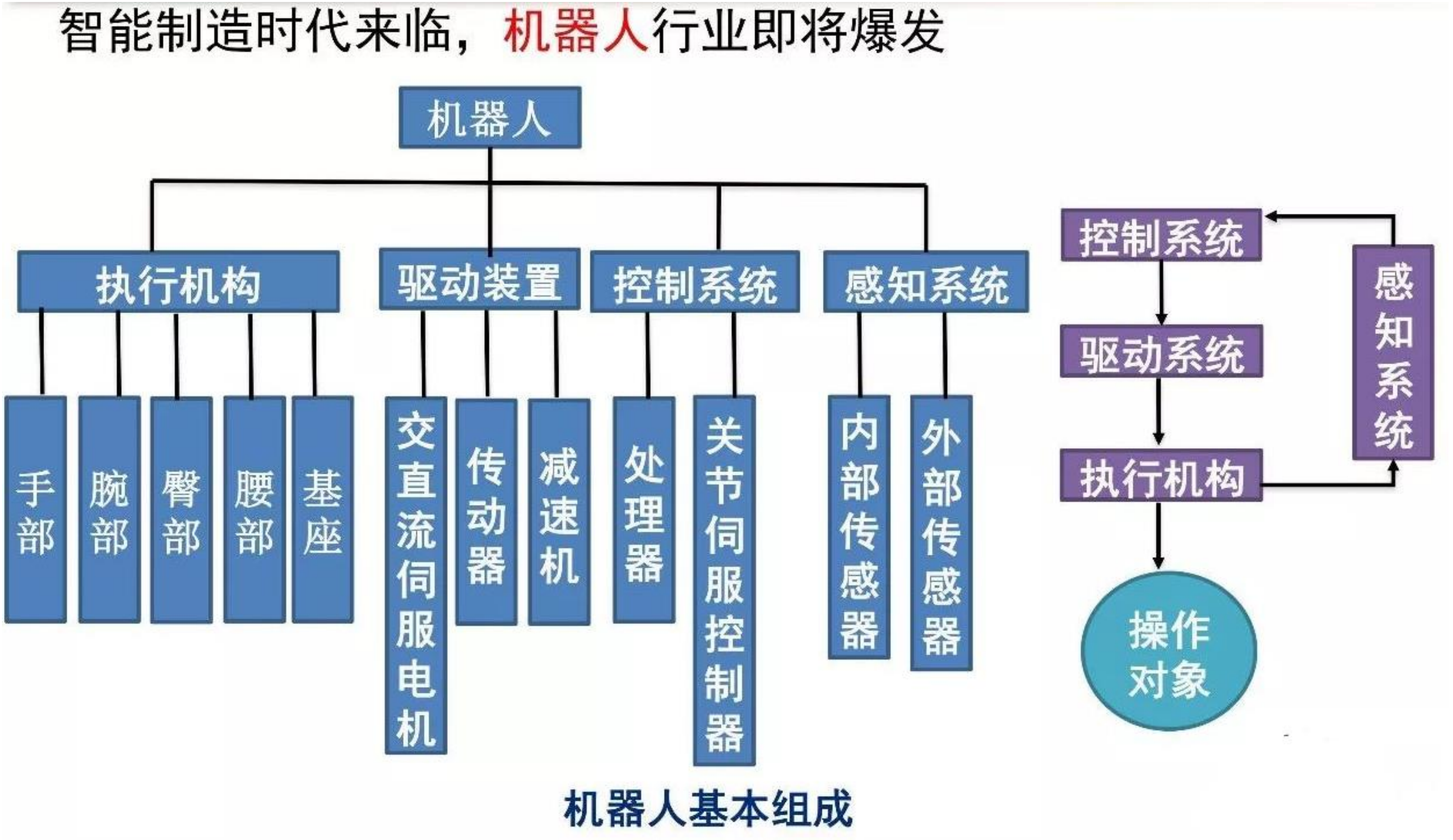
Autonomous robot: applications

Dam Inspection



Shen, et al, 2014

Autonomous robot: applications



What is autonomous robot

- **Estimation** 估计
  - Low latency
  - High accuracy & consistency

- **Perception** 感知
  - 3D sensing & dense perception
  - Map fusion & integration for planning

- **Planning** 规划
  - Complex & unknown environments
  - Safety & dynamical feasibility
  - Limited sensing & computation

- **Control** 控制
  - 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.

Groups and Researchers

University of Pennsylvania

- GRASP Lab,Vijay Kumar
  Research Interests: planning, control, swarm
  Homepage: www.kumarrobotics.org

Massachusetts Institute of Technology

- Jonathan How
  Research Interests: modelling, control, planning
  Homepage: www.mit.edu/~jhow

- Nicholas Roy
  Research Interests: perception, learning
  Homepage: groups.csail.mit.edu/rrg

Carnegie Mellon University

- Nathan Michael
  Homepage: www.rislab.org

- Sebastian Scherer
  Research Interests: perception, planning
  Homepage: theairlab.org

University of California, Berkeley

- Markus Mueller
  Research Interests: control, planning

ETH Zurich

- ASL Team, Roland Siegwart
  Research Interests: perception, control
  Homepage: asl.ethz.ch

- Raffaello D'Andrea
  Research Interests : control, swarm
  Homepage: raffaello.name

University of Zurich

- Davide Scaramuzza
  Research Interests : perception, control
  Homepage: [rpg.ifi.uzh.ch](rpg.ifi.uzh.ch)

Hong Kong University of Science Technology

- Shaojie Shen
  Research Interests : UAV
  Homepage: [uav.ust.hk](uav.ust.hk)

- Ming Liu
  Research Interests : UGV
  Homepage: [ram-lab.com](ram-lab.com)

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
- SOFT AND HARD CONSTRAINED TRAJECTORY OPTIMIZATION
  - Soft Constrained Trajectory Optimization
  - Hard Constrained Trajectory Optimization
  - Homework

MDP & MPC

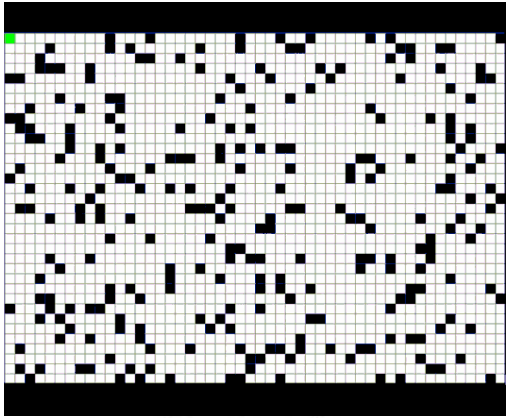
- MARKOV DECISION PROCESS-BASED PLANNING
  - Uncertainties in Planning and MDP
  - Minimax Cost Planning and Expected Cost Minimal Planning
  - Value Iteration and Real-Time Dynamic Programming
  - Homework

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

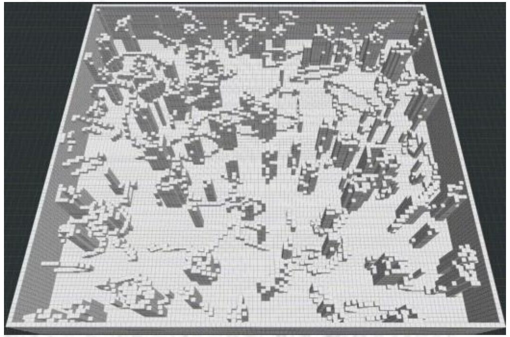
Overview

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



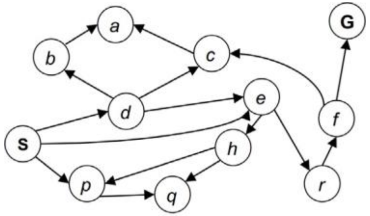
A* search example



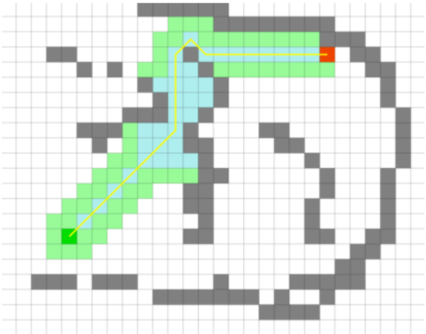
RRT example

Search-based Method

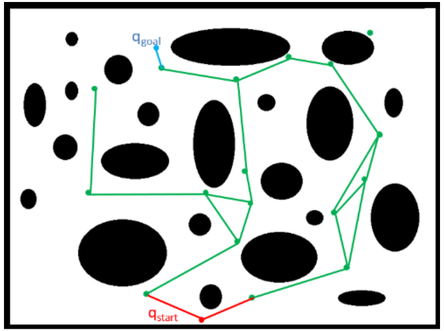
- For every search problem, there's a corresponding state space graph
- Connectivity between nodes in the graph is represented by (directed or undirected) edges



Ridiculously tiny search graph for a tiny search problem



Grid-based graph: use grid as verticesand grid connections as edges

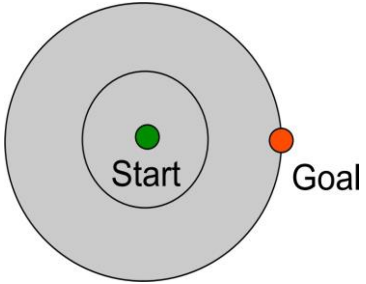


The graph generated by probabilisticroadmap (PRM)

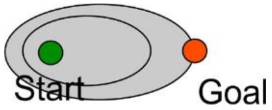
Search-based Method

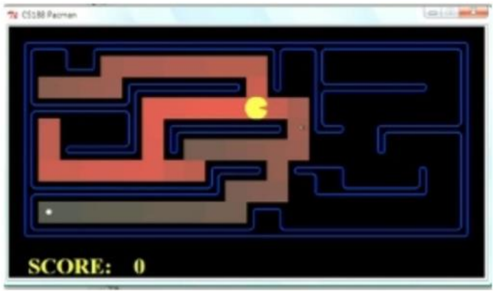
Dijkstra’s vs. A*

- Dijkstra’s algorithm expanded in all directions



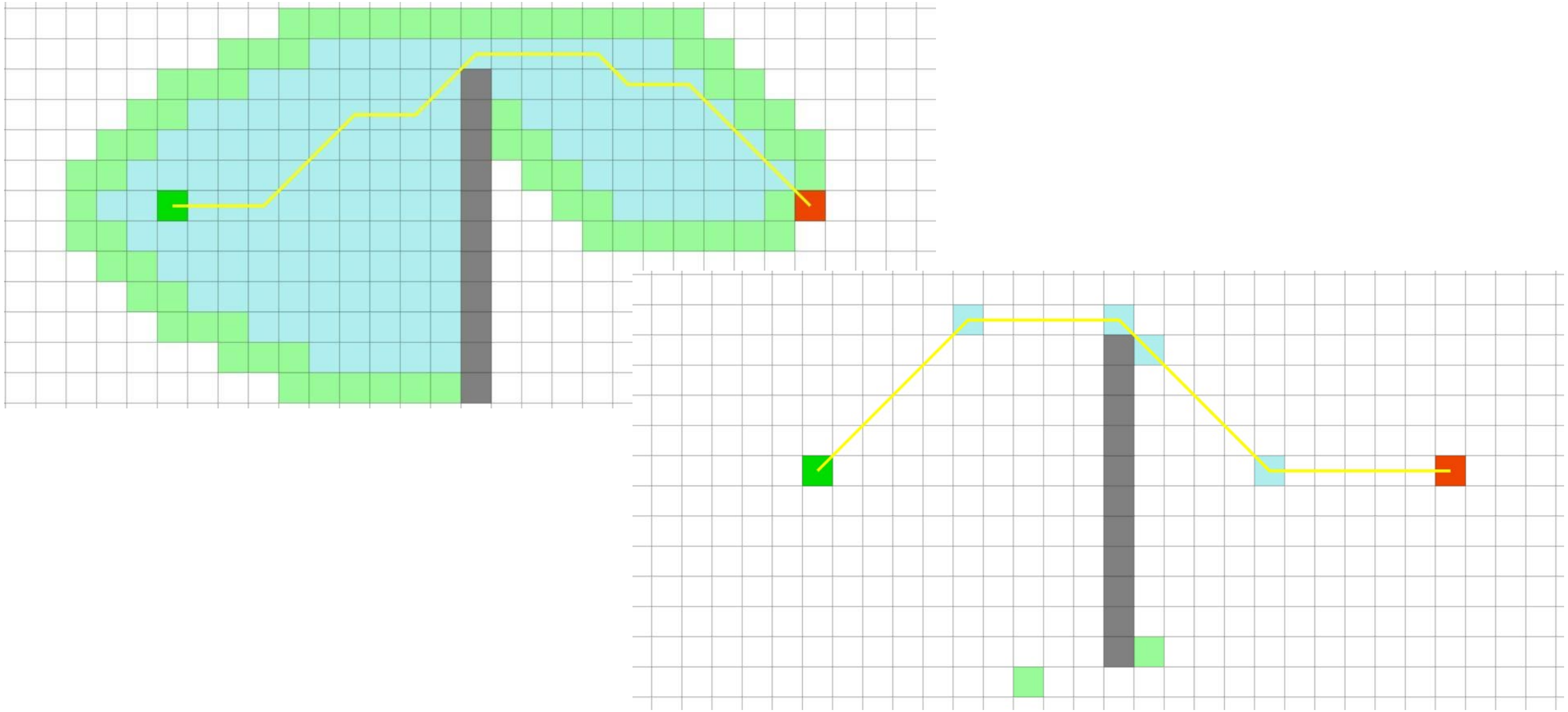
- A* expands mainly towards the goal, but does not hedge its bets to ensure optimality





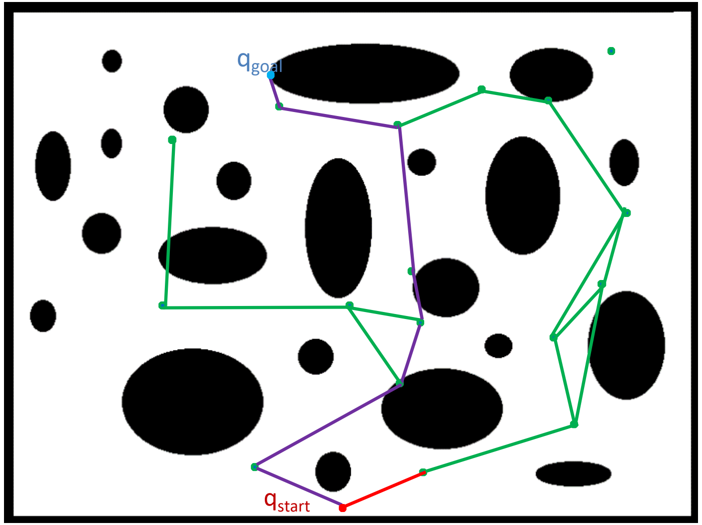
Search-based Method

A* vs. JPS



Sampling-based Method

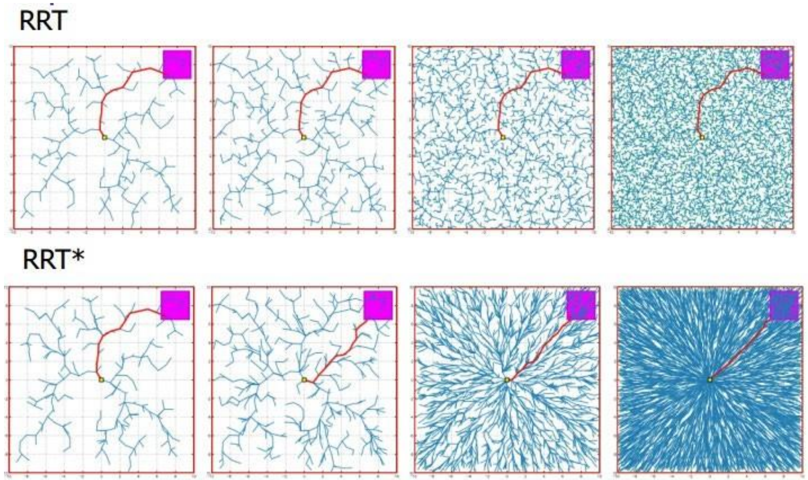
Probabilistic Roadmap (PRM)



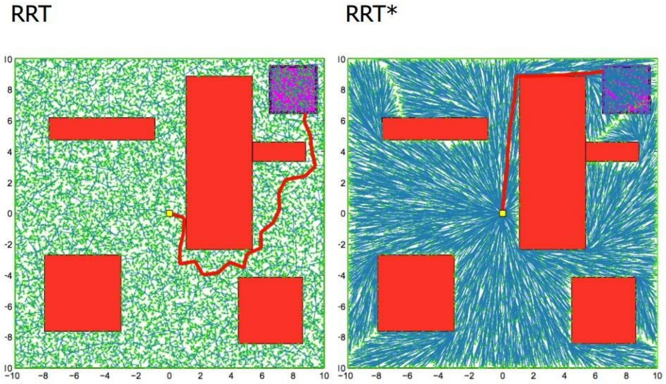
C-space

Sampling-based Method

RRT* vs RRT

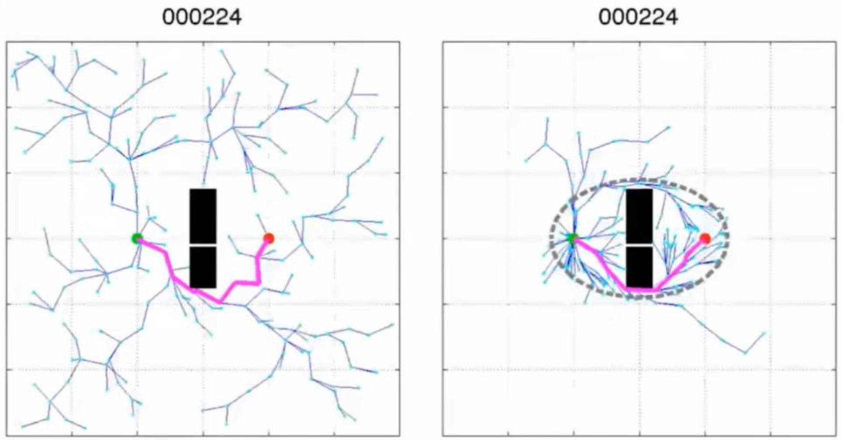


Source: Karaman and Frazzoli



Sampling-based Method

Informed RRT*

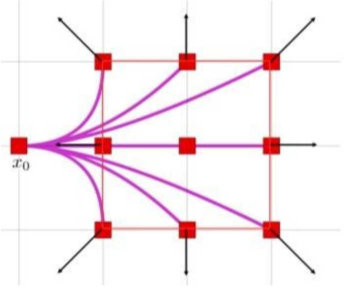


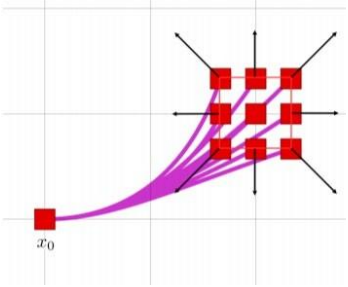
▷ RRT* ▷ Informed RRT*

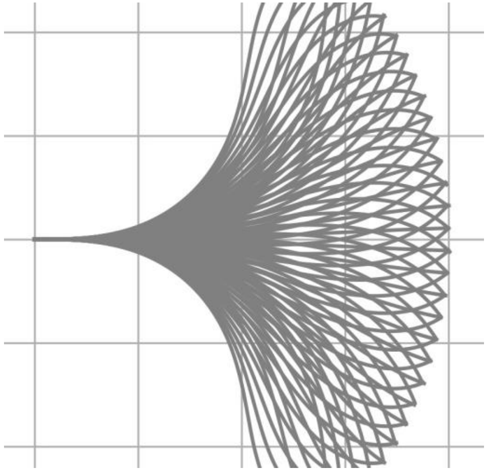
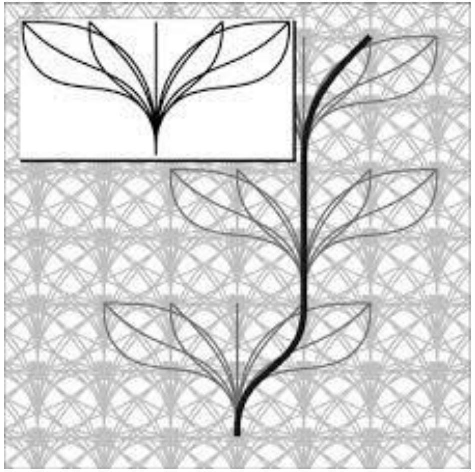
- In the following examples, RRT* and Informed RRT* were run until they found solutions of equivalent cost.
- Informed RRT* takes an **order of magnitude fewer iterations** than RRT* with no practical increase in computational cost.
- Informed RRT* makes **no simplifying assumptions or approximations** about the state space.

Kinodynamic Path Finding

State Lattice Search

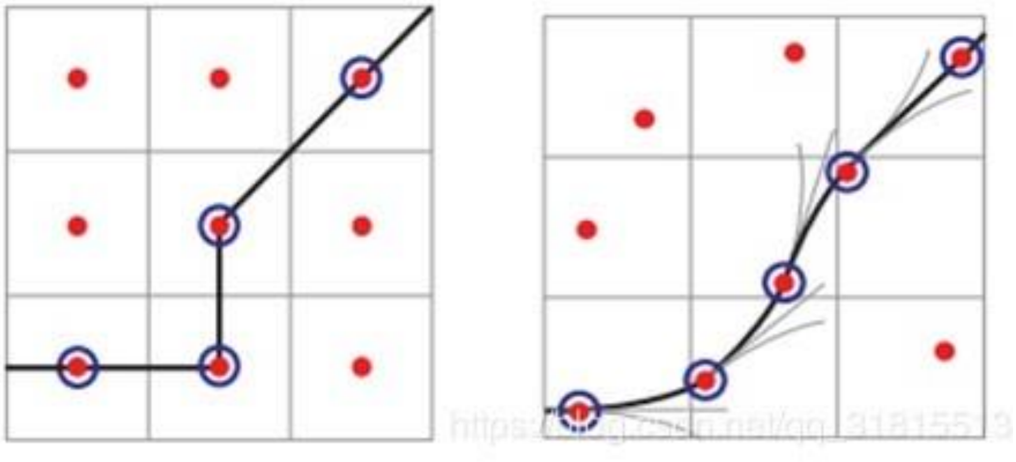




Kinodynamic Path Finding

Hybrid A*

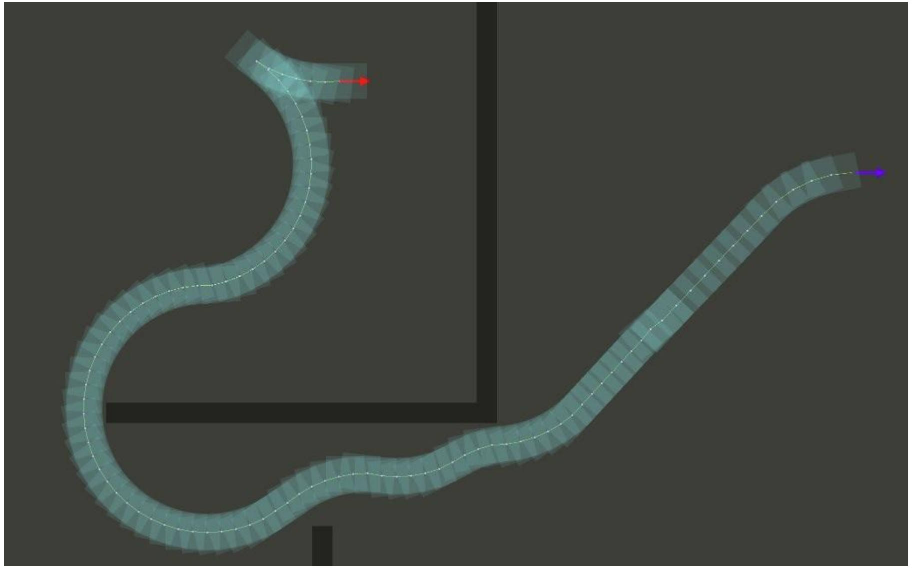


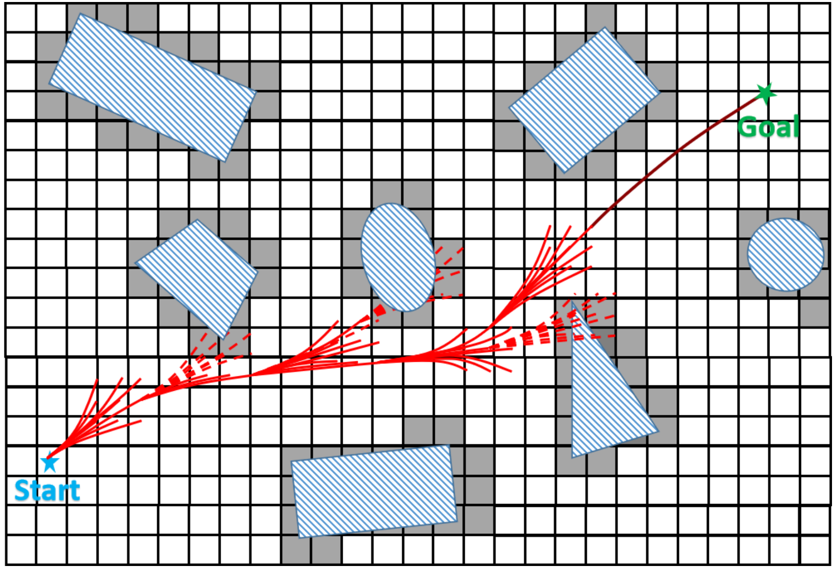
1.Follow A* algorithm
2.Forward simulate states with different discrete control inputs
3. Keep only 1 state in each grid

Discrete control

Kinodynamic Path Finding

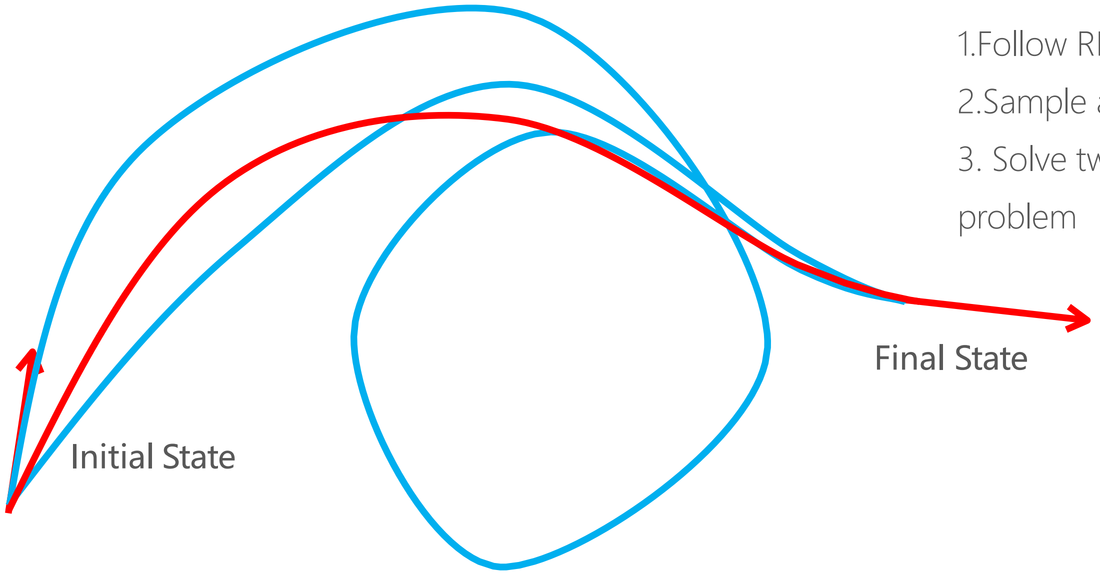
Hybrid A*





Kinodynamic Path Finding

Kinodynamic RRT*



1.Follow RRT* algorithm

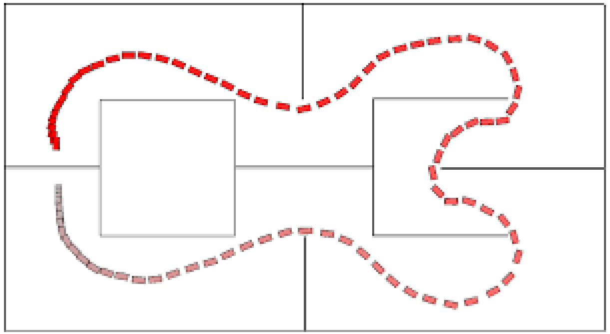
2.Sample a random state

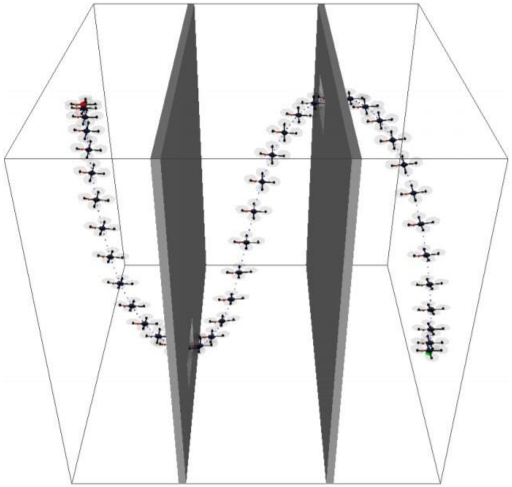
3. Solve two state boundary optimal control problem

Discrete state

Kinodynamic Path Finding

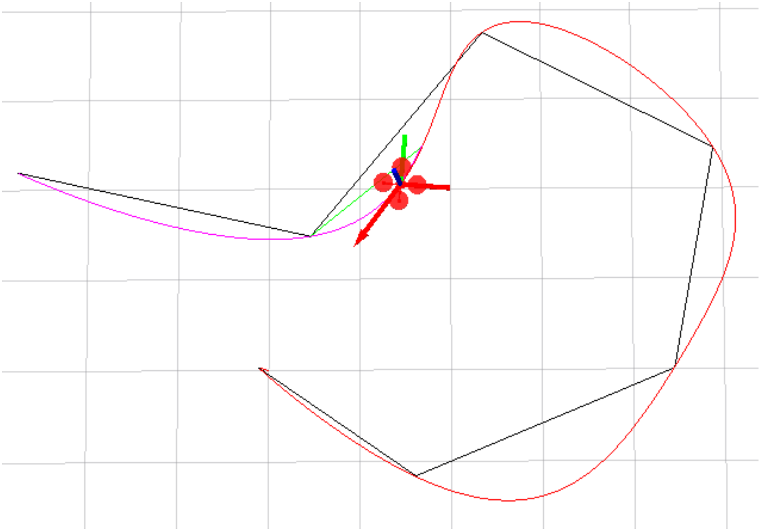
Kinodynamic RRT*





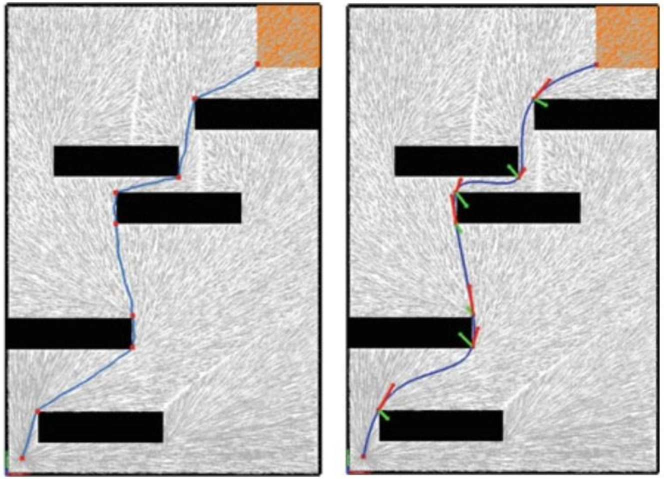
Back-end: Trajectory Optimization

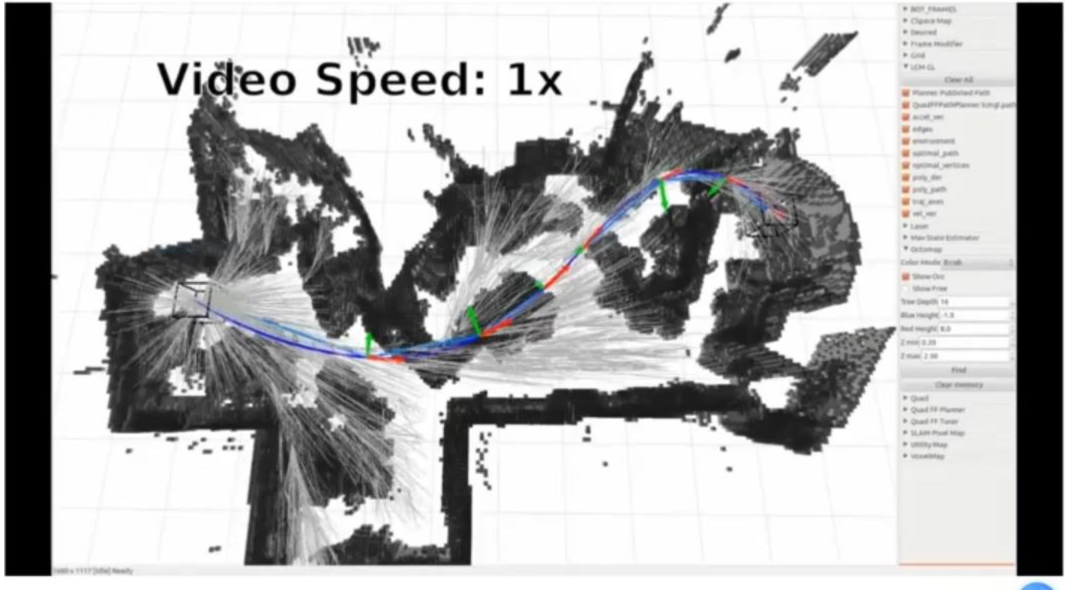
Basic Minimum-snap



Back-end: Trajectory Optimization

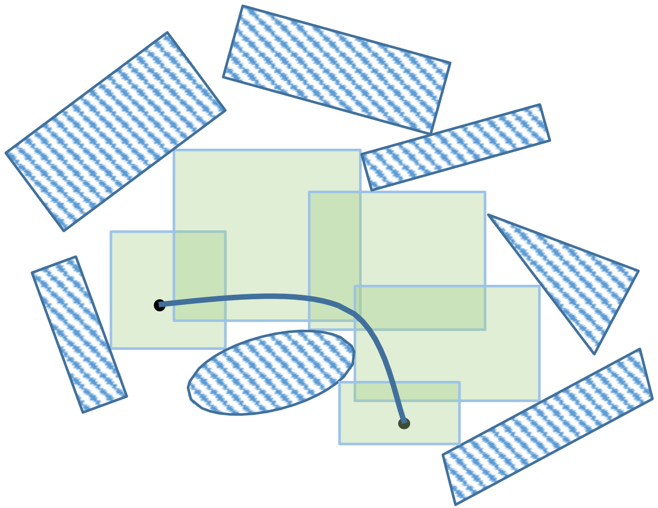
Basic Minimum-snap

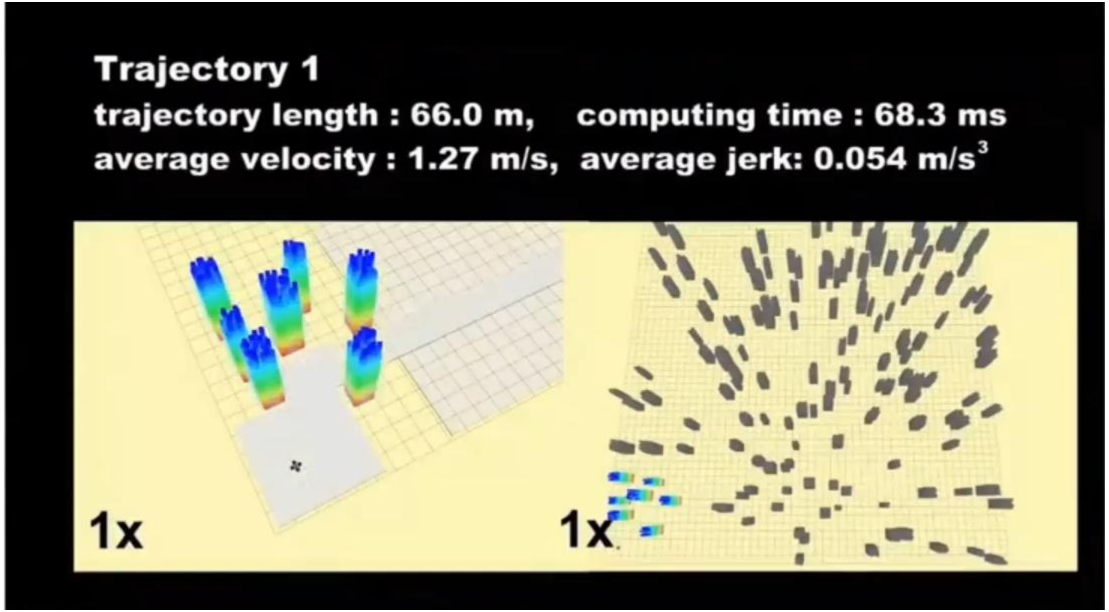




Back-end: Trajectory Optimization

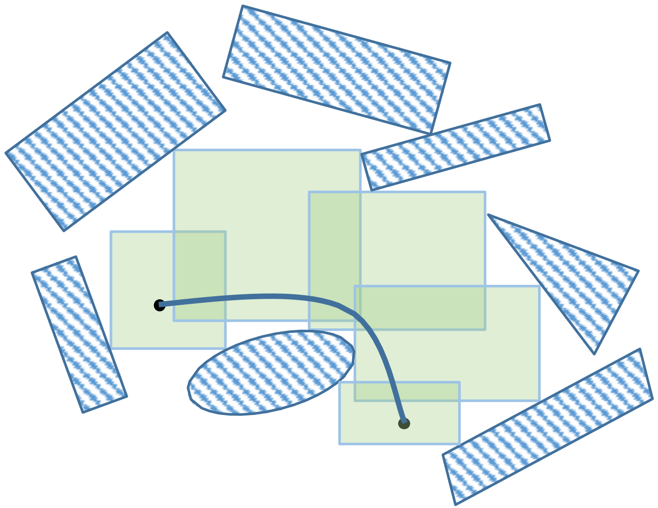
Hard constrained Minimum-snap

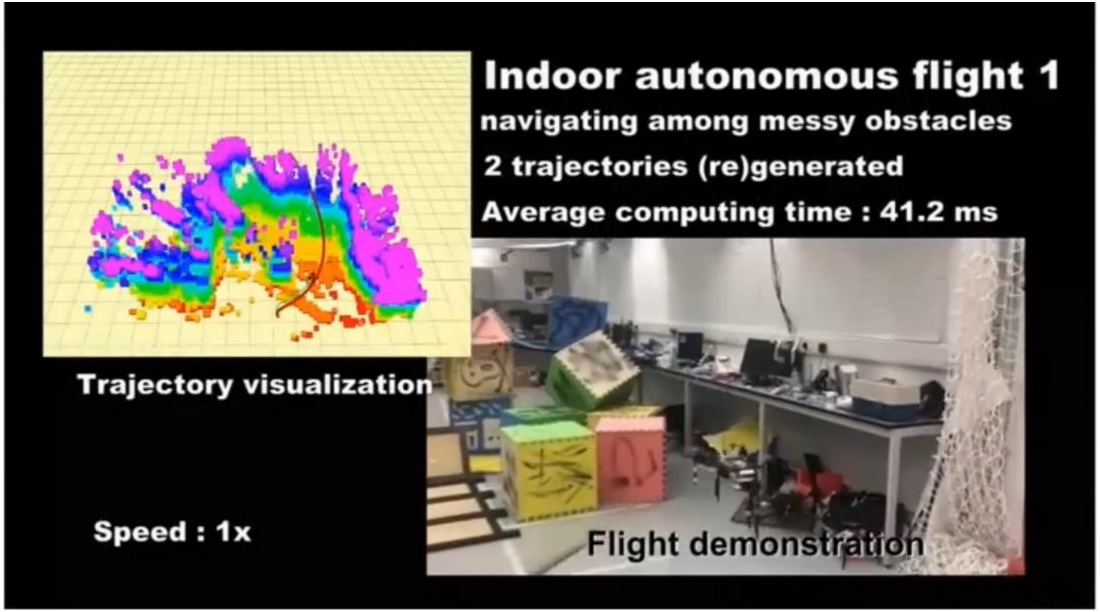




Back-end: Trajectory Optimization

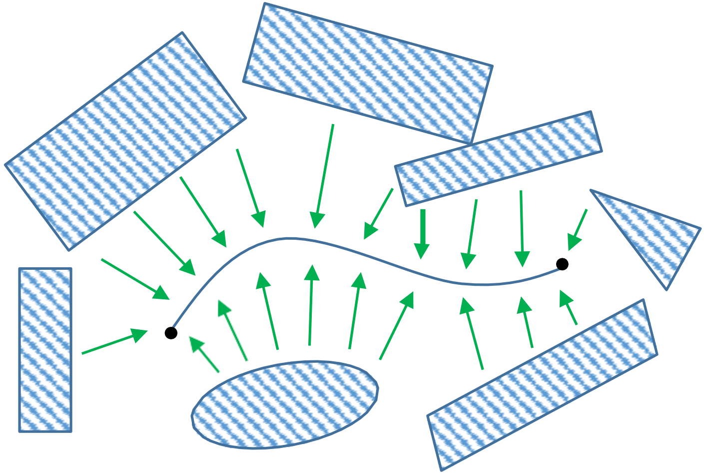
Hard constrained Minimum-snap

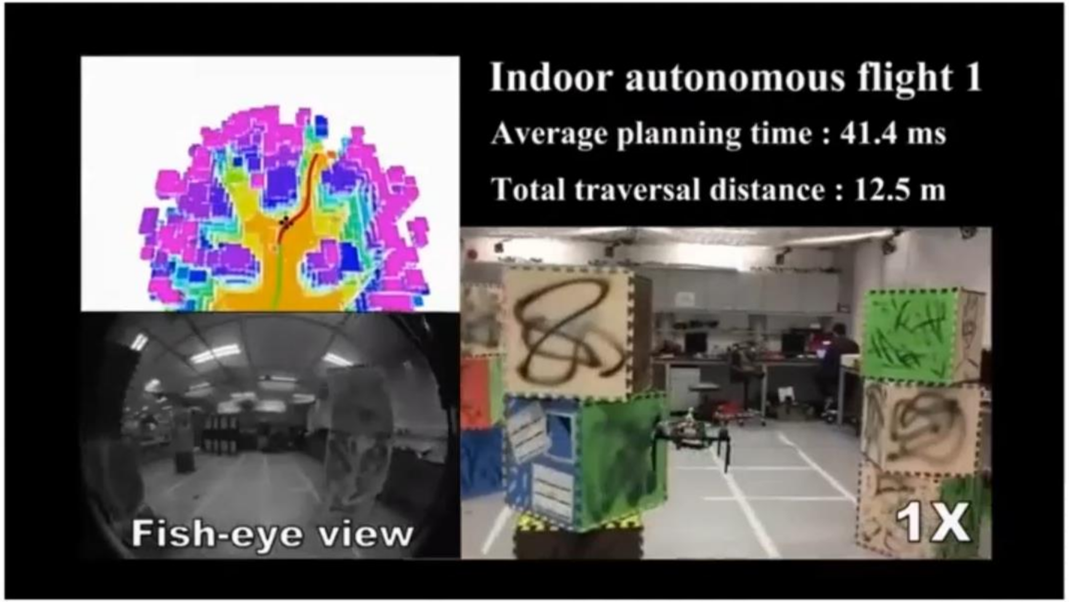




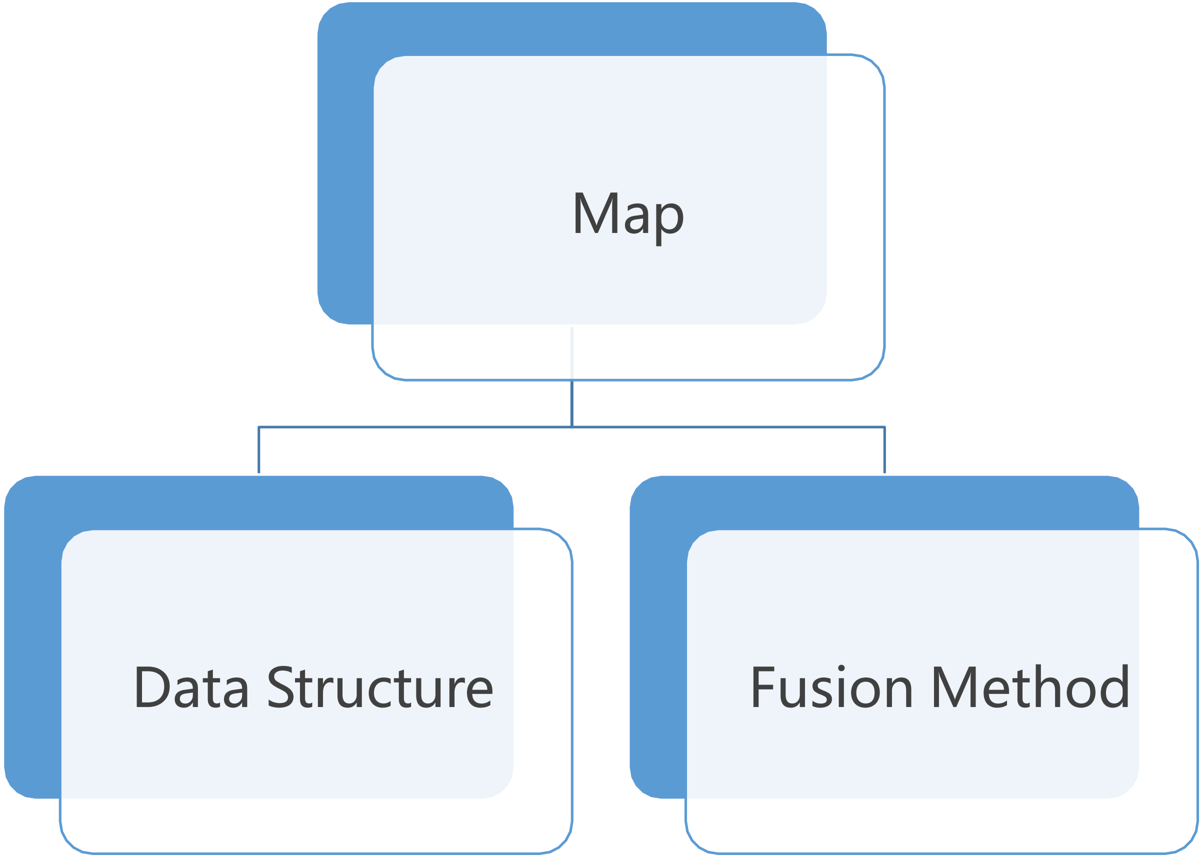
Back-end: Trajectory Optimization

Soft constrained Minimum-snap

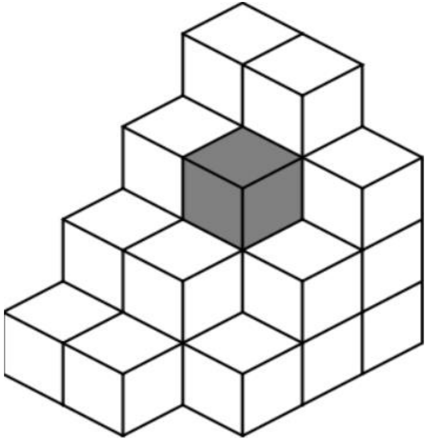


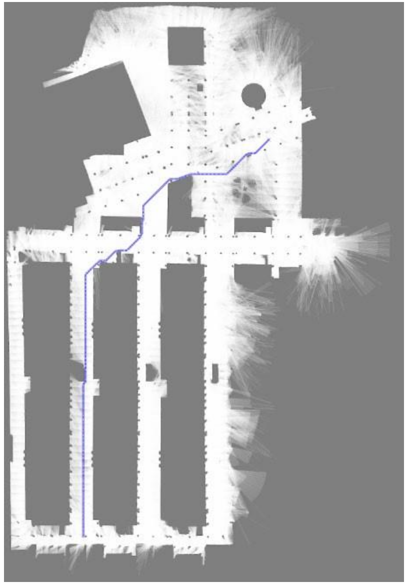


Map Representation

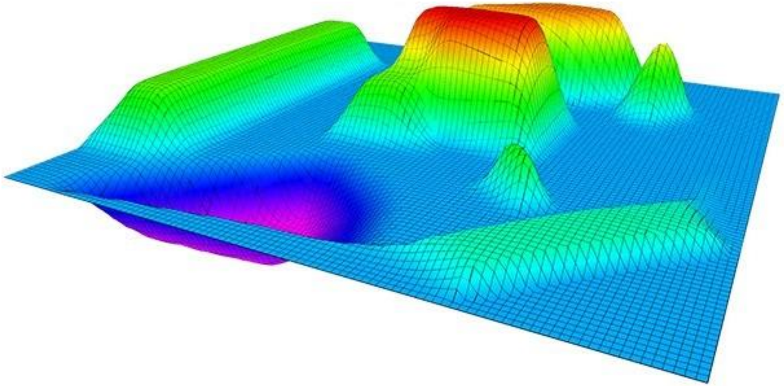

Map
Data Structure
Fusion Method

Occupancy grid map



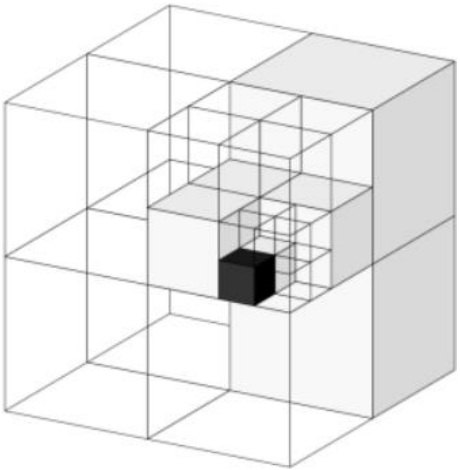


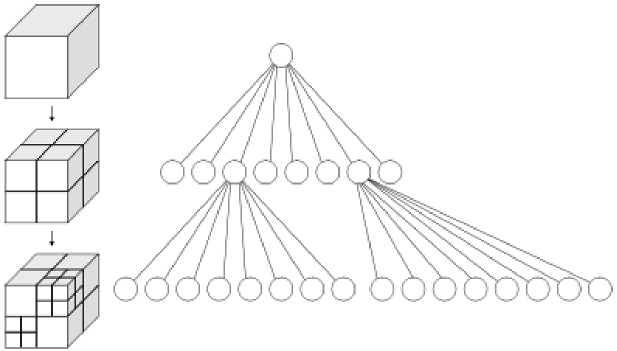
- Most Dense
- Structural
- Direct Index Query



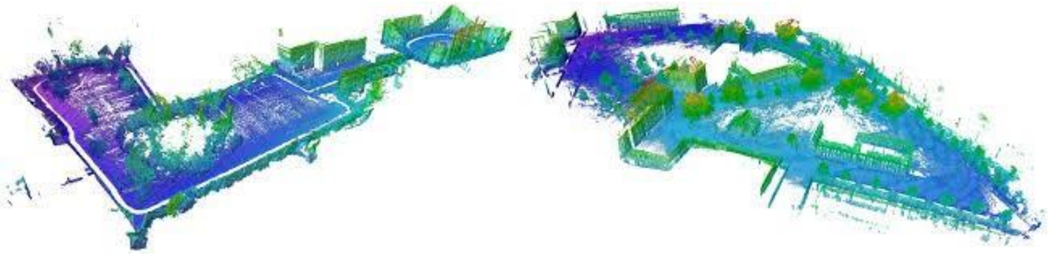
https://github.com/ANYbotics/grid_map

Octo-map



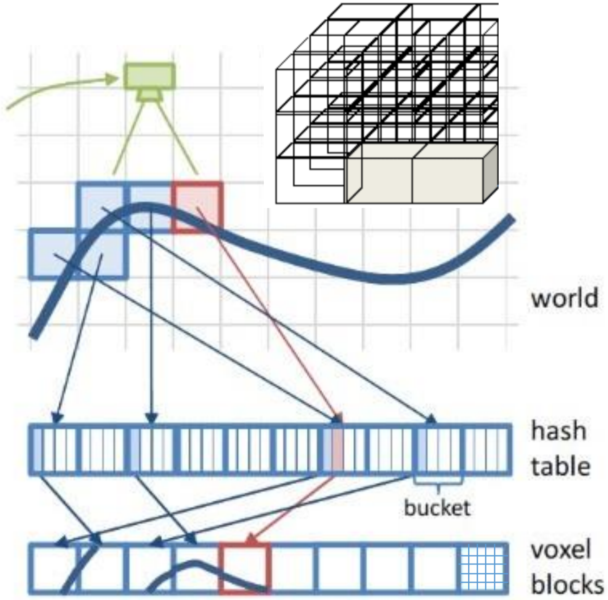


- Sparse
- Structural
- Indirect Index Query

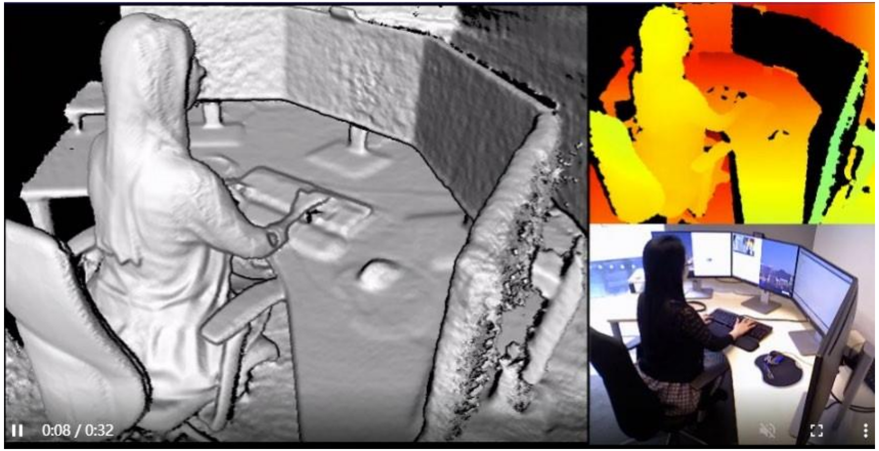


https://octomap.github.io/

Voxel hashing



- Most Sparse
- Structural
- Indirect Index Query



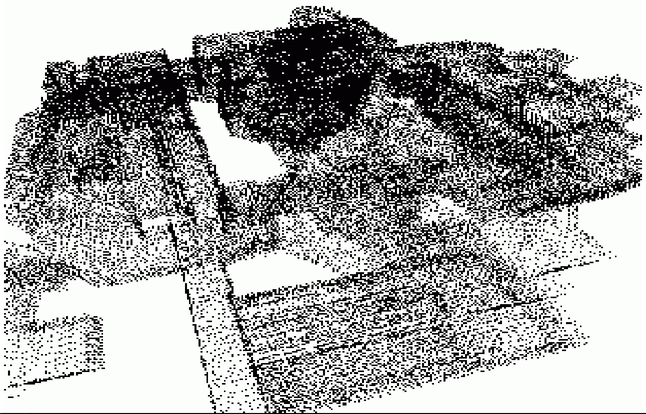
Voxel Hashing:

https://github.com/niessner/VoxelHashing

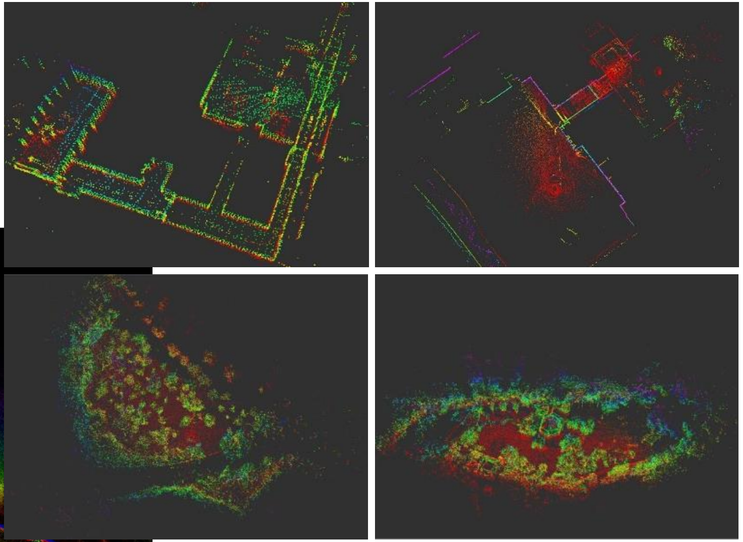
InfiniTAM:

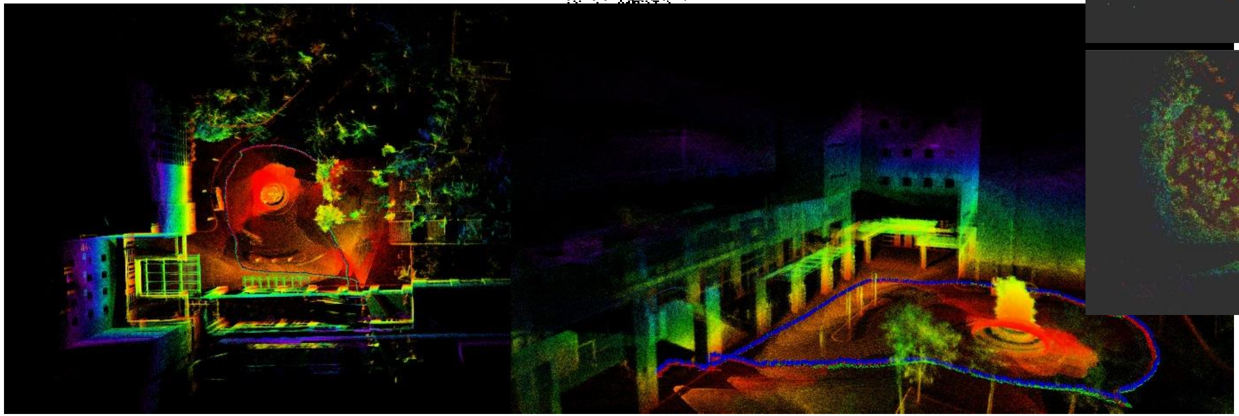
http://www.robots.ox.ac.uk/~victor/infinitam/

Point cloud map



- Un-ordered
- No Index Query



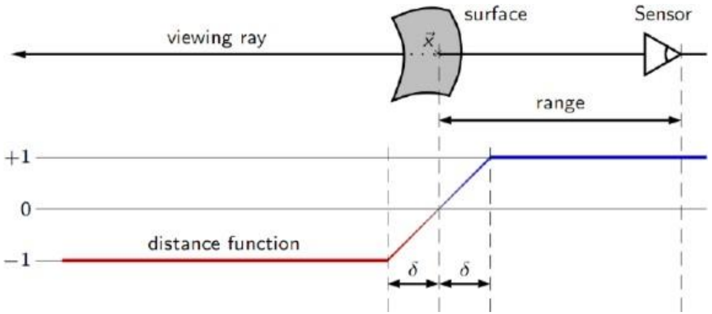


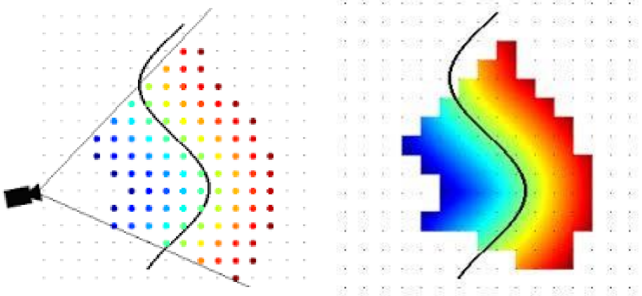
PCL

http://pointclouds.org/

TSDF map

Truncated Signed Distance Functions





OpenChisel

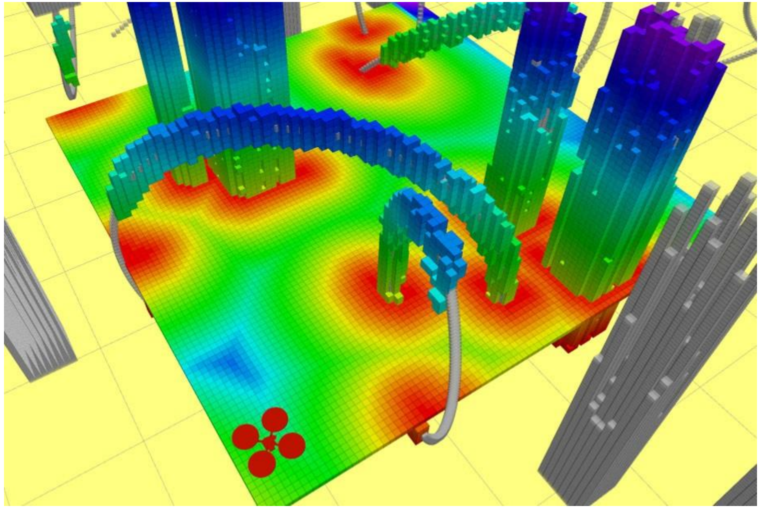
https://github.com/personalrobotics/OpenChisel

ESDF map

Euclidean Signed Distance Functions

Incremental Update, Global Map

Batch Update, Local Map



Distance Transforms of Sampled Functions, PF Felzenszwalb

VoxBlox

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

FIESTA

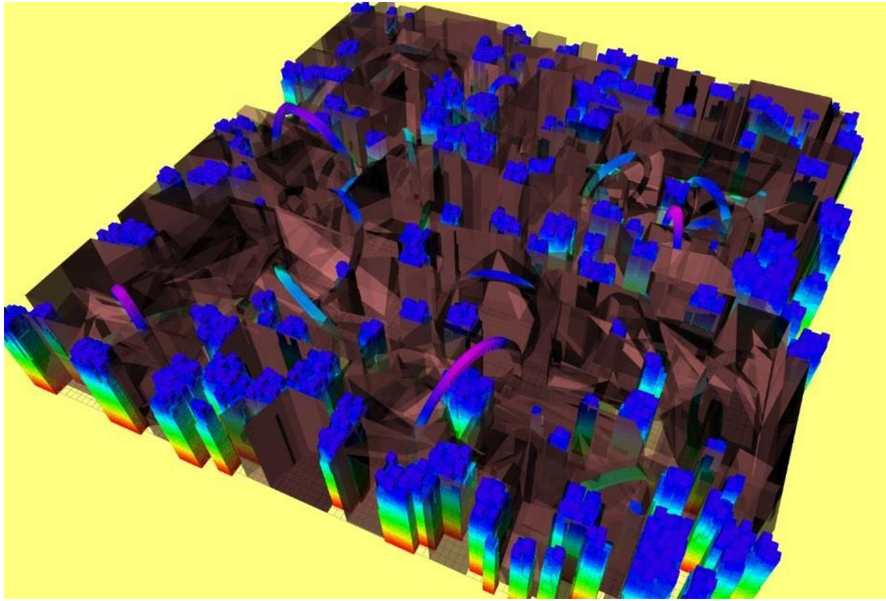
https://github.com/HKUST-Aerial-Robotics/FIESTA

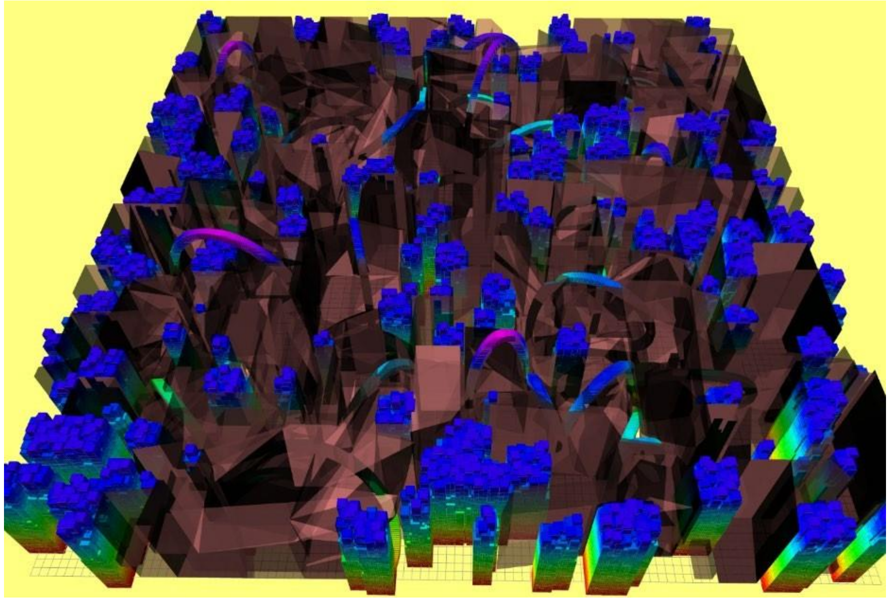
TRR's Local Map

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

More ?

Free-space Roadmap

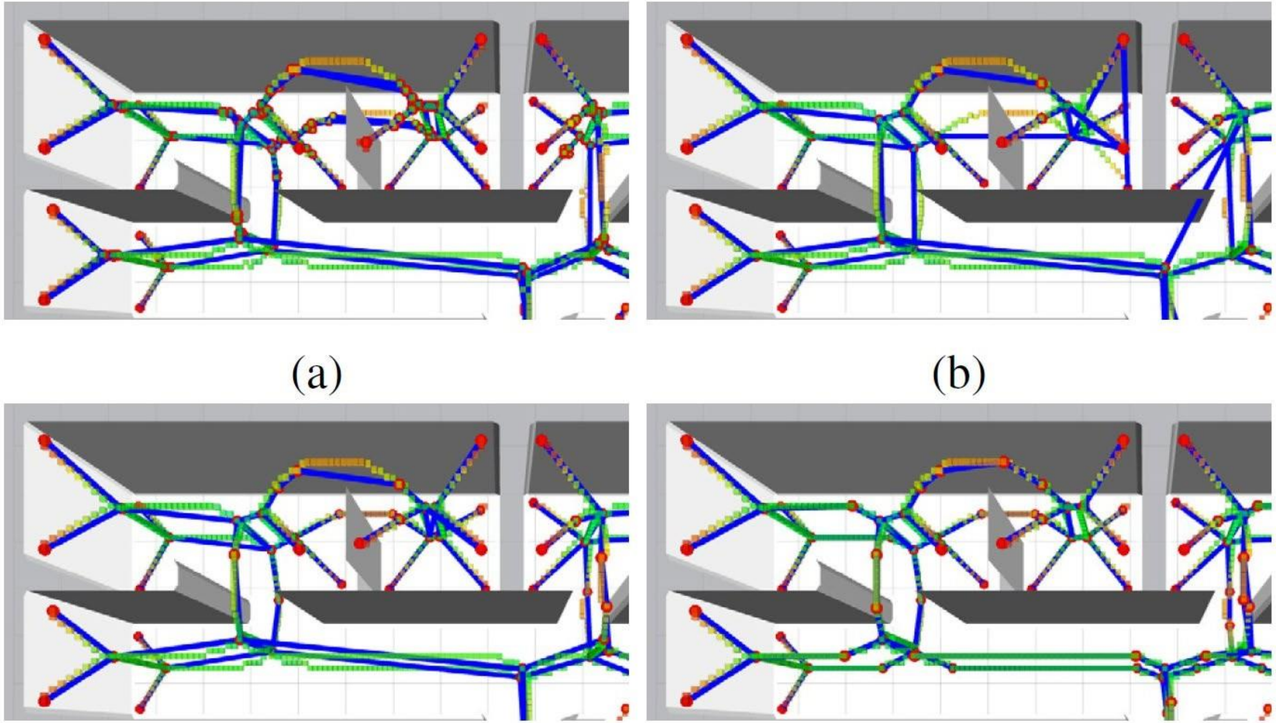




https://github.com/HKUST-Aerial-Robotics/Teach-Repeat-Replan

More ?

Voronoi Diagram Map



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

Pre-requirement

Matlab

- Please install Matlab

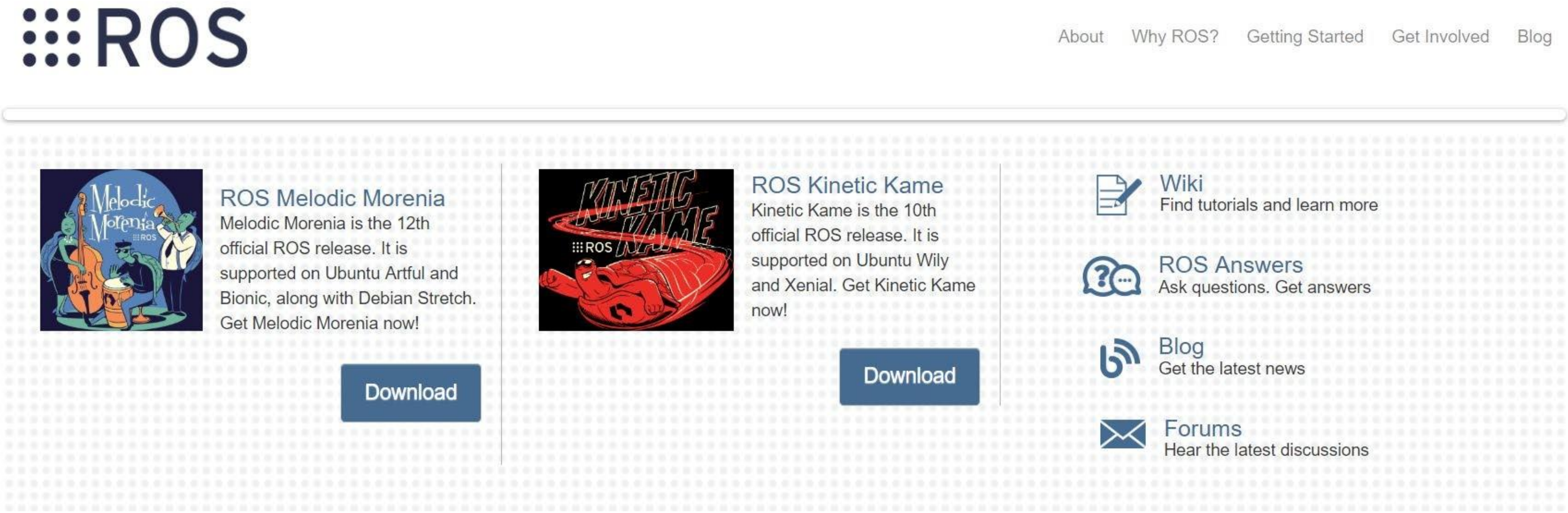
Linux

- Linux file system
- How to install software in linux
- Useful commands

C++ and GCC Toolchain

- C with class ?
- Gcc, Makefile, CMakeList
- Write CMakeList
- How to solve problems: google and document

ROS



- Follow ROS tutorial
- Ubuntu 16.04 + ros kinetic is recommended

Homework

Thanks for Listening!