Han Wang

hanwang028@gmail.com | wanhan@student.ethz.ch

EDUCATIONAL BACKGROUND

2021.09 - Master: Robotics, Systems and Control — ETH Zuerich

GPA：5.6 / 6.0 Tutor：Emilio Frazzoli

2017.09 - 2021.06 Bachelor: Mechanical Engineering — Wuhan University

GPA：3.83 / 4.0 (91.2 / 100) Rank：2 / 121

RESEARCH EXPERIENCE

2023.03 - 2023.08 **Decision Making and Motion Planning Intern，Qcraft (an autonomous driving company)**

- Optimized the decision on leading objects selection and leading cost: 1. take the future position of objects into account when selecting leading groups, ensuring objects with large velocity difference not to be assigned to the same group. 2. add another leading cost to penalty candidate trajectories which fail to overtake the leading groups behind the ego. These improvements enable the motion planner to cope with complex scenes better and prevent the 'Initializer' (one of the stage of the planner module) generating homogeneous trajectories across multiple searches.
- Derived leading groups (when lane changing) and leading objects (when lane keeping) from the CaptainNet (a neural motion planner) trajectory, preparing for the implementation of A* motion searcher.
- To accelerate the 'Initializer' search, another motion search algorithm A* was implemented as an alternative to the previous search algorithm, DP. Inspired by CaptainNet, A* is able to generate feasible trajectories in a much shorter time without sacrificing the trajectory quality. Compared with DP, the passing rate in large-scale simulation of A* increases by 0.38%, and the number of 'Initializer' computation timeouts is reduced by 77%. Experiments on the hardware platform showed that the average computation time is reduced by 34%, and the new motion searcher is finally deployed on L4 after dev test. The whole development included also putting forward a proposal, refactoring algorithms, composing algorithms, adding debug info, visualizing graph extension, benchmarking and testing on the real vehicles.
- To further deploy A* on L2, utilized reference line and reference speed table to compute the heuristic function, and added caches to speed up multiple searches. Compared with DP, the passing rate in simulation of A* decreases by 0.08%, but the number of 'Initializer' computation timeouts is reduced by 96%. The average computation time of the 'Initializer' and the whole planner module is reduced by 70.42% and 41.59% respectively. The dev test is going to be performed and the new searcher will be deployed on L2.
- Some other work including making the SimxOdd dashboard, editing test scenes and implementing metric evaluator also helps increase productivity and perfect performance evaluation.

2022.02 - 2022.06 **Semester Project, ETH Zuerich**

Temporal Sampling-Based Algorithm for Motion Planning in Dynamic Environments

Report: https://drive.google.com/file/d/1nWI-TFd0OvxST6b4Oz1cm-8SW6Fwh9Tj/view

- Implemented a novel motion planner, Temporal RRT*. Considering the time dimension while searching makes the motion planner able to generate safe trajectories in dynamic environments.
- Incorporated the steering function of Dubins RRT* and Kinodynamic RRT* so that the planner is capable of generating dynamically feasible trajectories for the dubins car and drones, without any collision with moving objects.
- Performed a benchmark to compare the performance of Temporal RRT*, Temporal PRM, OMPL PRM and OMPL RRT*. The experiments in a 10x10x10 map demonstrated that the success rate of Temporal RRT*'s planning is always 100% when the number of moving objects is 5, 10, 15, and 20 respectively. While the success rate of classical planners like OMPL RRT* is only 66% at the worst. Compared with Temporal PRM, the average length of safe trajectories decreases by 10%, and computation time decreased by 70%.
- Deployed the motion planner on PX4, experiments showed that replanning is possible for Temporal RRT* to cope with obstacles with varying velocity (in ROS).

2020.04 - 2020.10 **Research Assistant, University of Wisconsin, Madison**

Design Safe Trajectories for the Crazyflie Nano Quadcopter

Report: https://drive.google.com/file/d/1bYtbcUd8SZ4IfQaOqUsz-uD4NFkksvli/view

- Implemented, analyzed and compared some classical planning algorithms including Dijkstra, A*, RRT, RRT*, informed RRT*, PRM and Lazy PRM.
- With the object of minimum snap, generated smooth trajectories fitted by clamped B-splines to connect waypoints. To accelerate the optimization task, transformed the original QP problem to an unconstrained QP problem, which speeds up the computation by 86% without affecting the trajectory quality. Experiments showed that optimizing trajectories fitted by clamped b-splines consumes 66% less time than that fitted by polynomials, if both are transformed to unconstrained QP problems.
- Designed a cost function and adopted gradient descent to optimize both the snap and the trajectory total time. Under the optimized time allocation, the trajectory is able to connect waypoints better without sharp curves. Then kinematic constraints can be easily satisfied via time scaling.
- Putting all these together, a complete safe trajectory generation process was achieved: 1. obtain discrete waypoints from RRT*, 2. extract waypoints according to some density and get optimized trajectory, 3. perform time reallocation and time scaling to make the trajectory more feasible, 4. check collisions for the trajectory, and resample waypoints for the track segment where the collision happens and reoptimize it.

2021.09 - 2024.04 **Other courses and projects in ETH Zuerich**

- Probabilistic Artificial Intelligence: mini-projects about Gaussian Process, Bayesian Deep Learning, Bayesian Optimization and Reinforcement Learning like Actor-Critic.
- Vision Algorithms for Mobile Robots: implementation of a visual odometry pipeline.
- Deep Learning for Autonomous Driving: projects about Multi-task learning for semantic segmentation and depth estimation, 3D object detection with Point-RCNN.
- Planning and Decision Making for Autonomous Robots, Dynamic Programming and Optimal Control, Model Predictive Control, Robot Dynamics, High Performance Computing for Science and Engineering

---

SKILLS C++, ROS, Python, OpenMP, Matlab

If interested, you may find more details from https://hanwang028.github.io/