Jianheng Liu

☎ +86 15625293598 • ✉ 943678231@qq.com

https://github.com/jianhengLiu • https://jianhengliu.github.io/

I am currently a postgraduate in Harbin Institute of Technology (Shenzhen), China, supervised by **Prof. Haoyao Chen**. I obtained my bachelor degree at Harbin Institute of Technology (Shenzhen), China in 2021. My research interest lies at **Robotics and Autonomous Systems, Localization and Mapping, Motion Planning, Unmanned Aerial Vehicles** and **Autonomous Navigation**. Multimedia and Chinese CV are available at https://jianhengliu.github.io.

Education

- **Harbin Institute of Technology (Shenzhnen)** **Recommended exemption Graduate**
  Control Science and Engineering (Master degree), *2021/09–Present*
- **Harbin Institute of Technology (Shenzhnen)** **Rank: 15/70**
  Automation (Bachelor degree), *2017/09–2021/06*

Publication

- **RGB-D Inertial Odometry for a Resource-restricted Robot in Dynamic Environments**
  Jianheng Liu, XuanFu Li, Yueqian Liu and Haoyao Chen
  RA-L and IROS, 2022 (**Under review**)
- **Vision-encoder-based Payload State Estimation for Autonomous MAV With a Suspended Payload**
  Jianheng Liu, Yunfan Ren, Haoyao Chen and Yunhui Liu
  IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, 2021)

Honor & Awards

- Graduate Academic Scholarship of First-class (2021-2022), Undergraduate Academic Scholarship of First-class (2019-2020), Third-class (2018-2019), Second-class (2017-2018)
- National ROBOCON competition of First Price (2020), Second Price (2019)
- the Third Prize for 2019 National Challenge Cup, the Bronze Prize for 2019 Internet plus of Heilongjiang Province, the Golden Price for 2019 ZuGuang Cup of Harbin Institute of Technology (Shenzhen)
- the Second Prize for 2018 National English Competition for College Strudents
- the Grand Prize for the second International Youth Drone Competition

Research

- **RGB-D Inertial Odometry for a Resource-restricted Robot in Dynamic Environments:**
  Jianheng Liu, XuanFu Li, Yueqian Liu and Haoyao Chen
  RA-L and IROS, 2022 (**Under review**)
  Dynamic-VINS is a real-time RGB-D Visual Inertial Odometry (VIO) system for resource-restricted robots in dynamic environments. It is extended based on VINS-Mono. It combines object detection and RGB-D cameras for dynamic feature recognition to reduce the computational cost, achieving an effect comparable to semantic segmentation. It adopts grid-based feature detection and proposes a fast and efficient method to extract high-quality FAST feature points. A competitive localization accuracy and robustness in dynamic environments are shown in a real-time application on resource-restricted platforms.
  - **Referred Code:** Dynamic-VINS
  - **Referred Video:** Youtube, Bilibili

- **VINS-RGBD-FAST:** VINS-RGBD-FAST is a SLAM system based on VINS-RGBD. I do some refinements both in frontend and backend to improve the system's efficiency in resource-constrained embedded paltform, like HUAWEI Atlas 200DK, Raspberry Pi. For example, this system extracts FAST feature instead of Harris feature and solved feature clusttering problem, adds stationary initialization function, add IMU-aided feature tracking and extracted-feature area's quality judgement function, lowers the required bandwidth of the system, makes a trade-off of accuracy and efficiency by constrain the optimized variables in backend. Furthermore, we made this system as a module and applied it into UAV as a state feedback to track a generative trajectory stably.
  - **Referred Code:** VINS-RGBD-FAST.

- **Vision-encoder-based Payload State Estimation for Autonomous MAV With a Suspended Payload:**
  Jianheng Liu, Yunfan Ren, Haoyao Chen and Yunhui Liu
  IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, 2021)
  Dynamic-VINS is a real-time RGB-D Visual Inertial Odometry (VIO) system for resource-restricted robots in dynamic environments. It is extended based on VINS-Mono. It combines object detection and RGB-D cameras for dynamic feature recognition to reduce the computational cost, achieving an effect comparable to semantic segmentation. It adopts grid-based feature detection and proposes a fast and efficient method to extract high-quality FAST feature points. A competitive localization accuracy and robustness in dynamic environments are shown in a real-time application on resource-restricted platforms.
  - **Referred Code:** Vision-encoder-based-Payload-State-Estimator
  - **Referred Video:** Bilibili

- **MatRix:** A extreme interesting prototype developed in 2020 XBOT PARK Smart Product Innovation Boot Camp. An interactive smart carpet, which can achieve infinite splicing through the magnetic suction connector with anti-dull design. MatRix can be used as your home intelligent terminal, game console, decoration and so on.
  - **Referred Video:** Bilibili

- **quad-controller-SE3 & FlightController:** quadrotor controller based on PX4/mavros and SE3 geometric control. And I also develop a simulation based on CoppeliaSim software to compute the desired thrust and torque of quadroter according to dynamic modelling, and use distribution matrix to decide the motor's speed. Furthermore, I conduct a trajectory tracking controller to follow a generative minimum snap trajectory for experiment.
  - **Referred Code:** quad-controller-SE3, FlightController
  - **Referred Video:** Bilibili-1, Bilibili-2

- **BezierTrajGenerator & MinimumSnapTrajGenerator & MapManager:** Trajectory Generator based on Bezier Curve and Minimum Snap for autonomous robot. And I develop a 2D Map Manager for the verification and visualization for different algorithms.
  - **Referred Code:** BezierTrajGenerator, MinimumSnapTrajGenerator, MapManager

- **CoppeliaSim/V-Rep Steeling Wheel Robot Tutorial:** A detailed tutorial for a CoppeliaSim/V-Rep beginner to construct their own Steeling Wheel Robot and control it via ROS.
  - **Referred Code:** CoppeliaSim-Steeringwheel-Tutorial

- **Manipulator-GUI:** C++ Course Project (Complied in CodeBlocks). A three dimentional manipulator's forword/inverse kinematics calculation and visualization.
  - **Referred Code:** Manipulator-GUI