Autonomous Navigation for Flying Robots

Lecture 8.1:
Visual Navigation
with a Parrot Ardrone

Jürgen Sturm
Technische Universität München
Parrot Ardrone

- Low-cost platform (300 USD)
- Controllable via wifi
- API is open-source
- Many language bindings
  - C/C++
  - Python
  - JavaScript
  - …
Software Architecture for Robotics

- Robots became rather complex systems
- Often, a large set of individual capabilities is needed
- Flexible composition of different capabilities for different tasks
Best Practices for Robot Architectures

- Modular
- Robust
- De-centralized
- Facilitate software re-use
- Hardware and software abstraction
- Provide introspection
- Data logging and playback
- Easy to learn and to extend
Robotic Middleware

- Provides infrastructure
- Communication between modules
- Data logging facilities
- Tools for visualization
- Several systems available
  - Open-source: ROS (Robot Operating System), Player/Stage, CARMEN, YARP, OROCOS
  - Closed-source: Microsoft Robotics Studio
Example Architecture for Navigation

Robot Hardware

- Sensor driver(s)
- Sensor interface(s)

- Localization module

- Global path planning
  - User interface / mission planning

- Local path planning
  - Local path planning + collision avoidance

- Actuator interface(s)
- Actuator driver(s)
Robot Operating System (ROS)

- http://www.ros.org/
- Installation instructions, tutorials, docs

ROS.org

Documentation

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license.

Install

Install ROS on your machine.

Getting Started

Tutorials: technical overview, and links to getting help. Also, check out the ROS cheat sheet.

Contribute

How to contribute to the ROS community, such as submitting your own repository. See the ROS Planet for what others are doing.
Camera-based Navigation

[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

- Monocular SLAM @18Hz
- Extended Kalman Filter @200Hz
- PID Control @100Hz

Quadrocopter

Video

IMU
Camera-based Navigation
[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

Jürgen Sturm
Autonomous Navigation for Flying Robots
Monocular SLAM

- Based on PTAM library [Klein and Murray, ISMAR 2007]
- Visual SLAM
  - Match visual features between keyframes
  - Optimize camera poses and 3D feature points
- Optimized for dual cores, highly efficient, open-source

Thread 1 (real-time)

- Track camera
- Track camera
- Track camera

Thread 2 (not real-time)

- Optimize map
- Optimize map
- ...
Camera-based Navigation

[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

Jürgen Sturm

Autonomous Navigation for Flying Robots

11
Camera-based Navigation
[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

- Based on PTAM
- Our contributions:
  - Enhanced reliability by incorporating IMU into PTAM
  - Maximum likelihood scale estimation from ultrasound altimeter and IMU
Camera-based Navigation
[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

- Input: PTAM estimate, IMU, controls
- Output: pose estimate
- State vector: \( (x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\psi})^T \)
- Full, calibrated model of the flight dynamics
- Delay compensation (~200ms)
Time Delays
[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

obs. PTAM pose: \[\rightarrow\]
obs. \(\dot{x}, \dot{y}, \dot{z}\): \[\rightarrow\]
obs. \(\Phi, \Theta, \Psi\): \[\rightarrow\]
EKF prediction: \[\rightarrow\]

\[\approx 90 \text{ ms} \quad \approx 35 \text{ ms} \quad \approx 75 \text{ ms}\]

- Last visual observation
- Last IMU observation
- Now
- Command received
Camera-based Navigation

[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

- Based on predicted state from EKF
- Approach and hold target position \((x, y, z, \Psi)^\top\)
- High level control:
  Hold position, assisted control, follow waypoints
Results

[Engel, Sturm, Cremers; IROS 2012; RAS 2014]

Hold Position

- autonomous flight
- only onboard sensors
- no prior knowledge about environment
- automatic mapping and scale estimation

Lessons Learned

- Parrot Ardrone
- ROS as a middleware
- Monocular SLAM with PTAM
- Example: Visual navigation with the Parrot Ardrone
  - Fast & accurate navigation (with up to 2 m/s)
  - Source code available on http://www.ros.org/wiki/tum_ardrone