PROJECT ARTEMIS
VISUAL NAVIGATION FOR FLYING ROBOTS

Mohammed Kabir
STATE OF THE INDUSTRY

• Highly dependent on GPS in assisted modes.
• Requires sufficient piloting skills in non GPS-assisted modes.
• Chance of ‘flyaways’ due to bad GPS reception.
• Immediate need for robust GPS-agnostic navigation methods.
• No environmental awareness.
• Not truly autonomous.
CHALLENGES

- Multicopters are highly dynamic systems.
- They are inherently unstable and require active control strategies for stable flight.
- System dynamics are coupled and fast.
- Limited in terms of onboard computing and sensing hardware that can be carried.
- QoS for wireless datalinks to the MAV cannot be relied on in all environments.
LOCALISATION

- We use a SLAM (Simultaneous Localisation and Mapping) technique on our robot.
- Visual SLAM is globally consistent, and centimetre-level accurate unlike GPS, and works indoors and outdoors.
- Tight fusion with time-synchronised inertial measurements greatly increases robustness and accuracy.
PERCEPTION

• Multiple cameras provide proprioceptive information about the environment.
• All the cameras and the IMU (Inertial Measurement Unit) are synchronised in time with respect to each other.

• Forward stereo cameras are used to compute depth images in realtime.
• Depth images are used to build a 3D map of the environment is built incrementally.
SENSING SUITE

Drift speed

Framerate (dynamics)

High

Medium

Low

Ideal sensor

Compass

IMU

Cameras, Laser Rangers

GPS

Drift correction

Drift correction

Drift correction

Credit: Stephan Weiss, PhD thesis 2012
The system is designed to navigate using all available sensors in the environment - GPS, Vision and Lidar.

Sensor availability is not guaranteed - modular sensor fusion approach using a hybrid Kalman filter with fault detection is used.

Even if a particular subset or module were to fail, the overall system performance would not be compromised.
ROBUST MULTI-SENSOR FUSION
PLANNING AND CONTROL

• The global volumetric map is used to continuously compute a collision-free trajectory for the vehicle.
• Assisted modes - planner only intervenes if the operator’s high-level position commands could lead to a possible collision.
• Autonomous modes - planner computes optimal trajectories to completely explore the environment.

"His path-planning may be sub-optimal, but it's got flair."
OPERATOR INTERFACE

• We use a single laptop and a tablet for our ground control system.
• A long-range Ubiquiti modem is used as the primary air-to-ground datalink.
• The laptop runs SLAM visualisation and the tablet runs live FPV (first-person-view). The operator can use this high-definition feed to fly.
Vehicle Platform

- PX4 Autopilot
- Intel i7 computer
- Rocket M5 (wireless link)
- Propulsion system
- Stereo cameras (time-synchronised)
VEHICLE PLATFORM

• The current-generation developmental prototype was designed after multiple iterations, building on top of our previous visual MAVs.
• Intel Core i7 onboard computer running Ubuntu 14.04 Server.
• Pixhawk autopilot running the PX4 Flight-stack.
• Ubiquiti Rocket M5 long-range wireless datalink.
• Forward facing stereo cameras, bottom facing optical flow camera and separate monocular camera.
• All low-level sensors like GPS, compass and actuator controllers (ESCs) are interfaced via the CAN bus.
SOFTWARE FRAMEWORK

• Software architecture follows a high-level / low-level separation model for maximal reliability. The flight core is isolated from the application-level processing to ensure stability of the core vehicle operation, independent of the high-level system state.

• Low-level tasks critical to flight control, like motor actuation and attitude estimation run on the PX4 Middleware on the NuttX RTOS.

• High-level tasks like computer vision run on the onboard Linux computer, on top of the ROS (Robot Operating System) Middleware.
THANK YOU!

Visit www.uasys.io/research for more!

Our open-source software stack is available at www.github.com/ProjectArtemis