Real Time, Onboard-only Landing Site Evaluation for Autonomous Drones

PhD Thesis Proposal

Joshua Springer

Reykjavík University

March 2022

Presentation Structure

(1) Introduction
- Problem description and motivation
- State of the Art

(2) Completed/ongoing projects
- Initial proof of concept attempt
  - Continuation of master thesis (tested in simulation)
- Fiducial marker modifications
- Proof of concept

(3) Research Plan
- Methods
- Challenges and risk analysis

Introduction

Problem Description and Motivation

- Much of basic drone flight has been **automated**.
  - Takeoff
  - Waypoint-to-waypoint-flight
  - Track/orbit objects, take pictures, etc.
- Landing is still largely **manual**.
  - No continuous, autonomous mission cycles
  - Primitive, semi-autonomous methods are common (still require human operator)
  - Hand-catching is common

"Human-assisted landing"

Research Questions

- How can a drone autonomously land?
- What data do autonomous drone landing methods need?
- How can those methods execute in real time onboard a drone?

State of the Art

- GPS-based landing
  - RTK
- Known landing locations:
  - Visual matching
  - Visual markers
  - IR beacons
- Terrain analysis
  - Optical flow
  - RGBD, LIDAR
- Other methods

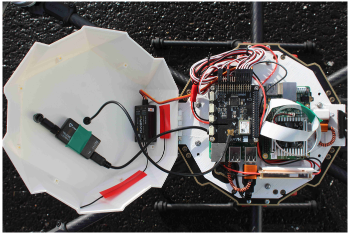
Completed and Ongoing Projects

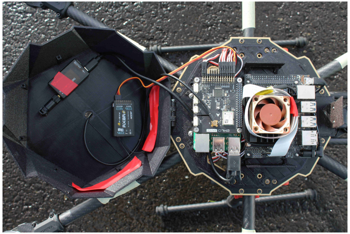
Test Hexacopters

- Two Tarot 680 hexacopters
- For real-world proof of concept of master thesis simulations.

Test Hexacopter Components

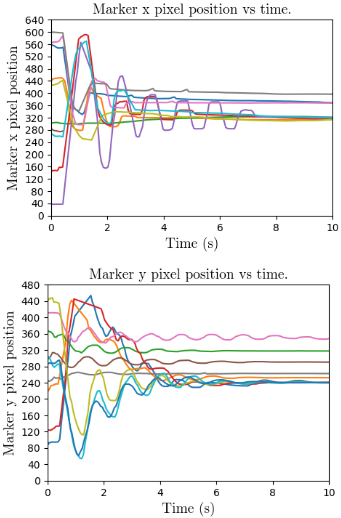
- Navio2 + RPi 3 autopilot combo
- Companion boards (for heavy onboard processing):
  - Google Coral (embedded TPU)
  - Jetson Nano (embedded GPU)
- Gimbaled camera modules
- 433 MHz telemetry
- 2.4 GHz R/C control
- Tested Autopilot Softwares
  - ArduPilot
  - PX4 (not technically supported)





Test Hexacopters' Performance

- Stable (manual) flight performance
- ~20 min flying time
- Successful marker tracking
- Errors during approach
  - Monocular pose estimation ambiguity
  - GPS inaccuracy
- No successful autonomous landing (but almost)



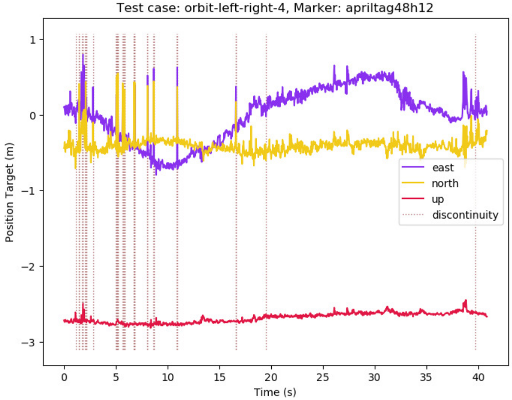
Heavy Lift IR Drone

- Project with Christopher Hamilton (geologist, University of Arizona) and Baldur Björnsson
- 1.3 m span, 25 kg lift
- FLIR camera
- Surveyed lava field at Fagradalsfjall
- Featured on BBC Click

Fiducial System Modifications

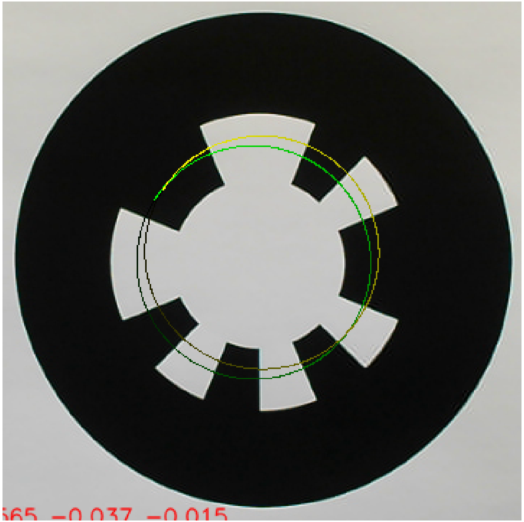
Necessary properties:

- Mitigates orientation ambiguity
- Detectable at long- and short-range
- Runs on embedded hardware



Orientation Ambiguity in WhyCode

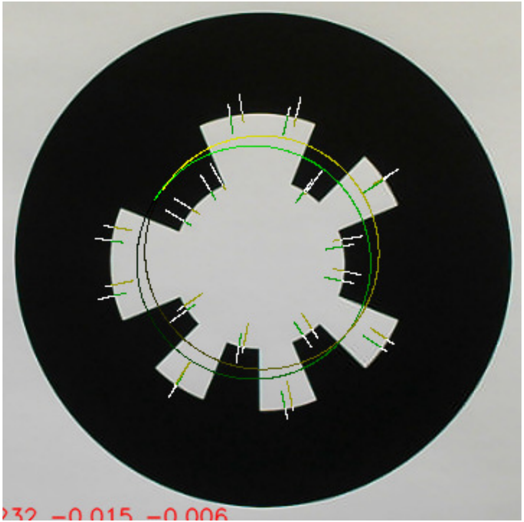
- Semi-axes → 2 possible orientations
- Better centered → correct
- Arclength of intersections with ID “teeth”



Fiducial System Modifications: “WhyCode Ellipse”

Approach 1: Extra tooth sampling

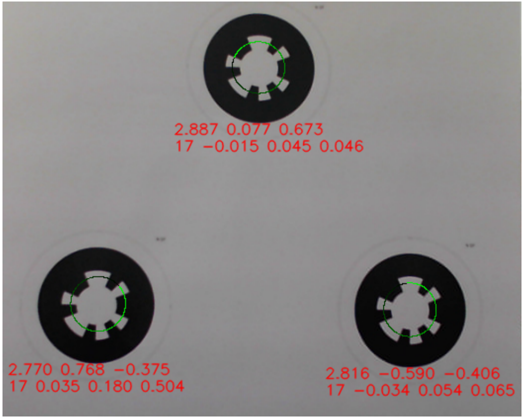
- Sample ID with original method.
- Add: radial sampling on tooth edges.
- Choose solution based on tooth edge predictions.



Fiducial System Modifications: “WhyCode Multi”

Approach 2: Coplanar marker arrangements

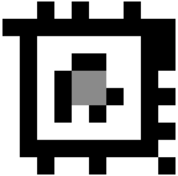
- Ignore individual marker orientations
- Calculate normal vector to the plane connecting the markers.
- Extract pitch and roll from the normal vector.
- Extract yaw from the marker IDs.
- Takes advantage of WhyCode’s efficiency.



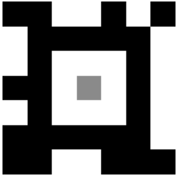
Fiducial System Modifications: April Tag

April Tag: less orientation ambiguity, but less computationally efficient.

April Tag 48h12: more sophisticated, "recursive."

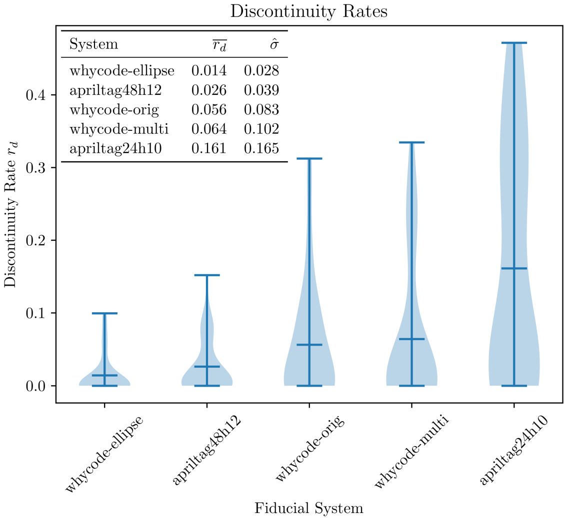


April Tag Custom 24h10: "recursive," smaller definition



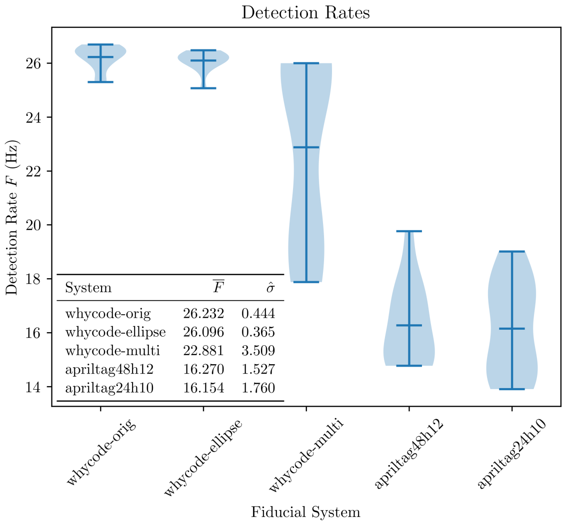
Performance Analysis: Discontinuity Rates

- Orientation ambiguity $\rightarrow$ discontinuities.
- Discontinuity rate $\overline{r_d}$ is the number of discontinuities per detection.
- Lower is better.



Performance Analysis: Detection Rates

- Detection rate $\overline{F}$ is the number of detections per second.
- Tested on Raspberry Pi 4.
- Higher is better.



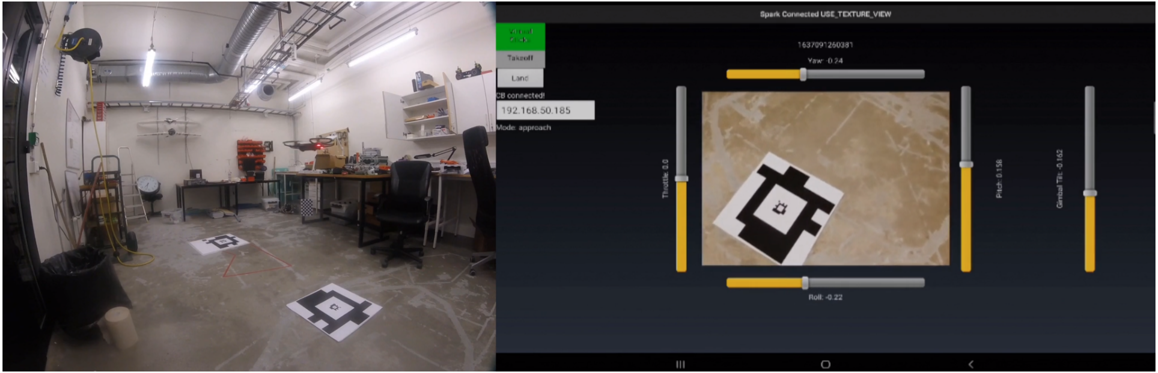
System	$\overline{F}$	$\hat{\sigma}$
whycode-orig	26.232	0.444
whycode-ellipse	26.096	0.365
whycode-multi	22.881	3.509
apriltag48h12	16.270	1.527
apriltag24h10	16.154	1.760

Autonomous Landing Proof of Concept

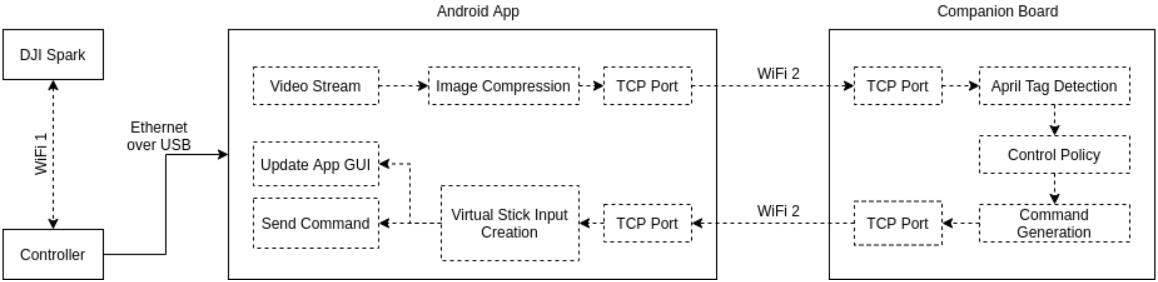
- Indoor experiments with DJI Spark
  - Reduces logistical considerations: transportation, weather
  - Stable out-of-the-box autonomous flight
  - Doesn't require GPS (uses other sensors)
- Requires DJI Mobile SDK, Custom Android App, and **lots** of workarounds.
- Video frames are offloaded (via WiFi) to Raspberry Pi 4 for processing
- Limiting factor: pre-transmission image compression on tablet (6-7 Hz)

(Banana for scale.)

Demo with worst-performing April Tag 24h10!



Autonomous Landing Proof of Concept: System Architecture



Publications

- Submitted: Evaluation of April Tag and WhyCode Fiducial Systems for Autonomous Precision Drone Landing with a Gimbal-Mounted Camera
- In Progress: results from autonomous landing proof of concept

Research Plan

Overview: Unstructured Autonomous Landing

- Focus on terrain analysis
  - Topographical analysis
  - Semantic segmentation
    - terrain type classification: (snow, ice, water, grass, rock, etc.)
    - classify according to predicted safety: (safe, questionable, unsafe, etc.)
- Focus on real time performance
  - Minimize computational requirements
  - Target specific hardware platforms
- Overall structure:
  - Input: sensor data
  - Process (quickly): ??
  - Output: safe landing sites (e.g. heat map) $\rightarrow$ flight control commands

Data Set Generation

AirSim: realistic simulator

- Automatic generation of large data sets
- Synthetic sensor data (LIDAR, RGBD cameras)
- Tag with IMU data
  - LIDAR $\rightarrow$ RADAR
- Specify realistic sensor parameters
- Segmentation masks for high-level label generation
- Labeling method can be slow, hand-tuned

Image source

Terrain Classifier Creation

- Test several methods
  - Conventional signal/image processing
  - Deep learning methods
  - Combination
- Pre-processing wrappers:
  - Rectification/calibration
  - Downsampling/resizing
- Performance comparison
  - Reduce false positives

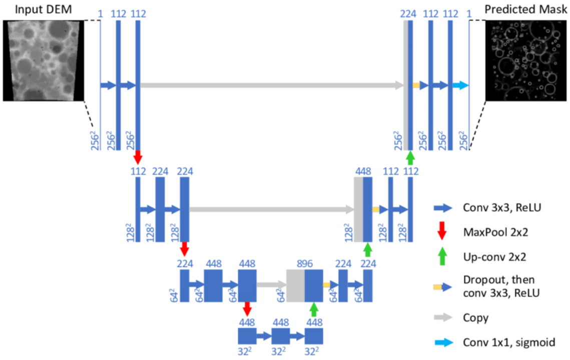


Image source

Testing in Simulation

- Post-processing wrappers:
  - Safe region tracking
  - Translation to flight commands
- Integration with ArduPilot/PX4 SITL
- Simulated autonomous missions in AirSim
- Qualitative analysis:
  - Does the drone land at all?
  - Does the safe region tracking work?
  - Does the autopilot software accept the commands?

Simulation is not enough!

Testing in the Real World

- Offline
    - Accuracy on real world data

Testing in the Real World

- Offline
  - Accuracy on real world data
- Lab scenarios
  - Runtime framerate on embedded hardware
  - Power requirements on embedded hardware

Testing in the Real World

- Offline
  - Accuracy on real world data
- Lab scenarios
  - Runtime framerate on embedded hardware
  - Power requirements on embedded hardware
- Real world landing scenarios

Drone Upgrades

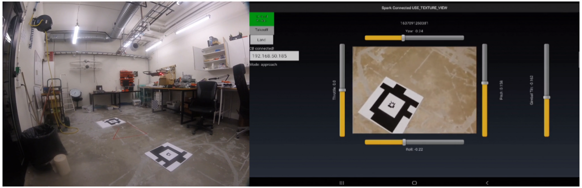
- New flight controller: Pixhawk Cube Orange
- Here3
- Supplement GPS
  - Optical Flow
  - LIDAR rangefinder
- Protective sensor cases, gimbal mounts
  - Intel RealSense D435 RGBD camera
  - Intel RealSense D455 RGBD camera (IMU)
  - Intel RealSense L515 LIDAR (IMU)
  - Texas Instruments IWR6843 60 GHz RADAR

Main Risks

- The synthetic data does not accurately represent the real world!
  - Show results in simulation.
  - Use real world data $\rightarrow$ no segmentation masks.
- The embedded hardware is too slow!
  - Reduce computational needs $\rightarrow$ prune network, decrease input data size
  - Use non-embedded hardware $\rightarrow$ generate reliable flight commands on real world data

Summary

- Goal: autonomous drone landing
- Past work: landing via fiducial markers at *known* landing pads
  - Contribution: gimbal-mounted camera setup, new marker variants
- Research plan: unstructured landing
  - Sensors: RGBD, LIDAR/RADAR
  - Topological/semantic terrain analysis
  - Synthetic data
  - Testing in simulation
  - Real world tests: power/framerate
  - Real world tests: landing with a physical drone
- Thank you! Are there any questions?



Click to watch on Vimeo

References