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 deep-dive and 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
  - Miscellaneous tasks:
    track/orbit an object,
    take a picture, etc.
- Landing is still largely **manual**.
  - Problem in continuous, autonomous missions
  - 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: inaccurate (especially in Iceland, high GDOP).
- RTK GPS: sufficiently accurate
  - Requires extra equipment (powered base station)
  - Overhead initialization time at every new mission
  - Limited range (must be deployed at mission site)
- Fiducial landing:
  - Requires little overhead (only a marker printout and monocular camera)
  - Limited detection abilities (fixed, downward-facing camera)
- Unstructured Landing
  - PnP matching (land at takeoff site after GPS-based approach)
  - Terrain analysis via LIDAR point clouds, RGBD images, or optical flow
  - Typically slow (processing must be offloaded), or
  - Add behavioral constraints (optical flow needs motion)
- Other methods
  - IR beacons
  -

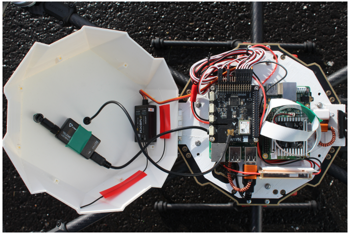
Completed and Ongoing Projects

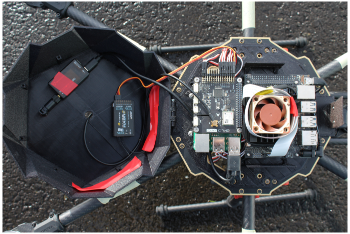
Test Hexacopters

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

Test Hexacopters' Components

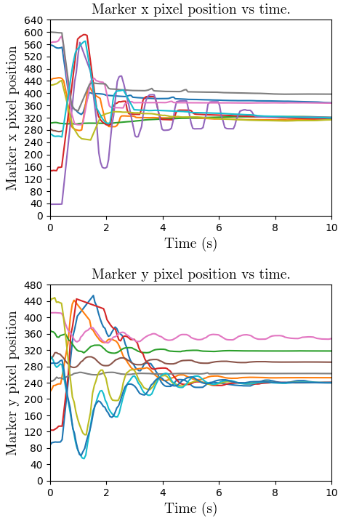
- Navio2 + RPi 3 autopilot combo
- Companion boards (for heavy onboard processing):
  - Google Coral (embedded TPU)
  - Jetson Nano (embedded GPU)
- Gimbaled camera modules
- 433 MHz telemetry
- 2.4 GHz R/C control
- Tested Autopilot Softwares
  - ArduPilot
  - PX4 (not technically supported)





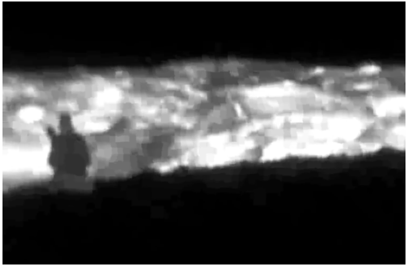
Test Hexacopters' Performance

- 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

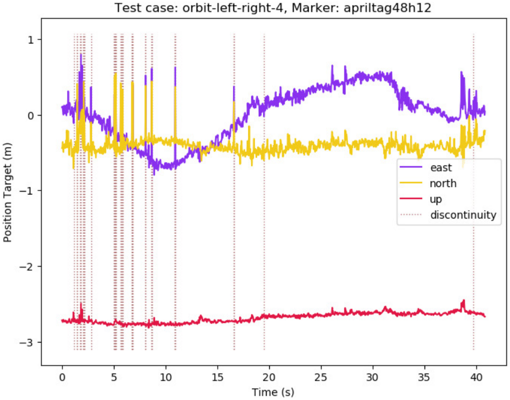
- Project with Christopher Hamilton (geologist, University of Arizona) and Baldur Björnsson
- 1.3 m span, 25 kg lift
- FLIR camera
- Surveyed lava field at Fagradalsfjall
- Featured on BBC Click



Fiducial System Modifications

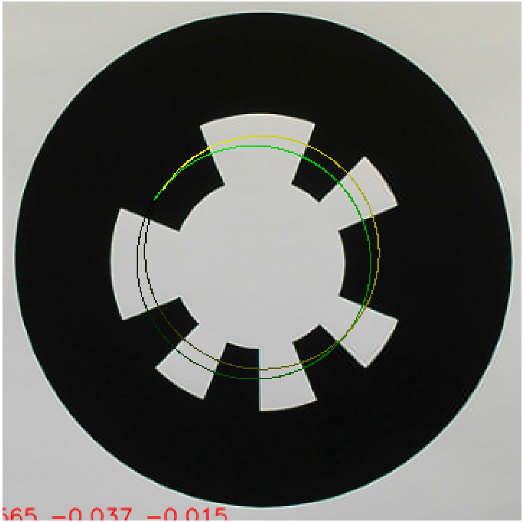
Necessary properties:

- Mitigate orientation ambiguity
- Maintain visibility of the marker at both long- and short- distances
- Be adequately computationally efficient to run on embedded hardware.



Orientation Ambiguity in WhyCode

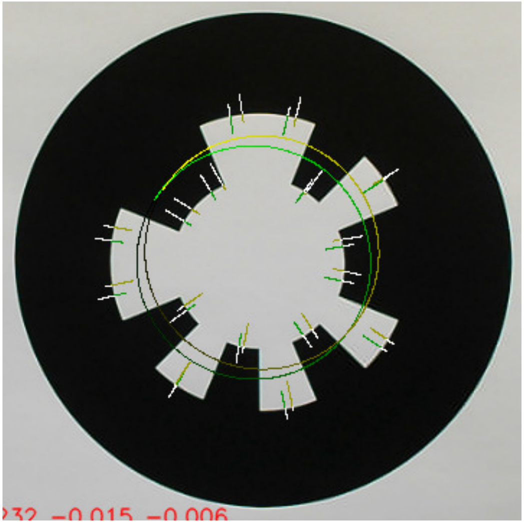
- 2 possible orientations
- Determined from semi-axes of outer black ellipse
- Choose the circle that is better centered on the marker
- Use arclength of intersections with ID “teeth”



Fiducial System Modifications: “WhyCode Ellipse”

Approach 1: Extra tooth sampling

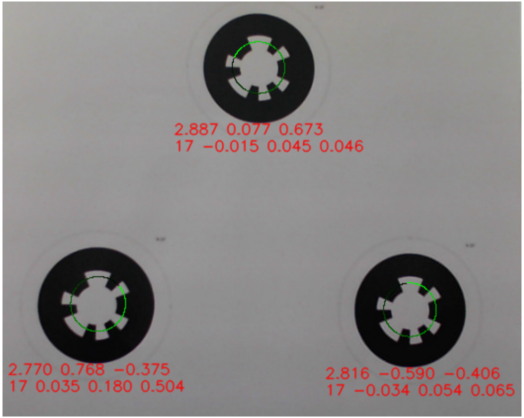
- Build on sampling of original method.
- Radial sampling through predicted tooth edges.
- Choose the solution that is centered better using predicted tooth edges.



Fiducial System Modifications: “WhyCode Multi”

Approach 2: Planar regression from at least 3 coplanar markers

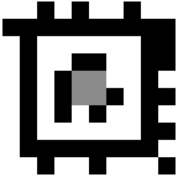
- Throw away individual marker orientations.
- Calculate normal vector to the plane connecting the markers.
- Extract pitch and roll from the normal vector.
- Extract yaw from the marker IDs.
- Takes advantage of WhyCode’s efficiency.



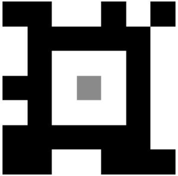
Fiducial System Modifications: April Tag

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

April Tag 48h12: more sophisticated, “recursive.”

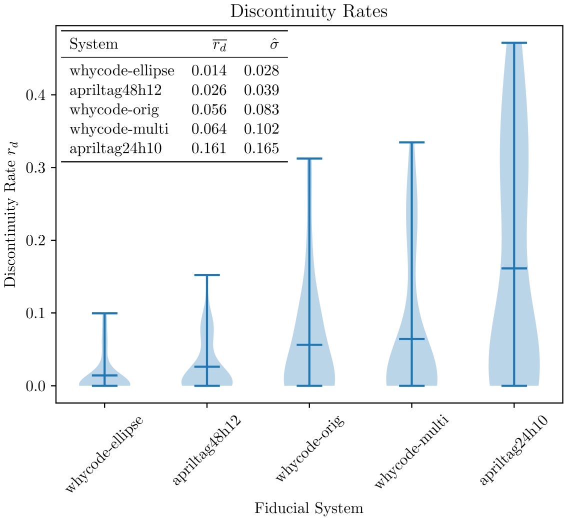


April Tag Custom 24h10: “recursive,” smaller definition



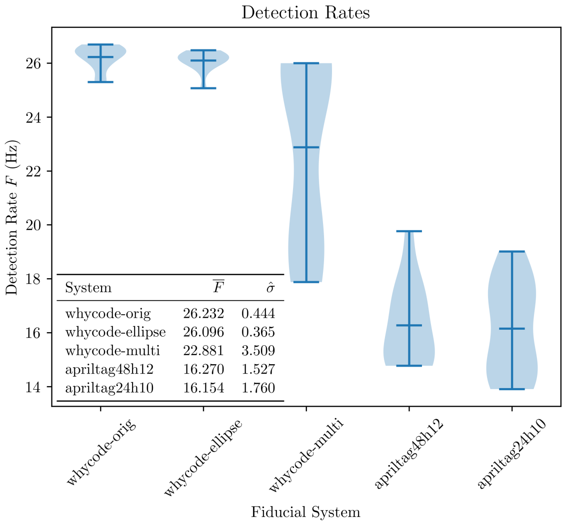
Performance Analysis: Discontinuity Rates

- 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.

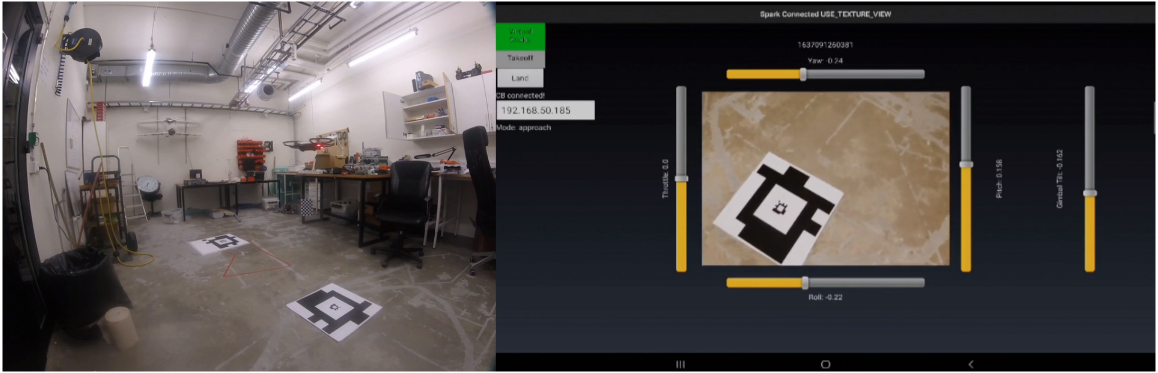


System	$\overline{F}$	$\hat{\sigma}$
whycode-orig	26.232	0.444
whycode-ellipse	26.096	0.365
whycode-multi	22.881	3.509
apriltag48h12	16.270	1.527
apriltag24h10	16.154	1.760

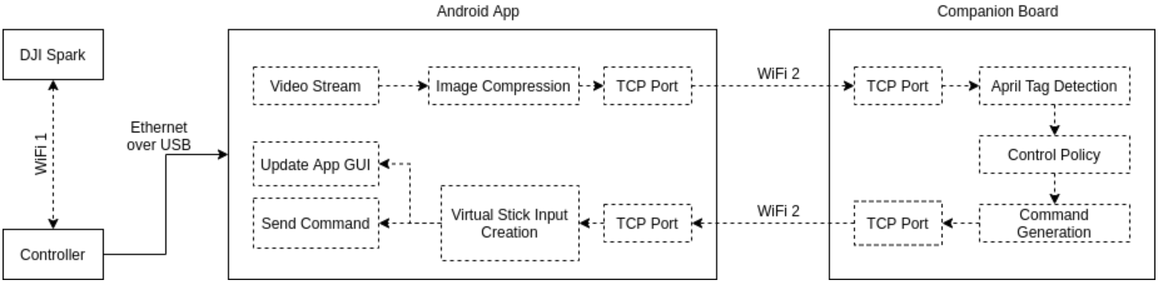
Autonomous Landing Proof of Concept **(FINALLY!)**

- 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)

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: Proof of Concept Results

Research Plan

Overview: Unstructured Autonomous Landing

- Focus on terrain analysis
  - Topological analysis: (smoothness, slant, flatness, size of flat regions)
  - Semantic segmentation: label terrain according to
    - the material it contains: (snow, ice, water, grass, rock, etc.)
    - how safe it appears for landing
- Focus on real time performance
  - Minimize computational requirements
  - Target specific hardware platforms
- Approach:
  - Input: sensor data (RGBD images, LIDAR/RADAR point clouds, etc.)
  - Base: CNN, U-net or Residual U-net for semantic segmentation
  - Potential pre-processing wrappers:
    - Rectification/calibration of images/point clouds
    - Downsampling of input resolution
    - Topological analysis
  - Potential post-processing wrappers
    - Safe landing site tracking
    - Translation to flight commands

Data Set Generation

AirSim: realistic simulator

- Scriptable, automatic generation of large, labeled, synthetic data sets
- Synthetic sensor data (LIDAR, RGBD cameras)
    - possible to imitate RADAR sensors with some tweaking of LIDAR sensor
- Specify realistic sensor parameters (distortion, range, update rate, etc.)
- Segmentation masks for high-level label generation
- Labeling method can be slow, hand-tuned

Image source

Offline Terrain Classifier Creation

- Performance comparison of several neural networks on the data set.
- Ranking in terms of classification accuracy
    - emphasis on reducing false identification of safe areas

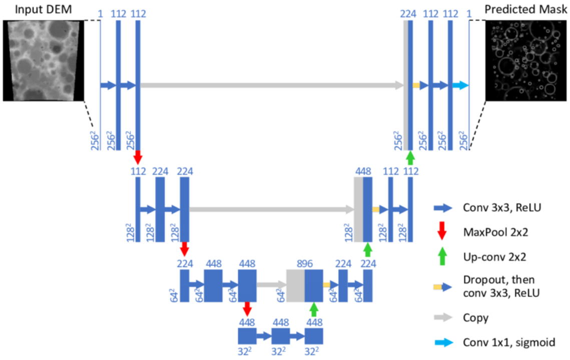


Image source

Testing in Simulation

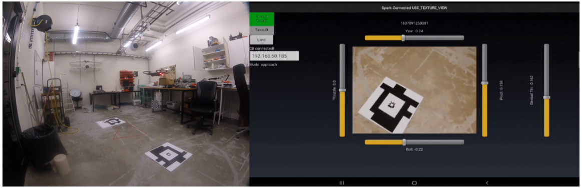
Testing in the Real World

Drone Upgrades

Risk Analysis

Summary

- Goal: autonomous drone landing
- Past work: landing via fiducial markers at *known* landing pads
- Research plan: unstructured landing (no known landing pad)
    - 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



Click to watch on Vimeo

References