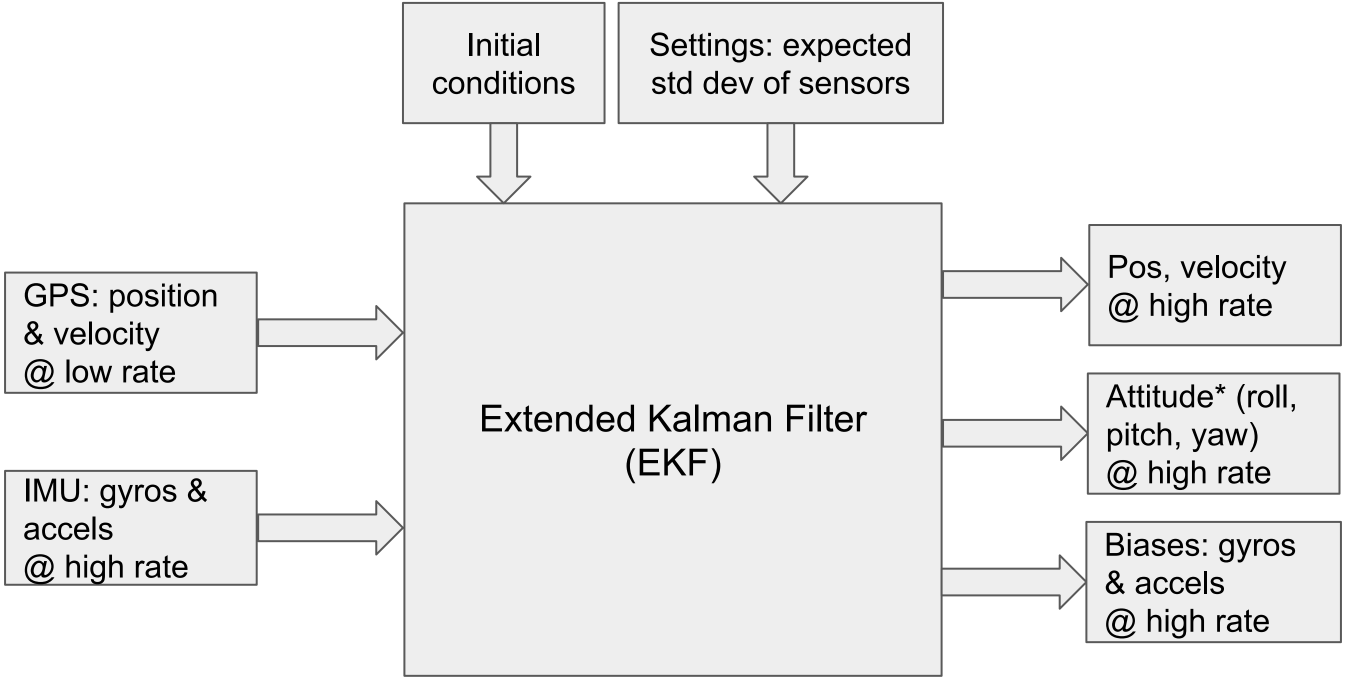
Exploring a Camera as a Sensor

Computer Vision, Augmented Reality, Visualization, (Truth?)

Curt Olson - UAV Lab - AEM
olson126@umn.edu

Extended Kalman Filter (EKF)



*Attitude states cannot be directly observed / validated.

Attitude States (aircraft roll, pitch, yaw)

- Attitude often feeds directly into critical flight control loops.
- A small amount of error is usually ok for getting around the sky.
- Chasing a bad yaw estimate could result in a fly-away.
- Chasing a bad roll or pitch estimate can be catastrophic.

Would it be possible to use a camera to observe aircraft attitude states?

Action Cameras

- We can see the horizon in the video.
- The horizon defines zero pitch and zero roll angles.

Challenges:

- Information is “visual”, not quantified.
- Different time reference / update rate.
- Camera is not aligned with aircraft body.

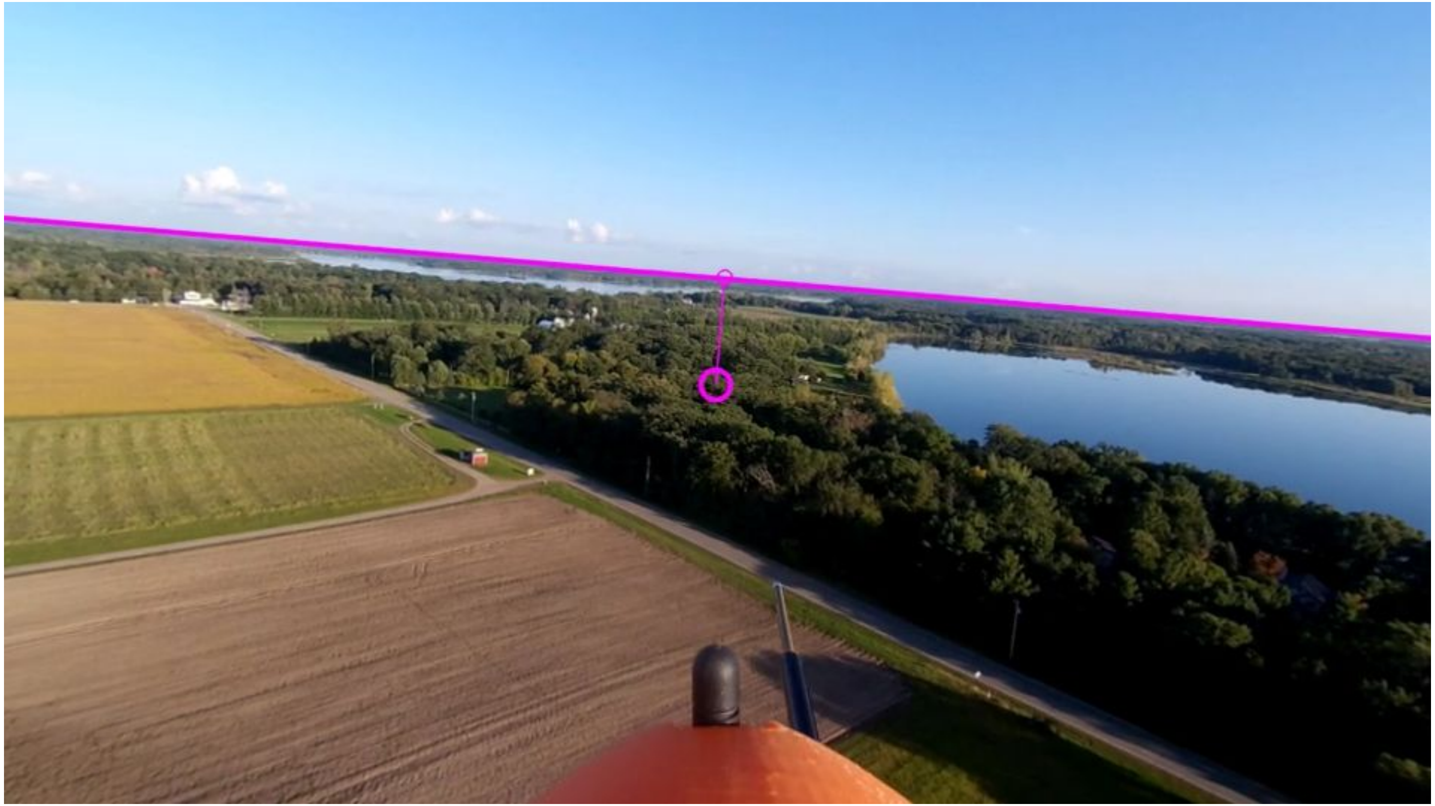
There it is! (the horizon)

Machine Vision Horizon Tracker

2020 Summer Internship - Nathan Noma (with some volunteer help by Isaiah Wardlaw)

- Goal: Develop a robust horizon tracker that works in a wide variety of weather and daylight conditions. Outputs horizon roll and pitch angle in camera reference frame.

Horizon located (pitch and roll angles)



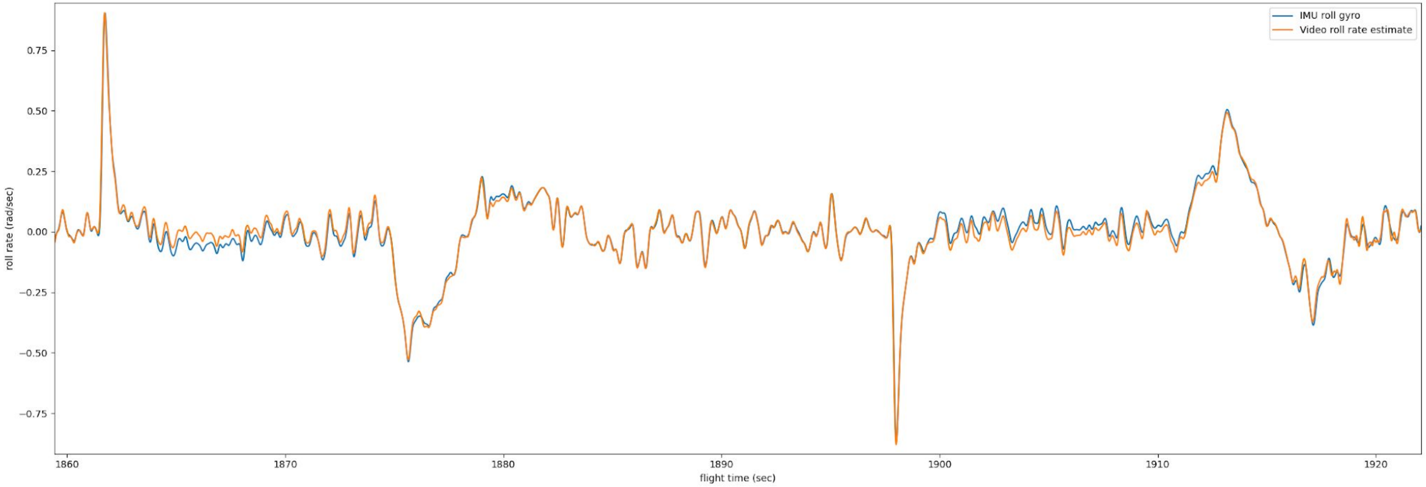
Horizon

1. Correct for lens distortion.
2. Extract the blue channel.
3. Canny edge detection on the blue channel.
4. Create an OTSU threshold on the blue channel.
   a. Segment connected components and rank them by size and position.
   b. Dilate the mask.
5. Use modified OTSU mask to eliminate unwanted canny edges.
6. Run HoughLines() on remaining edges.
7. Extract roll angle from HoughLines() result.
8. Measure angle from center of image to horizon line as pitch angle.

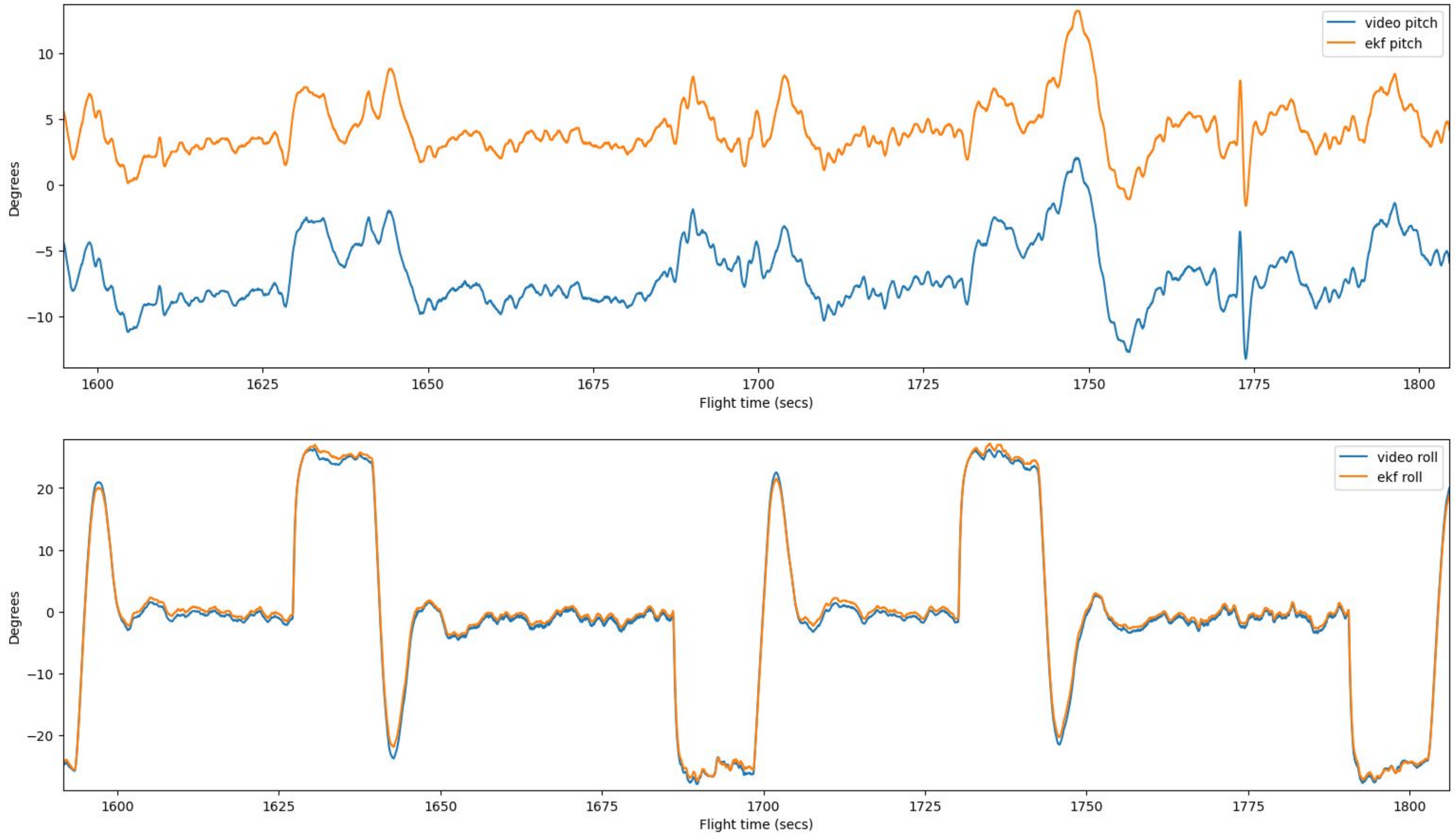
Challenging Scenarios

Correlating Video & Flight Data in Time

1. Estimate roll rate by differencing horizon roll angle between consecutive video frames.
2. Resample data to a common sampling rate.
3. Correlate video roll rate with flight data IMU roll gyro.



Camera orientation vs EKF orientation



Estimating Camera Mount Offset (v1.0)

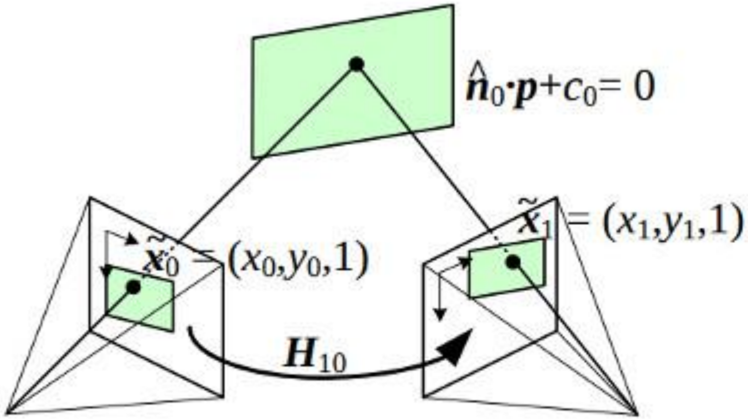
- Construct an optimizer to find a rotation transform that minimizes the error between EKF estimated roll/pitch angles (body frame) and observed roll/pitch angles (camera frame).
- We can also estimate the yaw offset of the camera (not just pitch, roll.)
- Disadvantage: the EKF attitude estimation errors get cooked into the camera mount transform.

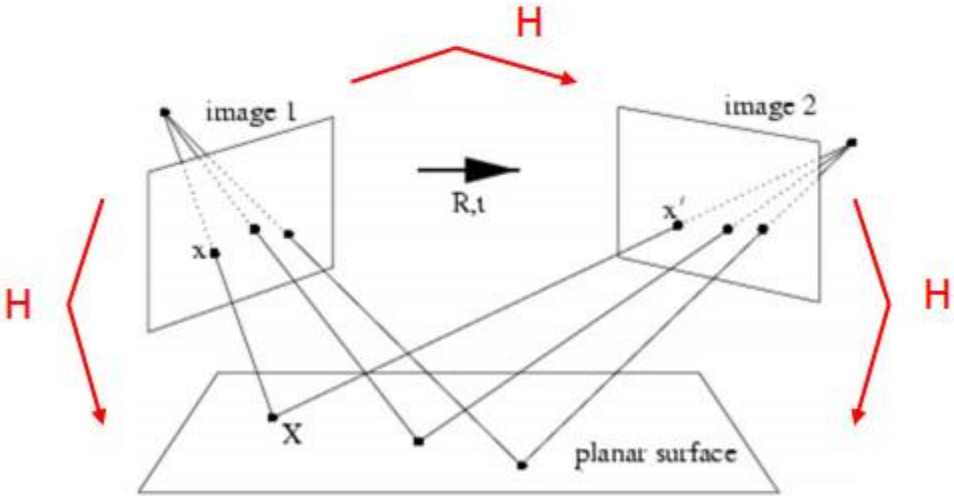
Estimating Camera Mount Offset (v2.0)

- Feature tracking (detection, matching)
- Fit a 2D affine transformation (4 dof) between point sets (feature motion.)
- Decompose into rotation (angle) and X, Y translation (pixels.)
- Rotation angle * fps = roll rate.
- X, Y translation (pixels) adjusted for camera field of view (* fps) = yaw and pitch rates.
- Forward flight motion biases pitch angle.
- Feature distribution in image space can bias X, Y translation.
- Yields and unsatisfying estimate. Can we do better?

Homography Transformation

- Perspective transformation between two planes.
- Decomposes into: plane normal vector, translation vector, rotation matrix.





Estimating Camera Mount Offset (v3.0)

- Feature tracking (detection and matching.)
- Fit a Homography transform (3x3 matrix) between source & dest planes.
  - Considers lens calibration (fov.)
  - Estimates rotation and translation of the plane (in the camera reference frame.)
  - Assumes matched features lie an a plane (usually ok.)
- Rotation matrix can be decomposed into Euler angles (Φ, Θ, Ψ)
- Euler angles * fps = (p, q, r) rates in the camera reference frame.
- Find the rotation transform that minimizes the error when projecting the aircraft body gyro (IMU) into the camera frame (virtual IMU.)

Mars Helicopter Technology Demonstrator

“Navigation (NAV) Camera. This is a global-shutter, nadir pointed grayscale 640 by 480 pixel sensor (Omnivision OV7251) mounted to a Sunny optics module. It has a field-of-view (FOV) of 133 deg (horizontal) by 100 deg (vertical) with an average Instantaneous Field-of-view (IFOV) of 3.6 mRad/pixel, and is capable of acquiring images at 10 frames/sec. Visual features are extracted from the images and tracked from frame to frame to provide a velocity estimate.”
https://rotorcraft.arc.nasa.gov/Publications/files/Balaram_AIAA2018_0023.pdf

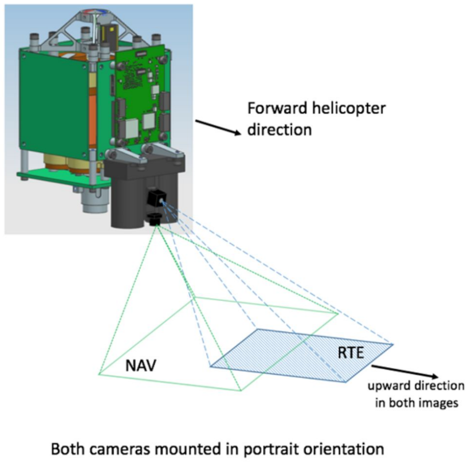


Fig. 11 Location and mounting of helicopter cameras.

Summary so far ...

- Quantified visual data from flight video.
- Aligned video & flight data in time.
- Estimated the camera offset relative to the aircraft body.

New things we can do now:

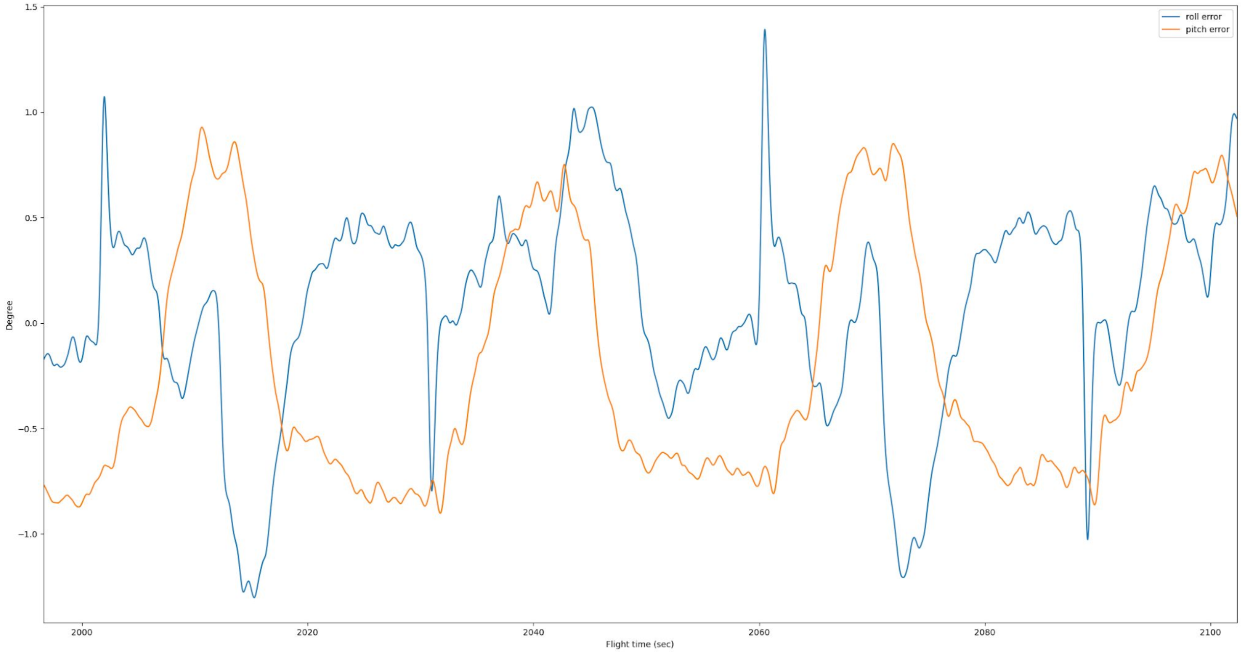
- Use observations in the video to validate flight data and EKF attitude estimates.
- Project flight data back into the camera frame of reference.

Abstract Map

Open Questions...

- How accurate is our estimate of the camera offset really?
- IMU gyro sensor alignment vs. IMU accelerometer sensor alignment?
- Are IMU axes exactly orthogonal?
- Noise, bias, outliers in visual feature matching?

EKF Roll & Pitch Errors (100 secs of flight)



Visualization

Horizon:

- Start with EKF attitude estimate.
- Apply correction $\rightarrow$ truth.
- Project into camera frame of reference: $R_{body \rightarrow cam}$
- Project onto image plane (pinhole camera model.)
- Draw.

Augmented Reality

Real world objects:

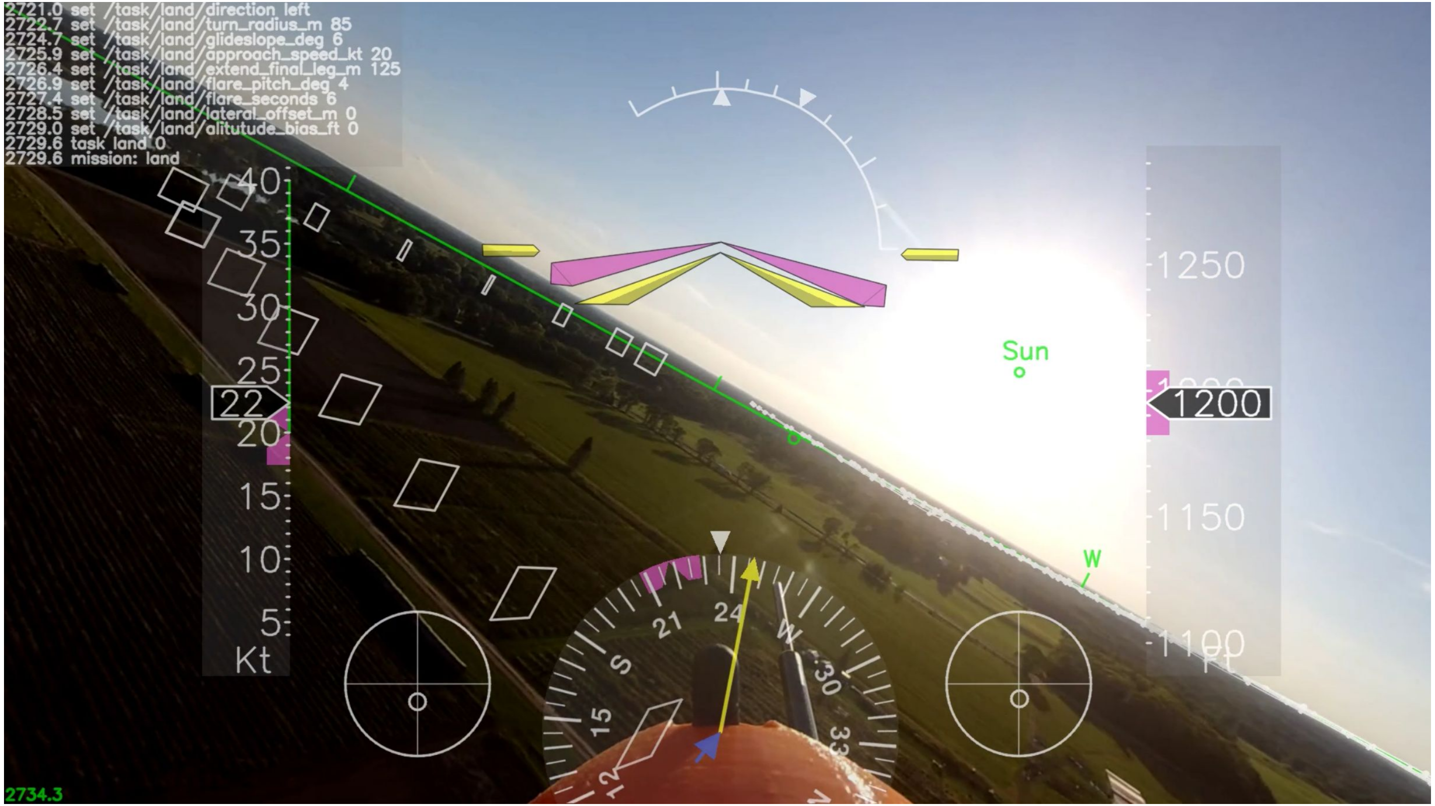
- Start with an LLA coordinate (or ned.)
- Convert to ned (if needed.)
- Project into (corrected) aircraft frame of reference.
- Project into camera frame of reference.
- Project onto image plane (pinhole camera model.)
- Draw.

Visualization (cont)

Things we can visualize:

- Conformal: horizon, pitch ladder, the bird, vbars (flight director), compass points, velocity vector.
- Non-conformal: bank angle, airspeed, altitude, heading, ground track, wind vector, events/commands, time.
- Augmented reality: flight path, sun, moon, shadow, known landmarks (ex: airport location/distance)
- Flight control system: target ground track, target bank, target pitch, target airspeed, target altitude, trajectory gates, effector commands.

Full HUD Visualization



Future Work

- Use our new visual “truth” reference to optimize EKF tuning (objectively evaluate the results of turning the gain knobs.)
- Fly with a camera pointed straight down to derive a yaw angle truth reference.
- Use map stitching strategies to construct a location and ned velocity truth reference.
- “Real-time” horizon tracking system as a backup to EKF (or as an input to EKF.)

Links

- Code (python/opencv): https://github.com/UASLab/ImageAnalysis/video/
- Videos: https://www.youtube.com/channel/UCwzx9ASLy_O3Zm5So6ARZjw