Visual Flare Assist for General Aviation Aircraft

Zouhair Mahboubi

Stanford, CA

CS231A (Winter 2015)

Motivation

- Airplanes are designed to fly
  - taking-off is the easy part
- Landing on the other hand is the most difficult part of flying
  - It takes on average 20 hours for a student pilot to solo
  - A big hurdle towards soloing is mastering the landing

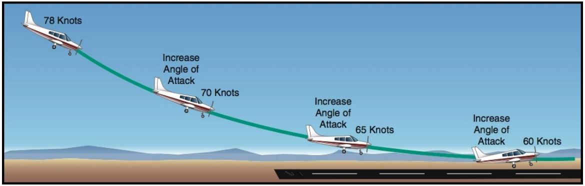
Motivation

- Airplanes are designed to fly
  - taking-off is the easy part
- Landing on the other hand is the most difficult part of flying
  - It takes on average 20 hours for a student pilot to solo
  - A big hurdle towards soloing is mastering the landing

Motivation

- Airplanes are designed to fly
  - taking-off is the easy part
- Landing on the other hand is the most difficult part of flying
  - It takes on average 20 hours for a student pilot to solo
  - A big hurdle towards soloing is mastering the landing

Landing is a controlled crash



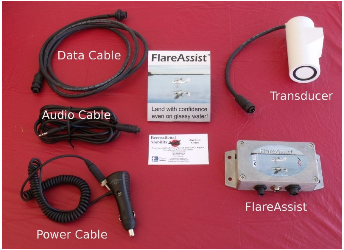
- The difficulty with landing is timing the flare (aka roundout)
- The airplane needs to be transitioned at the right altitude from a descending path to a horizontal path in which its slowing down
- During the slowing down, the airplane effectively stalls
  - Flare too low and you hit the ground too fast
  - Flare too high and you fall out of the sky
- Timing the flare at the right altitude takes practice

Technology to the rescue

- Large aircraft have radar altimeters to help the pilot and autopilot during landing
  - Often used in low-visibility conditions
  - Unfortunately these solutions are too expensive for small aircraft

Technology to the rescue

- Alternatively, some have considered using laser altimeters or sonars to estimate above ground altitude
  - Installing such sensors on small aircraft often requires certification
  - They can also be expensive (FlareAssist is $2500)

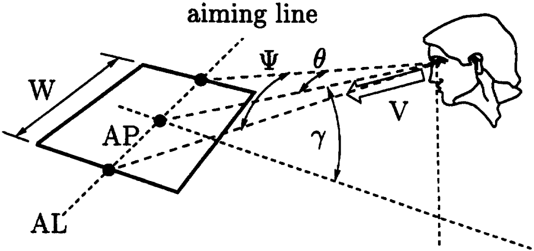


Proposed Solution

- Use computer vision to estimate the above ground altitude
  - This could be used as a dashboard solution if implemented on a phone
  - Would not require certification or new hardware and therefore cheap
- This project explores the feasibility of such a solution

Visual Angle

- This is effectively a problem of *single view metrology*
- We leverage the parallel runway edges to estimate camera pose
- We assume knowledge of runway width to remove scale ambiguity



Math Background

$$\tilde{k}_1 = tan(atan(k_1) - \hat{\phi}) \tag{1}$$

$$\tilde{k}_2 = tan(atan(k_2) - \hat{\phi}) \tag{2}$$

$$\hat{h} = \frac{\tilde{k}_1 \tilde{k}_2}{\tilde{k}_2 - \tilde{k}_1} \frac{cos(\hat{\theta})}{\cos(\hat{\psi})} B \tag{3}$$

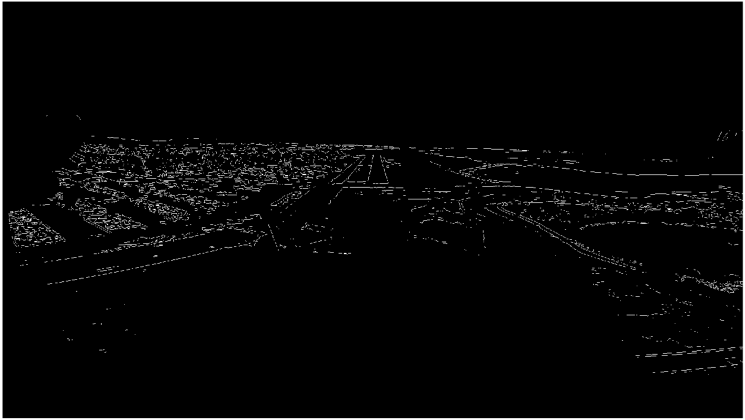
- $\hat{h}$ is the estimated altitude relative to the runway
- $\hat{\phi}, \hat{\theta}, \hat{\psi}$ are the roll, pitch and heading estimates of the aircraft.
- k_1, k_2 are the slopes of the runway edges
- B is the runway width

Data Collection

- Collected some video while flying in the Palo Alto pattern
- First the video was filtered by detecting and masking the static portion

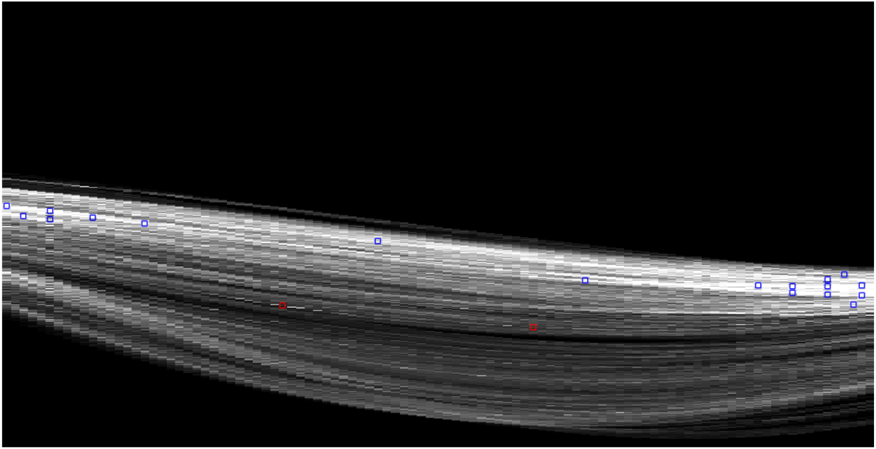
Edge Detection

- Ran an edge detection on the dilated/masked image
- Performed area opening to remove small blobs



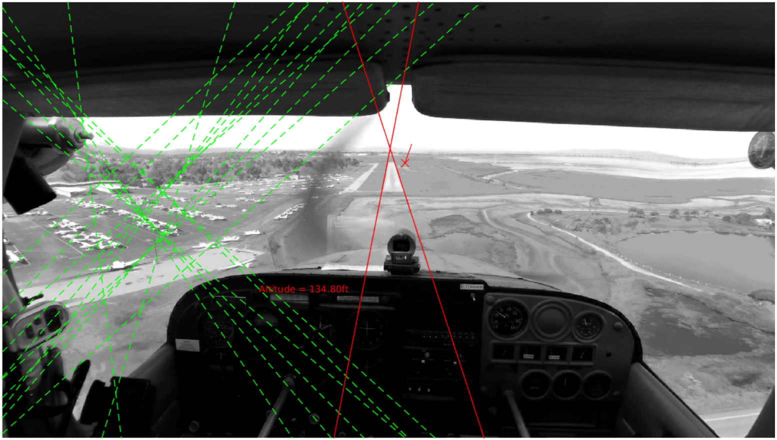
Hough Transform

- Performed Hough transform to detect lines and selected top 20
- Scored each pair of 20 lines (based on symmetry and other metrics)



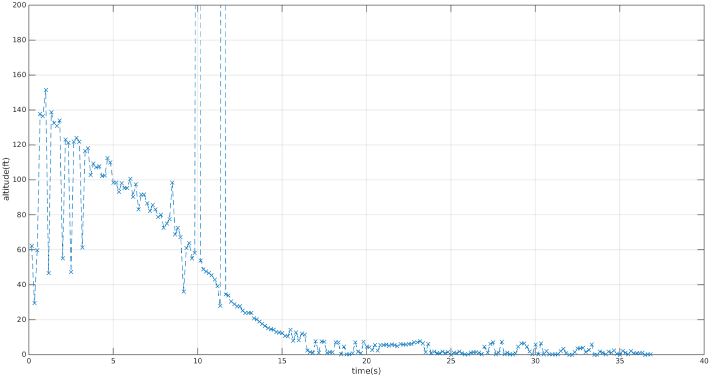
Processed Video

- Unfortunately did not have attitude data. Assumed level flight
- Was able to detect runway edges and estimate altitude [video link]



Extracted Altitude

- Extracted altitude from video sequence.
- A bit noisy due to false positives. Would benefit from Kalman filter
- Unfortunately did not have attitude data. Assumed level flight.



Questions?