Estimating Time and Location from solar shadows

- Krishna G

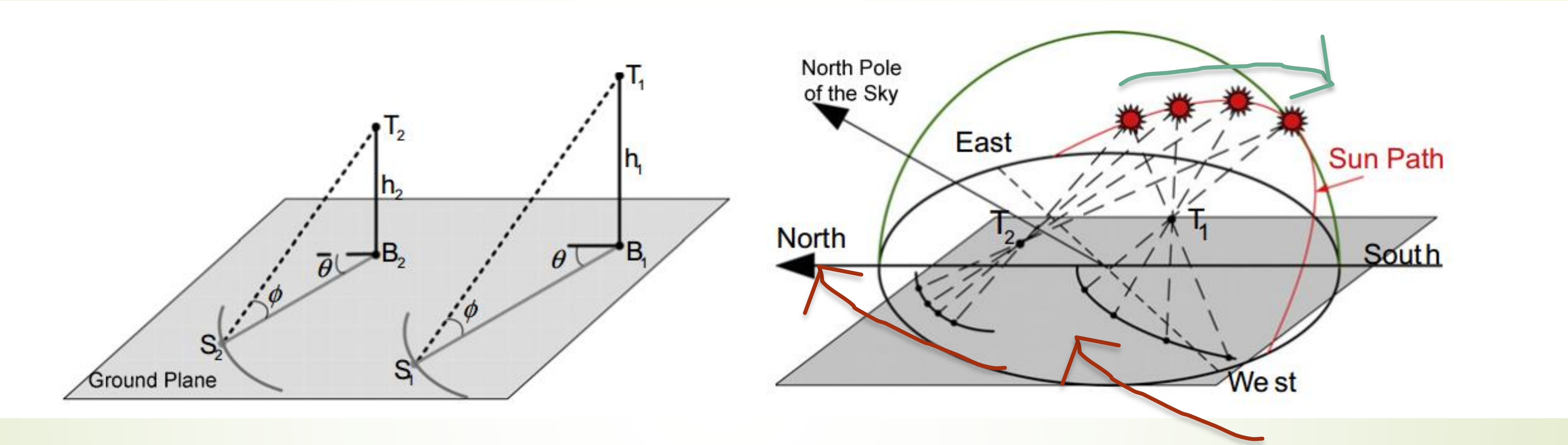
Introduction

- The market for using unmanned aerial vehicles (UAV) to inspect the utility transmission and distribution infrastructure is expected to reach $4.1B annually by 2024.
- Current civilian / commercial UAV's are vulnerable to many attacks, out of which GPS spoofing, GPS Jamming, Video Replay Attack, DeAuth Attack being a few.
- There's been a lot of research around detection / prevention of most of them. But, attacks related to video system are less researched.
- If UAV's are cyber attacked, there is a chance that Video Replay Attack is performed. There are no ways to detect it.
- In this project, I used Solar shadows of poles to estimate location and time.

Related Work

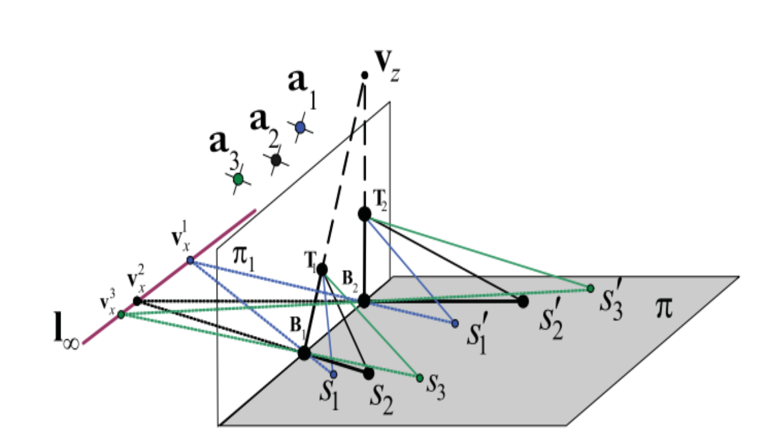
- Junejo, Imran N., and Hassan Foroosh. "GPS coordinates estimation and camera calibration from solar shadows." *Computer Vision and Image Understanding* 114.9 (2010): 991-1003.
- The above paper implemented the use of shadows, to geo-locate static cameras.
- This project aim is to implement it and extend it to UAV footage.

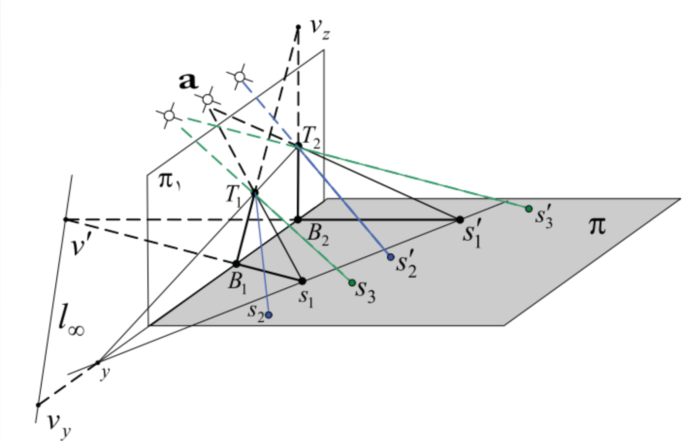
Theory



- Relatively sun moves 1^0 for approximately 4 minutes w.r.t earth
- For a single frame altitude angle of sun can be calculated, but its not enough to determine time (it might be on either side) and loc.
- Two frames, that can be used to differentiate the change by a significant amount will be able to determine time, but not location.
- We need at least 3 frames to determine location. More frames would reduce the noise effect and make it robust.

Theory





- Above fig. shows rays, shadow points, P_{inf} for all three frames.
- Calculating Altitude Angle (ϕ):

$$\cos\phi_i = \frac{v_i'^T \omega v_i}{\sqrt{v_i'^T \omega v_i'}\sqrt{v_i^T \omega v_i}}$$

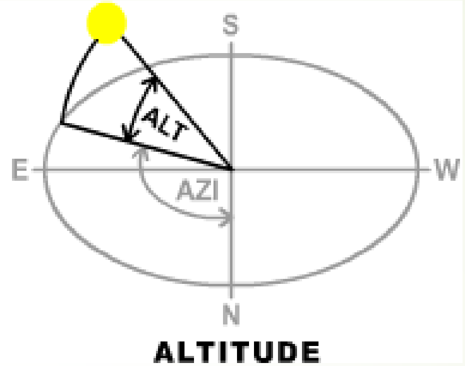
$$\sin\phi_i = \frac{v_z^T \omega v_i}{\sqrt{v_z^T \omega v_z}\sqrt{v_i^T \omega v_i}}$$

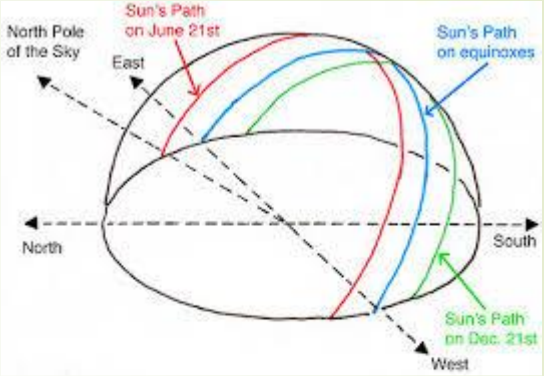
v_z is P_{inf} from the poles
$v_i{}'$ is P_{inf} from lines joining base points and shadow endpoints
v_i is P_{inf} from base lines and shadow endpoint lines
ω is Image of Absolute Conic (IAC) $\sim K^{-T}K^{-1}$

Theory

- Calculation Time:

- Altitude angle ranges from
  0 (Sun rise) -> 180 (Sun set)
- Dividing the angles into segments and finding the segment to the angle found earlier. From the choices, select the time based on the shift.





- Calculating azimuth angle (ψ):

$$\cos\psi_i = \frac{v_i'^T \omega v_x}{\sqrt{v_i'^T \omega v_i'}\sqrt{v_x^T \omega v_x}}$$

$$\sin\psi_i = \frac{v_i'^T \omega v_y}{\sqrt{v_i'^T \omega v_i'}\sqrt{v_y^T \omega v_y}}$$

Where v_x and v_y are vanishing points

Theory

- Calculation of Latitude (λ):

$$\lambda = \tan^{-1}(\rho_1 cos\alpha + \rho_2 sin\alpha)$$

Where $$\alpha = \tan^{-1}\left(\frac{\rho_1 - \rho_3}{\rho_4 - \rho_2}\right)$$ and

$$\rho_1 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_1 \cos\psi_1}{\sin\phi_2 - \sin\phi_1}$$

$$\rho_2 = \frac{\cos\phi_2 \sin\psi_2 - \cos\phi_1 \sin\psi_1}{\sin\phi_2 - \sin\phi_1}$$

$$\rho_3 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_3 \cos\psi_3}{\sin\phi_2 - \sin\phi_3}$$

$$\rho_4 = \frac{\cos\phi_2 \cos\psi_2 - \cos\phi_3 \cos\psi_3}{\sin\phi_2 - \sin\phi_3}$$

- Calculating Day Number:

$$N = \frac{365}{2\pi} \sin^{-1}\left(\frac{\delta}{\delta_m}\right) - N_0$$

Where N_0 = 284 : number of days from first equinox to Jan 1st

$\delta_m = 23.45°$: max. absolute declination angle of earth

δ : declination angle, calculated from Eqn.

$$(sin\delta)^2 - 2 * sin\lambda * sin\delta + (cos\lambda)^2 * ((cos\phi sin\theta)^2 - 1) + (sin\phi)^2 = 0$$

Theory

- Calculating longitude(γ):

$$\gamma = \gamma_0 + (\hbar - \hbar_0)$$

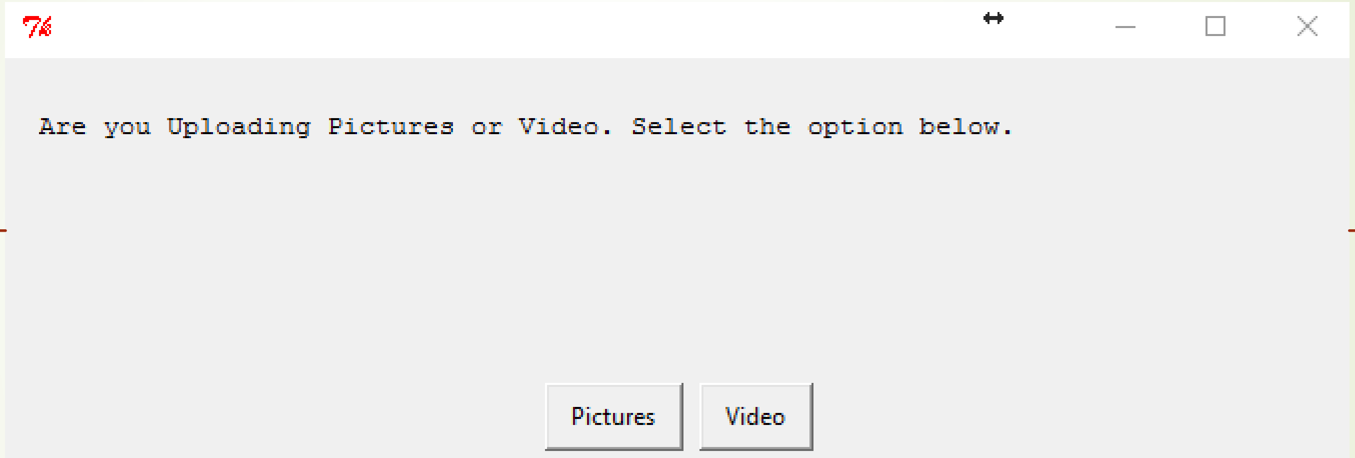
Where γ_0, $\hbar_0$ are known location longitude and hourangle. In here Greenwich longitude is considered for easy information (Link provided to user during input).

$\hbar$ is the local hour angle, can be calculated from equation

$$\sin \hbar \cos \delta - \cos \phi \sin \theta = 0$$

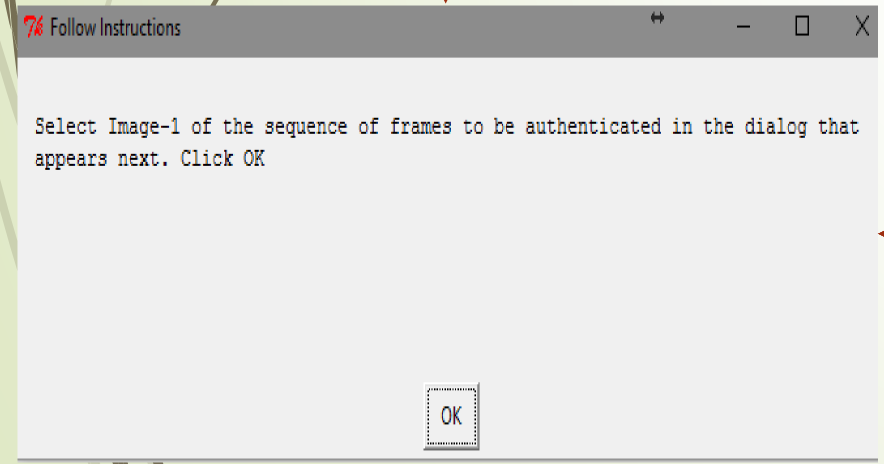
Implementation

- It has been implemented as a system, that prompts step-by-step instructions to user. On running, a prompt will be shown for user to select.



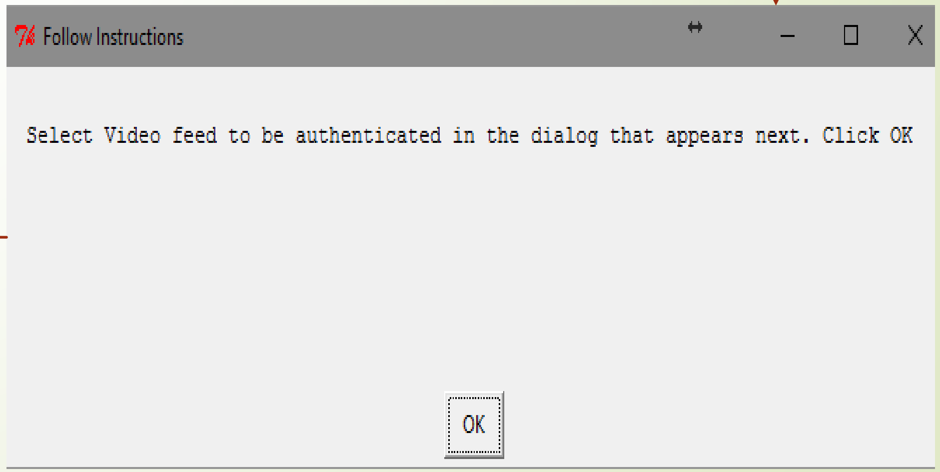
Pictures

Video



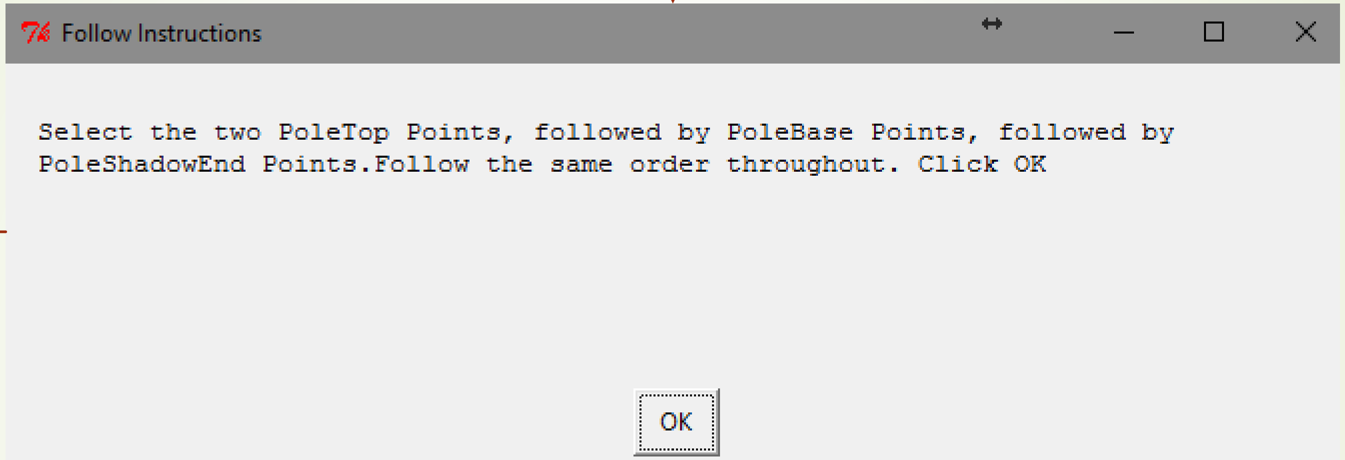
After converting video to frames

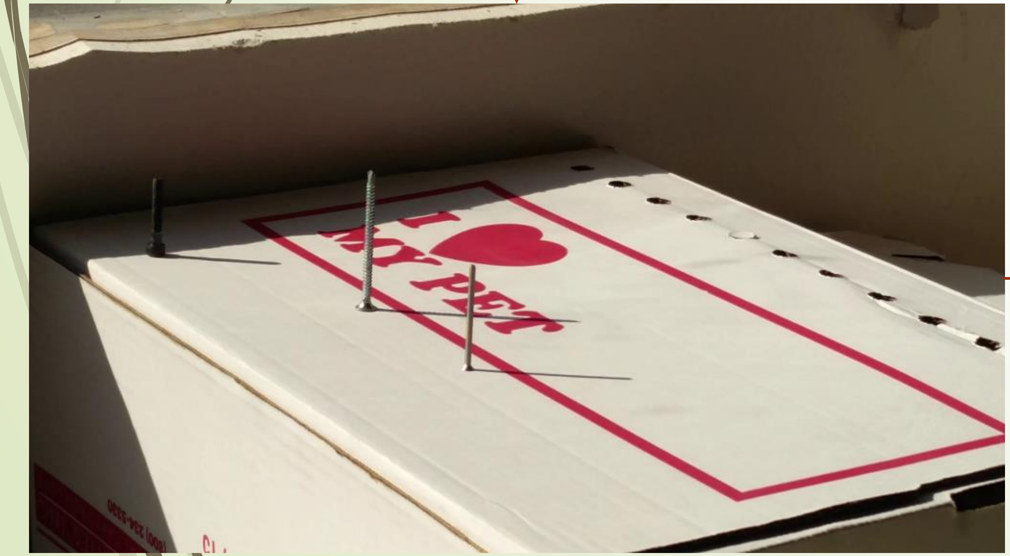
Location saved will be shown in a GUI before selection



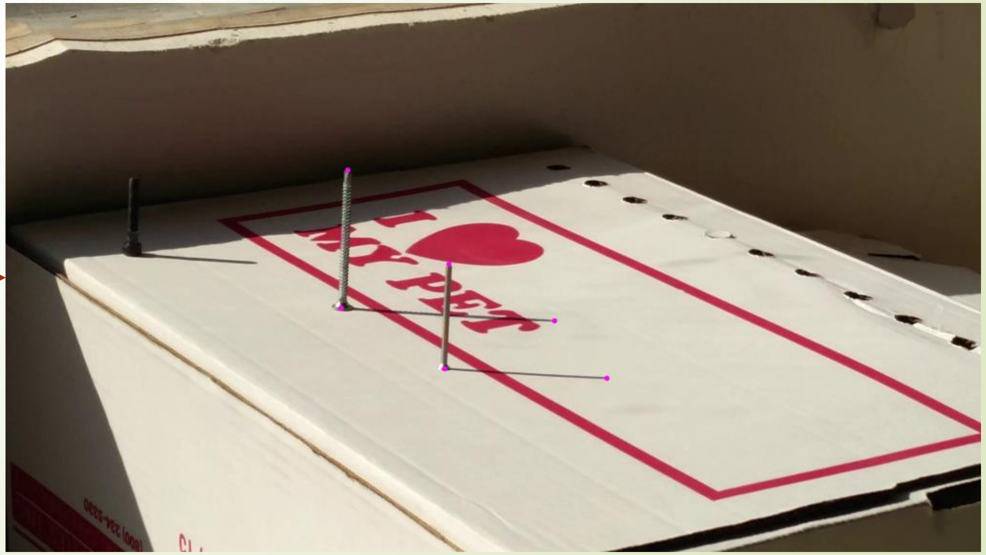
Implementation

After selecting three images

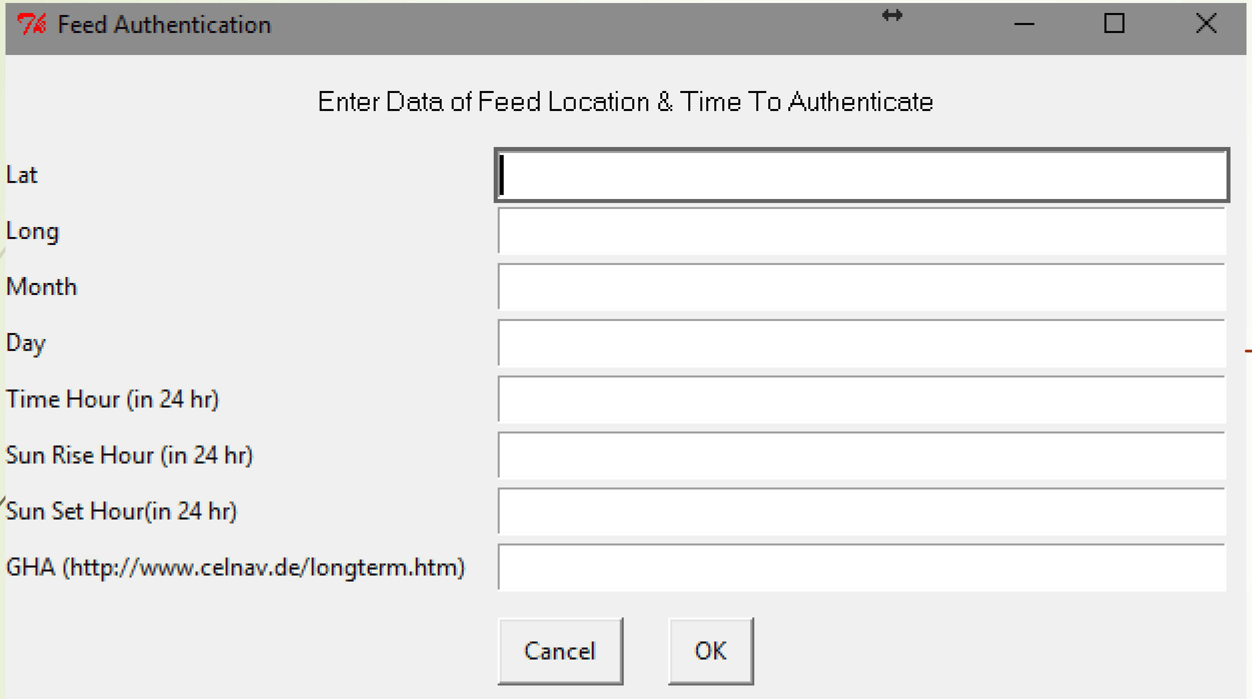




Select points (MouseClick and ‘b’ after done)



Implementation

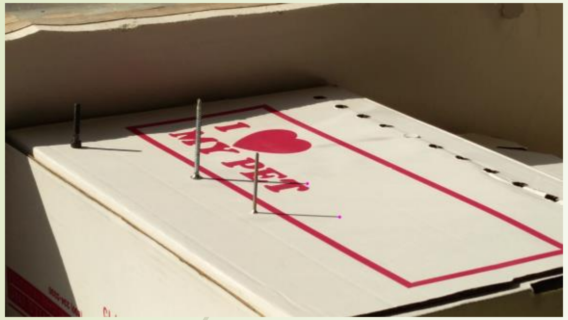
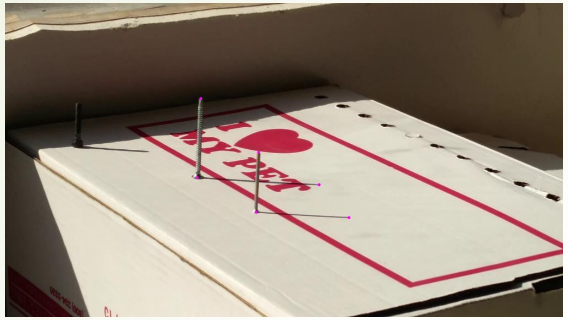
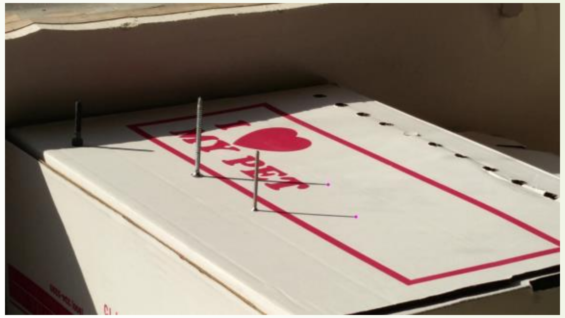
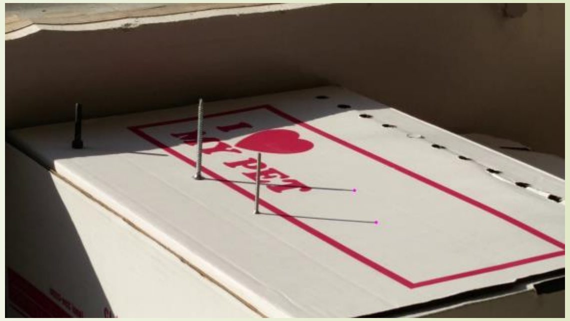


Result will be shown

User Input: ['30.618980', '-96.338946', '5', '6', '16.8', '6.5', '20', '75']

34.0467033927 -107.679037278 2017-04-22 17.290372062

Results

User Input: ['30.618980', '-96.338946', '5', '6', '16.8', '6.5', '20', '75']

34.0467033927	-107.679037278	2017-04-22	17.290372062
36.3358702944	-106.615763538	2017-04-25	17.1387713033
44.7230300356	-129.219302294	2017-04-10	17.332688675

These images are frames from the video taken by using an app @ 1 frame / min

Results

Modified the original to create another system to just determine time.

User Input: ['10.5', '7.5', '19.5']
9.4573336891
15.0067650854
Single Image is not sufficient to know time, it can be either

Saturday, March 25, 2017, 10:24:46 AM

Conclusion

- Flight time of an UAV is around 30 min. (DJI Phantom 3 – 25 min)
- The images that can be analyzed should be at least 4-5 min apart -> 12-15 min of static video from the UAV.
- Using most of the flight time to authenticate the feed is not an optimum way of using resources.
- Video stabilization cannot be applied, since it might crop the original size of the image.
- Homography might not help, since there is a chance that it might rectify shadow the wrong way.
- In conclusion, this implementation is not reliable and optimum way to authenticate the feed.