MECHANICAL ENGINEERING

Image Processing

Image Attributes

(NOTE: in this lecture I'll use the term "sensor" collectively to imply a camera & telescope apparatus)

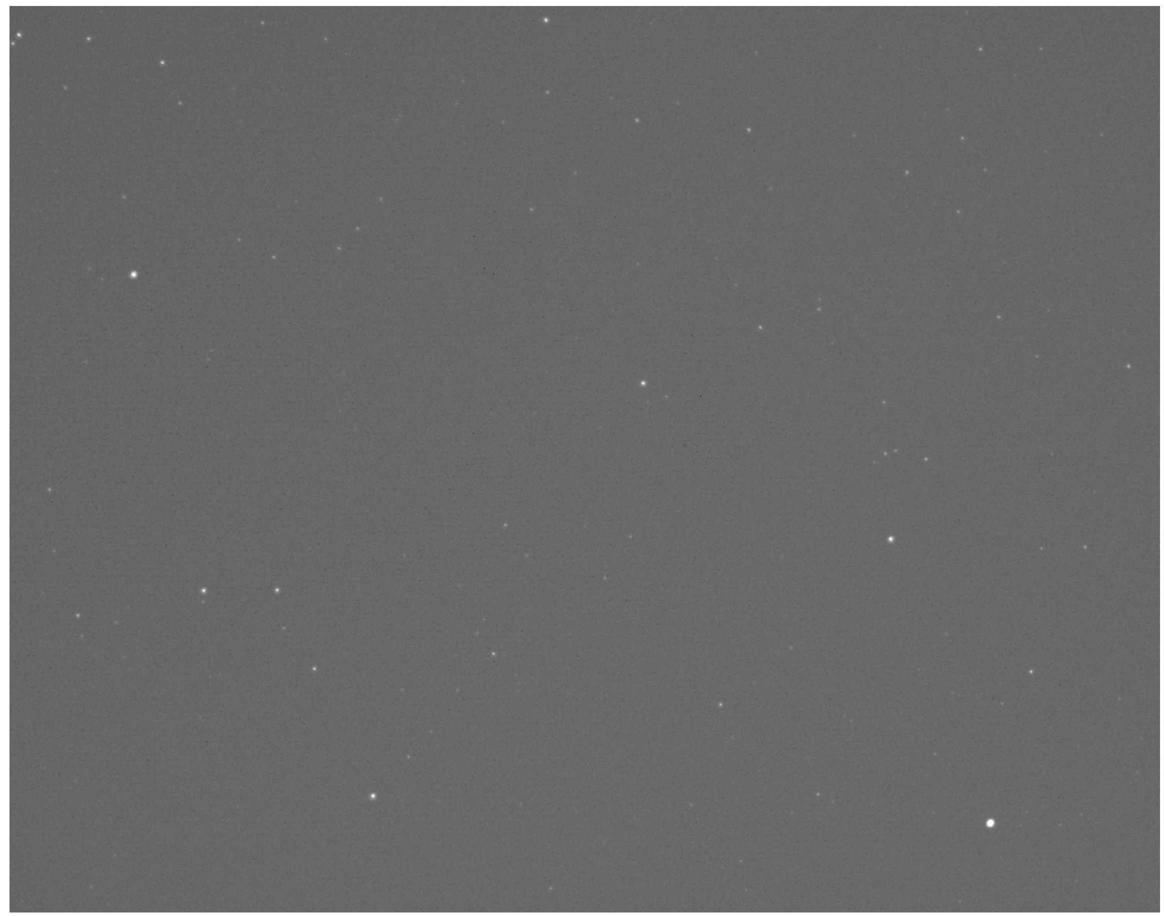
- In Module 3, we discussed some basic optics:
  - Focal length
  - Image plane → pixel dimensions
- We talked about how, once an object is identified in an image by its pixel coordinates, we can calculate LOS
  - Camera frame → TH frame → ECEF frame → ECI frame
- Image processing is the precursor to orbit determination
  - Identifying objects of interest in an image & calculating LOS measurements to input to the IOD/POD process

Image Attributes

- There are 2 fundamental facets to image processing for obtaining LOS of an object from image data:
  - Identification: determining which objects in the image are of interest for SSA (i.e. “non-stars”)
  - Registration: determining the orientation of the image relative to inertial space
- One crucial element of the identification step is observation assignment
  - It’s not enough just to ID a non-star in an image, but need to ID a non-star in several images AND attribute all of these observations to the same object
  - Difference between obtaining one LOS vs several LOS’s
- We’ll explore Identification & Registration one at a time

Object Identification

- Consider the following (real) space image from an optical sensor



- Each illuminated object is represented by a collection (or "blob") of pixels
- The vast majority of these (if not all) are likely stars
- Are there are Earth-orbiting objects in the image, & if so, how can we tell?

Object Identification

- First, we must be mindful of 2 attributes of our imaging campaign:
  - Imaging mode: is our sensor terrestrial or space/based? Is it fixed in a particular frame or moving?
  - Exposure time (or integration time): how long does our sensor capture photons of the scene?
- There are 3 common imaging modes for ground sensors
  - TH mode: sensor remains fixed in its TH frame, pointed at the same orientation (Az & El) for the entire image (e.g. zenith → directly upward)
  - Inertial mode: sensor slews (rotates) so as to point in the same direction in inertial space for the entire image (e.g. keeping the boresight fixed on a particular star)
  - Tracking mode: sensor slews so as to follow a particular object (normally a satellite as it passes over)
- Typically, spaced-based sensors can support inertial or tracking mode

Object Identification

- Because imaging involves a finite integration time, objects in the scene will move to varying extents while photons are collected

- For each mode, let's think about which objects in the image should move during the course of the image & which should not

Object Identification

Inertial mode:

- If perfect inertial pointing by the sensor is achieved, any objects distant enough to appear fixed in inertial space (e.g. outside the solar system: stars & other deep-space celestial bodies) will not move
- For closer-range objects (e.g. inside the solar system: planets and Earth-orbiting objects), their apparent movement depends on their proximity to the sensor vs length of integration time → Earth-orbiting objects more likely to show movement than, say, planets
- Keep in mind that because no sensor is fixed in inertial space, perfect inertial pointing is impossible
    - The more accurately we know a sensor's inertial motion, the more accurately we can account for it to keep the sensor inertially "aimed" → more difficult for space-based than terrestrial sensors
    - "Jitter" in the motor device (e.g. drive mount) that slews the sensor
    - Result is that everything in the image will move to some slight extent (but inertially fixed objects still move the least)

Object Identification

Tracking mode:

- Normally the goal of tracking mode is to point directly at a specific object, i.e. keep it on the boresight
- To do this, the operator must determine the path of the object in the camera frame & move the sensor along this path during the integration time
  - For an Earth-orbiting object, this would entail knowing the object's orbit & slewing the sensor to follow that (again, in the camera frame)
- If this is done perfectly, the object being tracked will (obviously) not move, but everything else in the image likely will move
  - An exception is an object that may be co-orbital with the object being tracked → since both objects will move at the same rate, the sensor will effectively track both objects
- As with inertial pointing, errors will cause everything to move slightly
  - In addition to pointing error, also error due to imperfect knowledge of the tracked object's orbit

Object Identification

TH mode:

- Here the sensor remains pointed at the same direction (Az/El) in the sky
- In this case, everything should be moving in the image, with one exception:
  - GEO objects (for sensors powerful enough to see them) will not move, because they always remain at the same place in a viewer's local sky

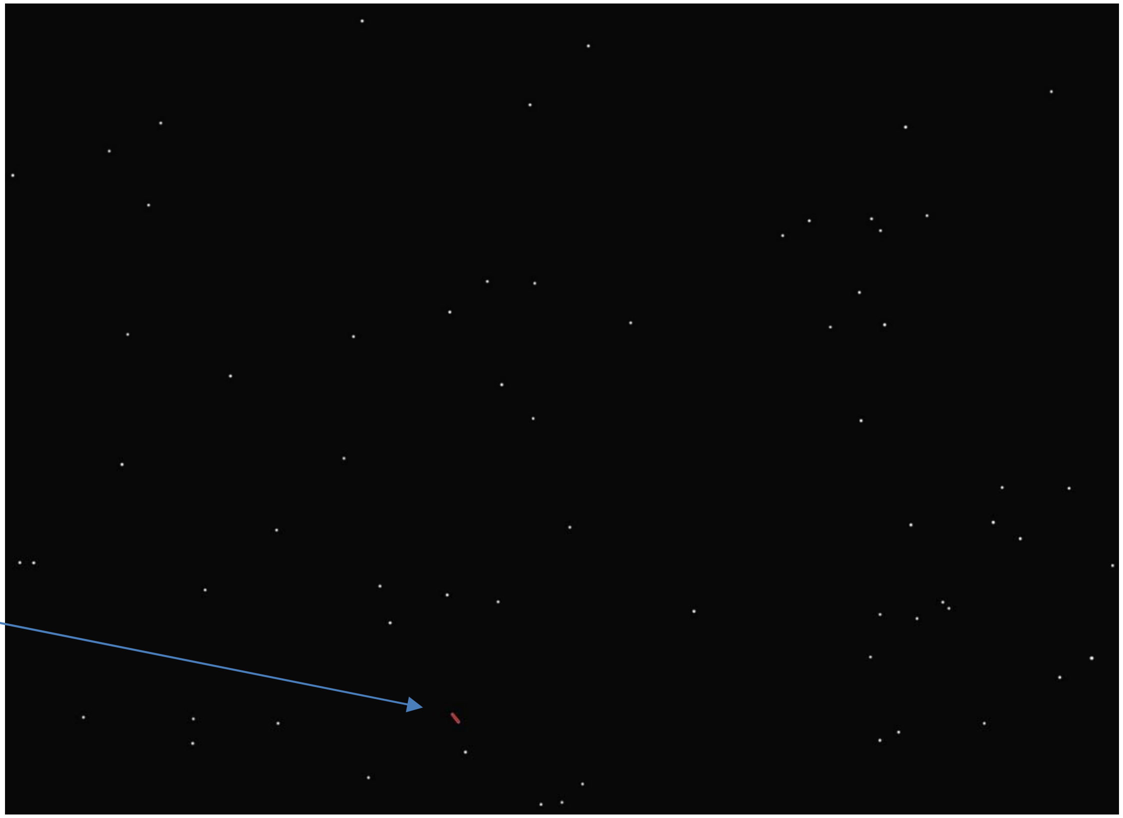
Object Identification

- For a long enough integration time, any object not stationary in an image will appear to streak across the image

- Here is an example of an image taken in inertial mode

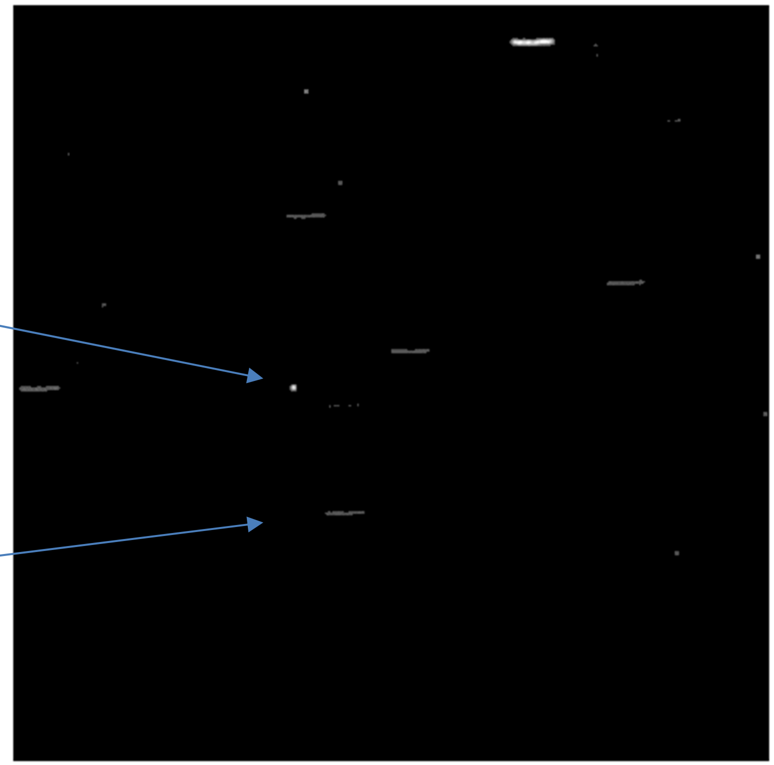
- Majority of objects are stars, which remain stationary

- Earth-orbiting object can be seen streaking near bottom of image (shown in red for clarity)



Object Identification

- Here is an example of an image taken in object tracking mode
- Object being tracked remains stationary (& near the boresight)
- Stars streak (& all at the same rate, i.e. streaks are the same length)

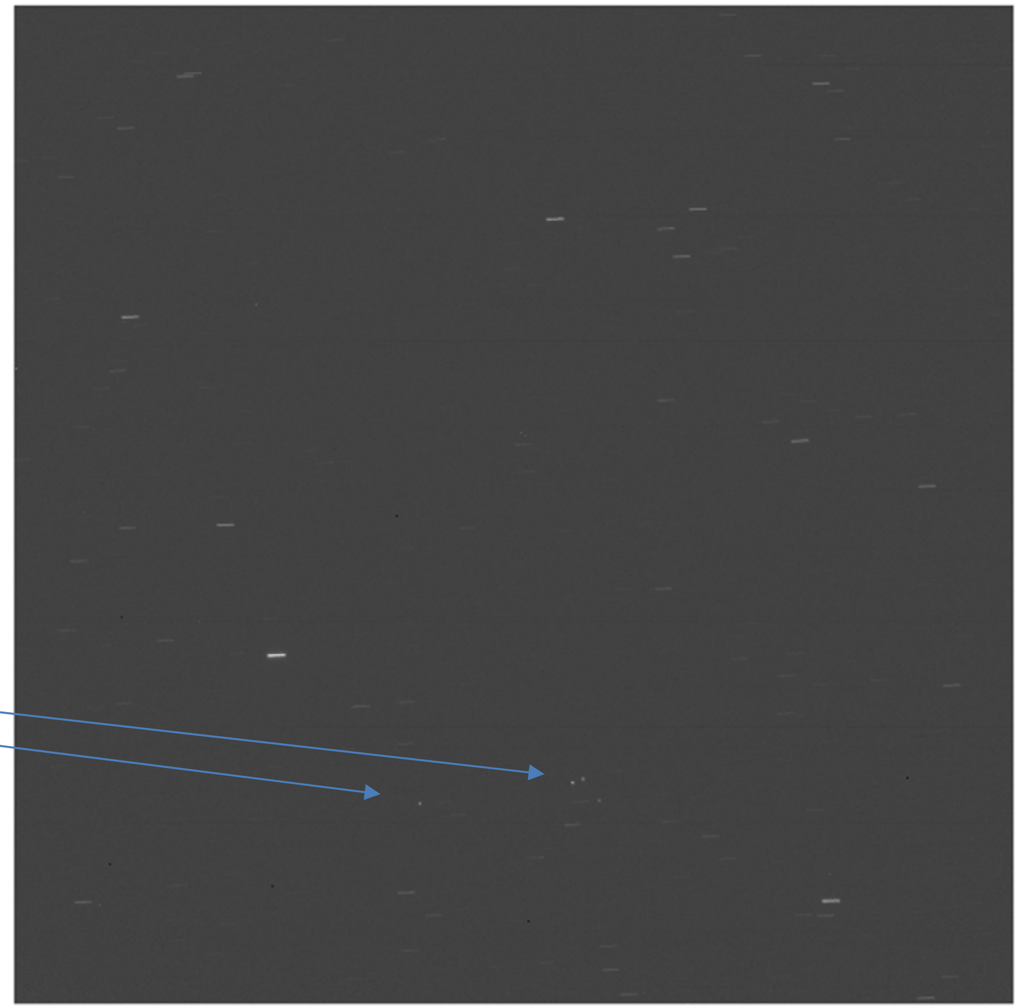


Object Identification

- Here is an example of an image taken in TH mode, looking toward GEO

- GEO objects are stationary

- Stars streak (again all at the same rate)



Object Identification

- The goal of object identification, from an SSA perspective, is to determine the stars from the non-stars (i.e. Earth-orbiting objects)
- Note how streaking can be quite helpful to this goal
- But how much integration time is required to achieve streaking?
- This varies depending on the sensor's location & the object(s)' orbit(s); this determines both:
  - How much a particular object will streak in a given time while in inertial mode
  - How much the stars will streak in a given time while following a particular object in tracking mode
- For short integration times, no streaking will be evident

Object Identification

- For some scenarios, the integration times required to generate noticeable streaking (even by an automated routine) may be more than the sensor can support

- A viable option may be to instead take a series (or set) of images

- The goal is then to look for frame-to-frame movement among the objects in the images → de facto streaking

Object Identification

- Here is an example of an image set taken in inertial mode
- Majority of objects are stars, which remain stationary from frame to frame
- An earth-orbiting object can be seen "streaking" (in the frame to frame sense) across the field of view

Object Identification

- Here is an example of an image set taken in tracking mode

- Stars are moving from frame to frame

- Earth-orbiting object can be seen stationary on the boresight (shown in red for clarity)

Object Identification

- The preceding development shows how streaking &/or frame to frame motion can facilitate manual (human-in-the-loop) object identification

- But it is preferable to automate this process

- There are instances where the sensor is not in inertial or tracking mode → difficult for a human to properly discern such images

- Even in inertial or tracking mode scenarios, automation will process the images more quickly & accurately

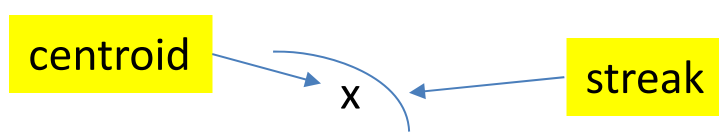
Object Identification

- First step in automating the process is centroiding, i.e. replacing each raw image with a set of object locations in the camera frame → how to do this?
- To answer this, consider some fundamental aspects of imagery:
  - An image consists of the set of intensity values in each pixel
  - Arranged as a matrix whose dimensions are determined by the image plane width & height (# of pixels)
  - For example, if a sensor's image plane is 1500 pixels long by 1000 pixels high, the image data is stored in a 1000 x 1500 matrix
  - What values do these intensities take, & at what granularity? → determined by "bit depth" e.g. from 0 to 255 (discussed briefly in Module 3)

Object Identification

Centroiding process essentially consists of the following steps:

- Thresholding: zeroing out all intensities below a chosen threshold value (to eliminate noise)
- Edge detection: determining each collection of illuminated pixels that constitutes an "object"
  - Do these edges represent the physical boundaries of each object? → NO, due to point spread function (PSF) → results in blurring or "blooming" of an object beyond its actual size
- Centroid calculation: calculating the geometric centroid of each object
  - Note that if an object moves/streaks in an image, in a curved path, the centroid location may not lie along the path at all!



Object Identification

- Next step after centroiding is observation assignment → correlating each object across each image
- Which centroid (or "dot") in Frame 1, Frame 2, etc, belong to Object 1? And which collection of dots across each frame belong to Object 2? Etc…
- Ability to successfully associate depends heavily on frame-to-frame motion
  - Generally, the less change in centroid locations from one frame to the next, the easier association will be
  - Inertial mode image sets are normally simple to associate → each centroid representing a star changes its location little (or none) from frame to frame → any centroid that changes by "x" amount or more must be an Earth-orbiting object
  - Most image sets require some kind of data association filter technique to properly associate centroids

Registration

- The result of centroiding & data association (if successful) is a tracklet for each Earth-orbiting object found in the image set

- Each data point in the tracklet corresponds to an (x, y) location in the camera frame → goal is to transform these (x, y) coordinates to ECI coordinates to feed into the orbit determination process

- For a ground sensor, this is done through the series of frame rotations (DCMs) discussed in Module 3 & reviewed on Slide 2
  - Camera frame → TH frame → ECEF frame → ECI frame

- For a space-based sensor, the TH & ECEF frames are not involved & the transformation is performed via attitude determination: determining the DCM from the vehicle's body frame to the ECI frame
  - Often the sensor is fixed on the vehicle, so that either the camera & body frames are identical or their DCM is a known constant
  - Series of DCMs is then Camera frame → Body frame → ECI frame

Registration

- But what if the Camera-to-ECI transformation is not accurately known?
  - For space-based sensors, attitude determination error pervades the Body-to-ECI transformation, while the Camera-to-Body transformation can be affected by misalignment of the sensor on the vehicle bus
  - For ground-based sensors, their pointing direction (Az & El) may not be accurately known (Camera-to-TH transformation)

- In such instances, a viable option is the process known as registration: determining the orientation of a sensor (i.e. which direction it is pointing) at the time it takes an image, based on features in the image

- For space images, the features analyzed are stars → pattern recognition algorithm known as star matching

Registration

- What registration software packages exist?

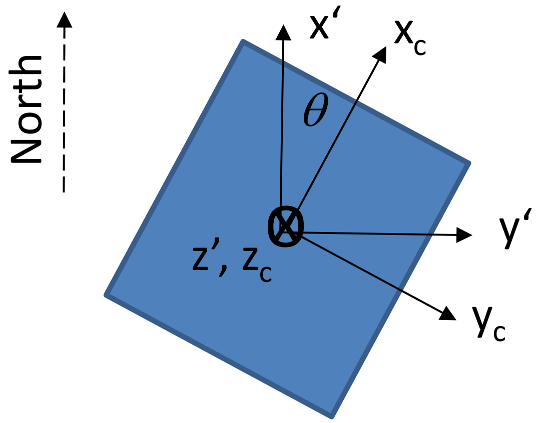
- Some spacecraft perform attitude determination via registration, using a star tracker to take images → generally proprietary

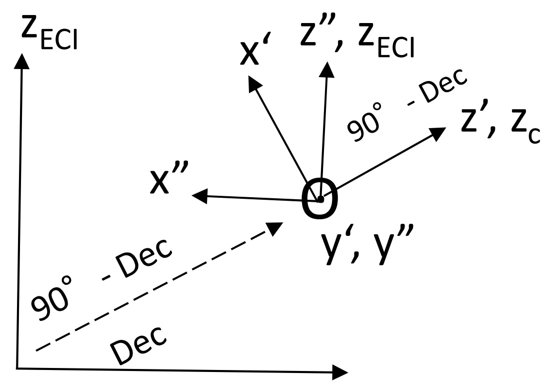
- Astrometry is an open-source package developed by astronomers (available on astrometry.net)
  - Matches star patterns in an image with patterns in its internal database, based on a star catalog (RA/Dec/magnitude for millions of stars)
  - Output is inertial pointing direction (RA/Dec) of the boresight & rotation angle of the image plane θ (relative to celestial North)

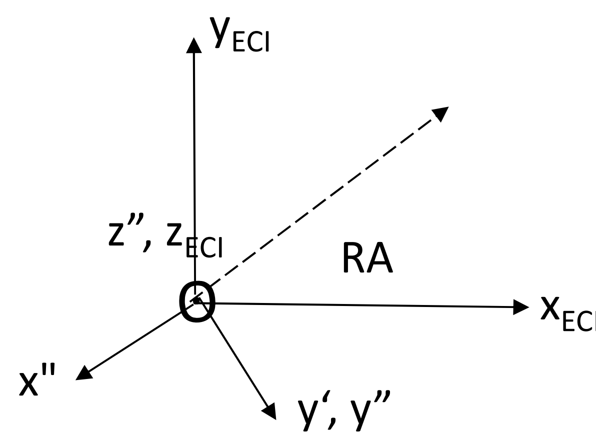
Registration

- RA/Dec/θ represents the relationship between the camera frame orientation ("right-down-forward" as discussed in Module 3) & ECI frame orientation; can be converted to a DCM

- Consider a sensor whose frame (x_c , y_c , z_c) corresponds to a particular RA/Dec/θ :







Begin with a rotation about z_c through ($-\theta$)

Followed by a rotation about y' through (90° - Dec)

Then a rotation about z_{ECI} through (180° - RA)

⊗ Into the page ⊙ out of the page

Registration

- So, given the coordinates (*x*, *y*) of an object in the sensor frame, we know from Module 3 that the sensor-to-object position vector is proportional to $\bar{r}|_{CAM} = [x\ y\ f]^T$ (where f is focal length)

- Then $\bar{r}|_{ECI} = Q_{CAM}^{ECI}\bar{r}|_{CAM}$ where

$$Q_{CAM}^{ECI} = R_3(180 - RA)R_2(90 - Dec)R_3(-\theta)$$

(R_2, R_3 are Euler rotation matrices discussed in Module 2)

- And the RA & Dec of the sensor-to-object position vector are

$$RA = tan^{-1}\left(\frac{y_{ECI}}{x_{ECI}}\right) \qquad Dec = sin^{-1}\left(\frac{z_{ECI}}{L}\right)$$

$$L = \sqrt{x^2 + y^2 + f^2}$$

These values should be fairly close to the RA & Dec values returned by Astrometry (which are the RA & Dec of the boresight)

End-to-End SSA Process

- In this course, we've learned many critical components of the SSA Process
  - Coordinate systems
  - Sensing
  - Image processing
  - Orbit determination

- Putting it all together in an end-to-end diagram, optical SSA would look something like this:

Epilogue

- This is the last official lecture of the course
- Next week will mainly be time to prepare for (& take) the Final Exam
- I will provide some material on the topic of Orbital Debris