National Aeronautics and Space Administration

Geometry-based Observability Metric

Colin Eaton

Bo Naasz

code 595

November 2, 2012

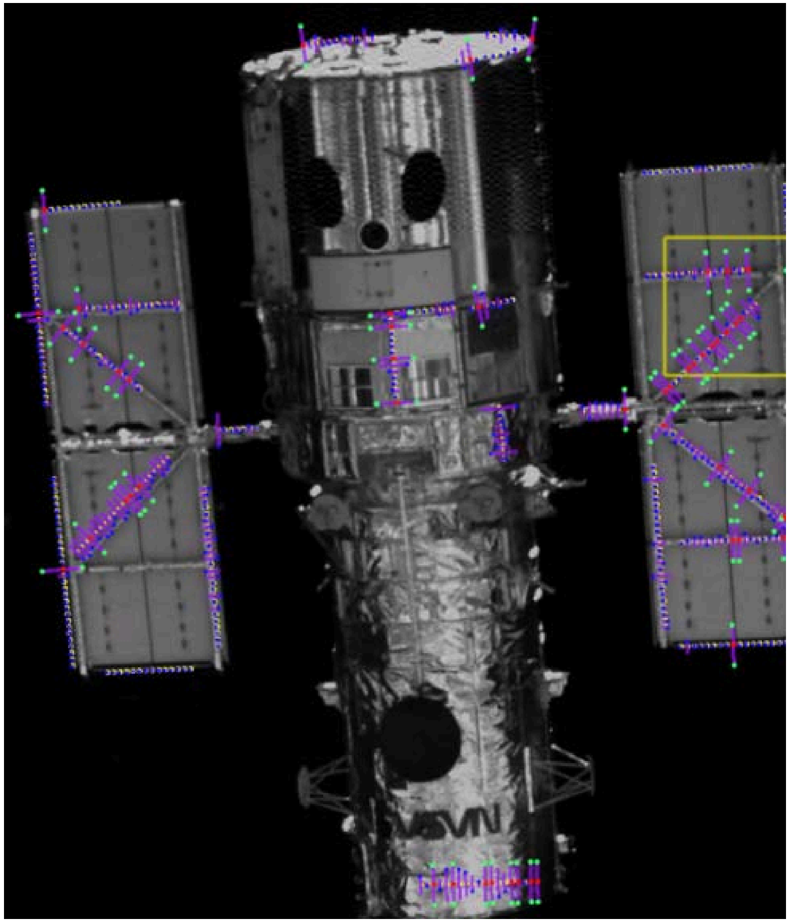
NAVIGATION & MISSION DESIGN BRANCH
NASA GSFC

Agenda

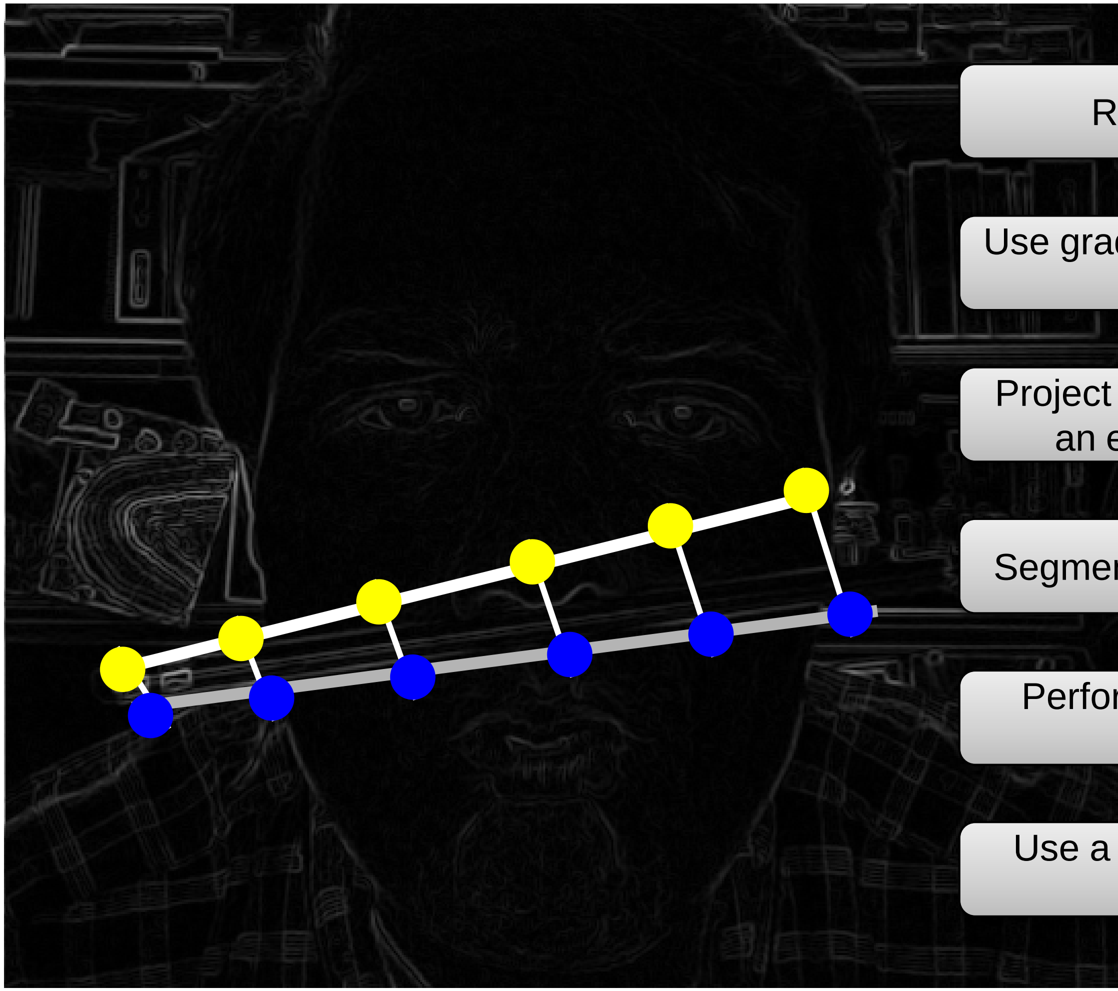
- Background Info
- How GNFIR Works
- Concept and Motivation
- Relevant Equations
- Strategy
- Simplified Problem Details
- Expanding to 6DOF Problem
- 6DOF Problem Results
- Conclusions and Next Steps

Background Info

- The Satellite Servicing Capabilities Office (SSCO) is currently developing and testing Goddard's Natural Feature Image Recognition (GNFIR) software for autonomous rendezvous and docking missions
- GNFIR has flight heritage and is still being developed and tailored for future missions with non-cooperative targets
  - DEXTRE Pointing Package System on the International Space Station
  - Relative Navigation System (RNS) on the Space Shuttle for the fourth Hubble Servicing Mission (*shown in figure*)



How GNFIR Works



Receive camera imagery

Use gradient filter to obtain gray-scale image

Project an *a priori* pose estimation of an edge into the image frame

Segment the edge into control points

Perform a search in the predicted edge normal direction

Use a least squares fit to minimize search distances

Concept and Motivation

- Hypotheses:
  - A metric of the target vehicle's pose observability can be determined (prior to receiving imagery) **solely as a function of the target's edge model and planned trajectory**
  - Certain parameters used in the Lie Algebra of GNFIR's edge-fitting algorithm will serve as a useful quality metric of the state estimation error

- Motivations:
  - Geometry-based quality metric
    - Used in relative navigation filter (RNF) to weight state estimates
    - **Analog metric,** as opposed to the binary metric currently used (edges found vs. expected)
  - Tool for creating **better models and trajectories**
    - Computationally efficient method can be used in Monte Carlo analysis to **improve overall system performance**

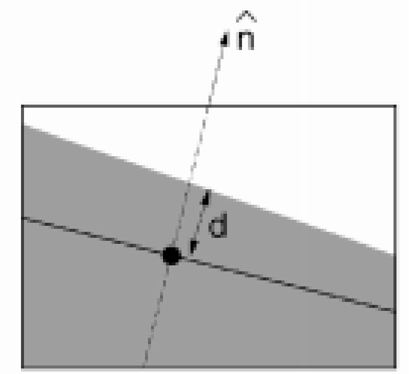
Lie Algebra Equations

$$C_{ij} = \sum_{\xi} f_i^{\xi} f_j^{\xi}$$

$$f_i = L_i \cdot \hat{n}$$

$$v_i = \sum_{\xi} d^{\xi} f_i^{\xi}$$

$$\alpha_i = C_{ij}^{-1} v_j$$



Equations and figure from Drummond and Cipolla (see ref's slide)

Variables

$C_{i\,j}$: projection matrix mapping the observed edges to the model

L_i : true image coordinates of features w.r.t. the i^{th} Lie group generator

n : edge normal direction (*see figure*)

f_i: edge-normal motion observed w.r.t. the i^{th} Lie group generator

d : scalar distance from control point to detected edge (*see figure*)

v_i : vector displacements projected into tangent space

α_i : linear approximation of the state residuals which minimizes the error between the model and observed edges

*** C^{-1} is the candidate for the observability metric**

Strategy

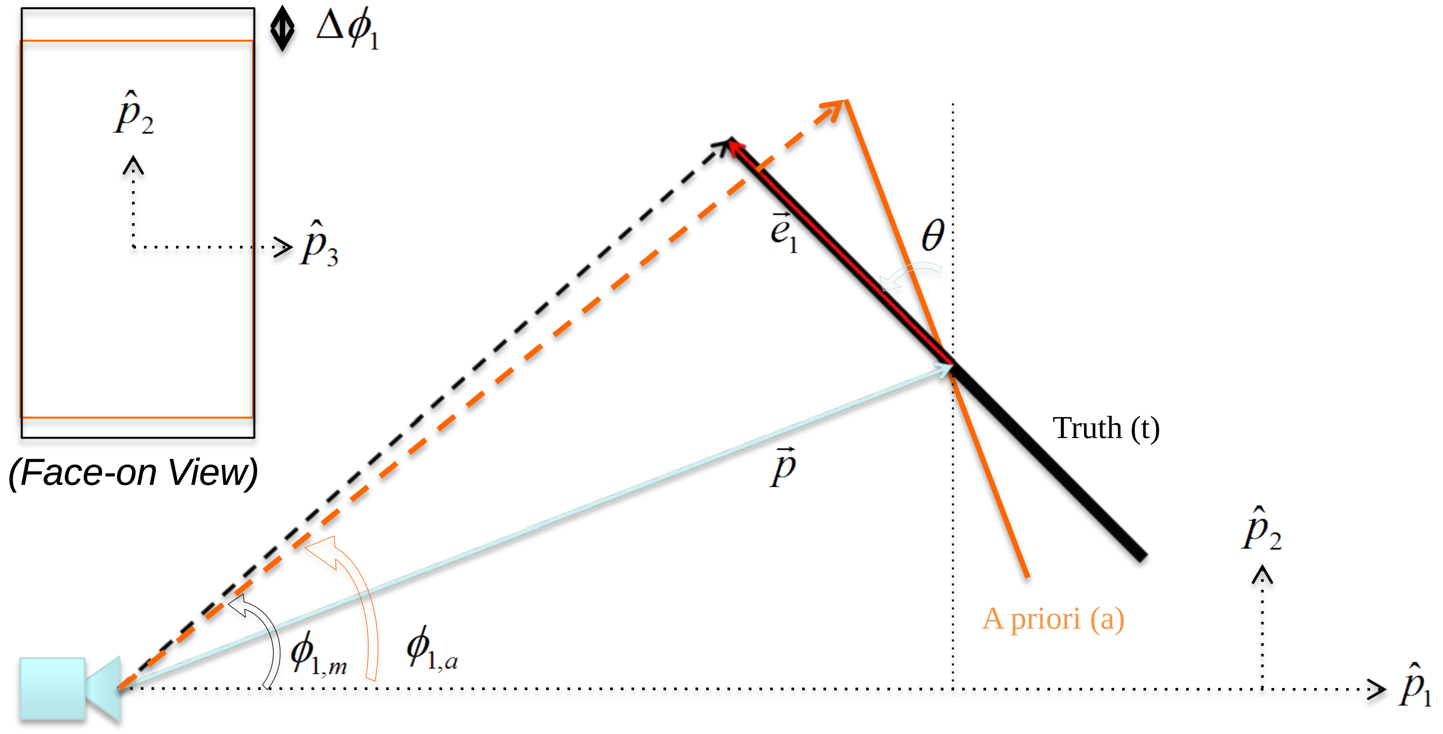
- Simplify the problem
  - Take away degrees-of-freedom (DOF) to a level which is more easily conceptualized
  - Manually derive the nonlinear measurement equations
  - Linearize these equations and apply standard least squares estimation
  - Disturb relevant parameters and observe resulting observability trends

- Expand knowledge to the full 6DOF problem
  - Make connections between simplified and 6DOF parameters
  - Verify that the same trends in observability are detected

- Identify how the exploits can be used to enhance the model and trajectory formulation

Simplified Problem: Set-up

- Flat plane aligned in roll direction (p_1) to camera
- Movement constrained to 3DOF: p_1, p_2, θ
- Arbitrary number of edge features (e) at arbitrary distances from center of model
- Measurements are the angular differences (Δφ) between *a priori* and measured edges (similar to "d" in comprehensive problem)



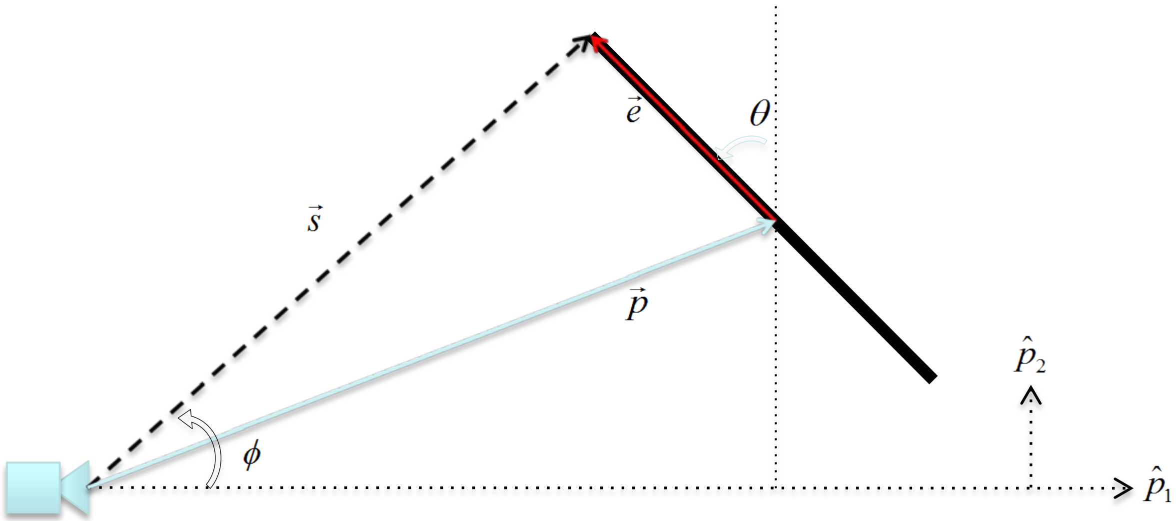
Simplified Problem: Equations

Nonlinear Measurement Equations:

$$\phi_i(\vec{p},\theta) = \arccos(\hat{s}_i \cdot \hat{p}_1)$$

$$\vec{s}_i = \vec{p} + \vec{e}_i = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + R_1(\theta)\begin{bmatrix} e_{i,1} \\ e_{i,2} \end{bmatrix}$$

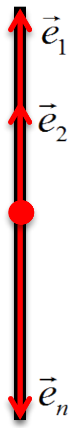
$$\phi_i(\vec{p},\theta) = \arccos\left(\frac{p_1 + e_{i,1}\cos\theta - e_{i,2}\sin\theta}{\sqrt{(p_1 + e_{i,1}\cos\theta - e_{i,2}\sin\theta)^2 + (p_2 + e_{i,1}\sin\theta - e_{i,2}\cos\theta)^2}}\right)$$



Simplified Problem: Procedure

- Define model by number and location of edges

Model: $\begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \cdots \\ \vec{e}_n \end{bmatrix}$



- Provide random *a priori* state close to truth state
- Generate measurements (z) on each edge using nonlinear equations and zero-mean Gaussian white noise (v)
- Compute linearized projection matrix (H) to satisfy:

$$z = Hx + v$$

Simplified Problem: Linear LS Estimation

- Linearize equations about the *a priori* state:

$$\vec{p}_1 \rightarrow \vec{p}_{1,a} + \Delta p_1$$
$$\vec{p}_2 \rightarrow \vec{p}_{2,a} + \Delta p_2$$
$$\theta \rightarrow \theta_a + \Delta\theta$$

- Derive projection matrix (H):

$$H = \frac{\partial \phi_i(\vec{p}_a, \theta_a)}{\partial \Delta x} \qquad \Delta x = \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \Delta\theta \end{bmatrix}$$

$$\Delta\hat{x} = (H^T H)^{-1} H^T z \qquad z = \begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \dots \\ \Delta\phi_n \end{bmatrix}$$

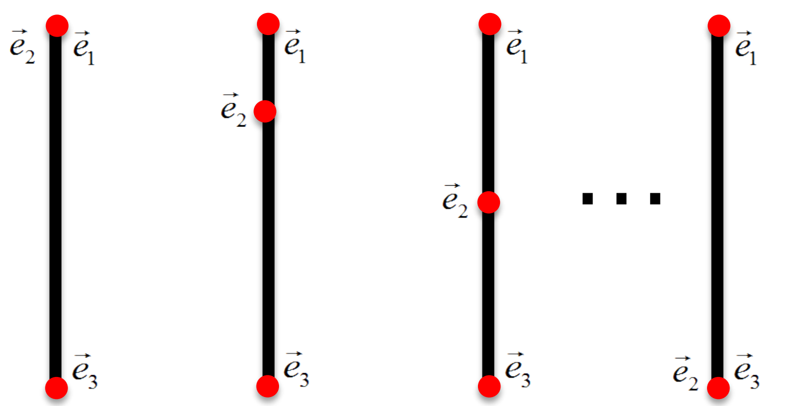
- Obtain state estimate:

$$\begin{bmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} p_{1,a} \\ p_{2,a} \\ \theta_a \end{bmatrix} + \Delta\hat{x}$$

Simplified Problem: Proof of Concept

- Hypothesis:
  - Model is more observable when features are spread farther apart – i.e. better geometry
- Test:
  - Use 3 edges per model (to be fully observable)
  - Slide the middle edge between the top and bottom edge
  - Average over 100 cases with different measurement noises and *a priori*

Models: (#1-3,11)



Expected Observability: Worst   Better   Best   · · ·   Worst

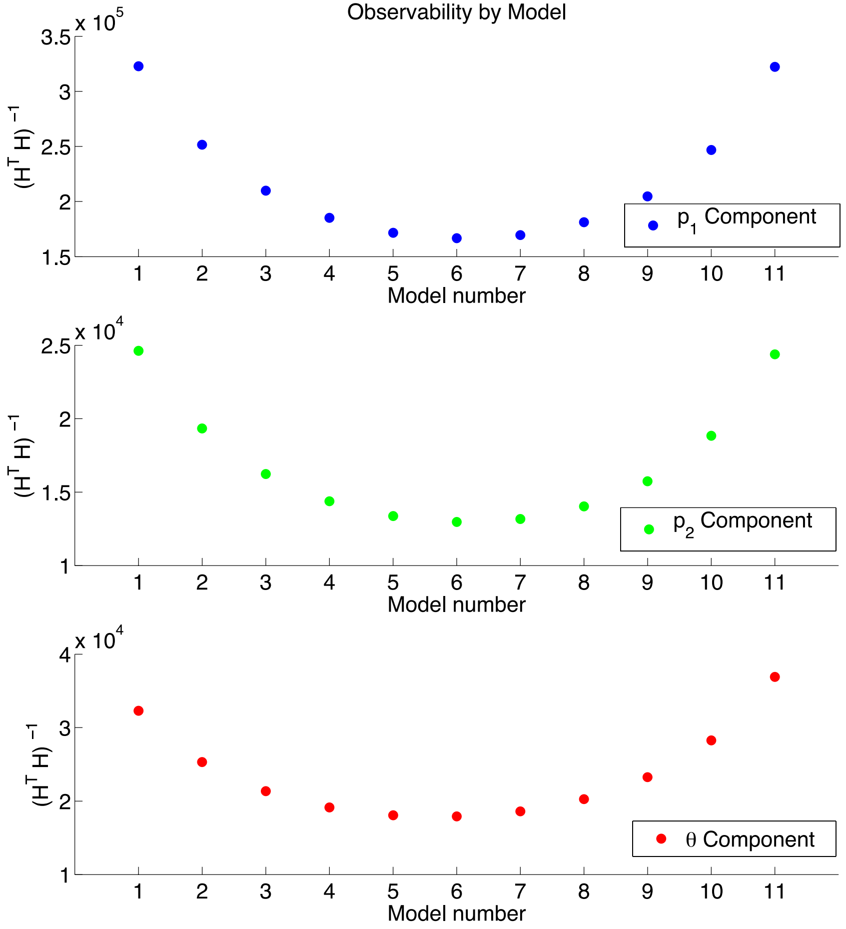
Truth state:

$$\begin{bmatrix} p_{1,t} \\ p_{2,t} \\ \theta_t \end{bmatrix} = \begin{bmatrix} 10m \\ 3m \\ 10^0 \end{bmatrix}$$

Measurement Noise:

$$\begin{bmatrix} \mu \\ \sigma \end{bmatrix} = \begin{bmatrix} 0m \\ 0.01^0 \end{bmatrix}$$

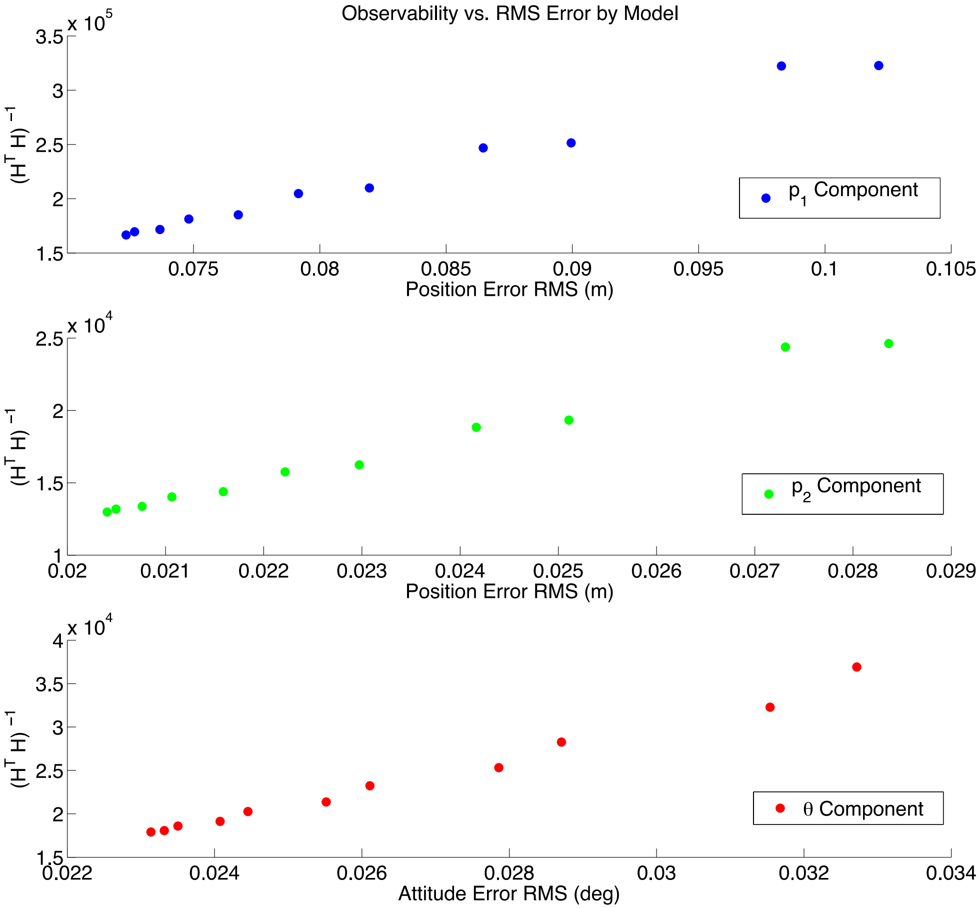
Simplified Problem: Proof of Concept



**Note: better observability is a lower HTHI*

- Observability is best when the edge features are most spread out (model 6)
  - True for both position DOF's (top) and attitude DOF (bottom)
- Proves that this model behaves as expected and is an appropriate representation of the 6DOF problem
  - Trends in observability and estimation error should be comparable

Simplified Problem: Trends



- Each DOF portrays a linear relationship between the observability and the estimation errors
  - Comparing component-by-component
- Should expect to see this same correlation in the 6DOF representation

Expanding to 6DOF Problem

Simplified problem: $\Delta\hat{X} = (H^T H)^{-1} H^T Z$

Full problem: $\alpha_i = C_{ij}^{-1} v_j \approx (f \cdot f)^{-1} \cdot (f \cdot d)$

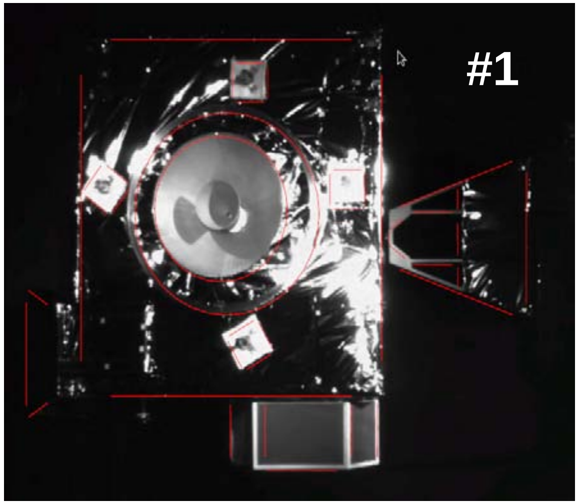
Parallel to GDOP: $Q = \left(A^T A\right)^{-1}$

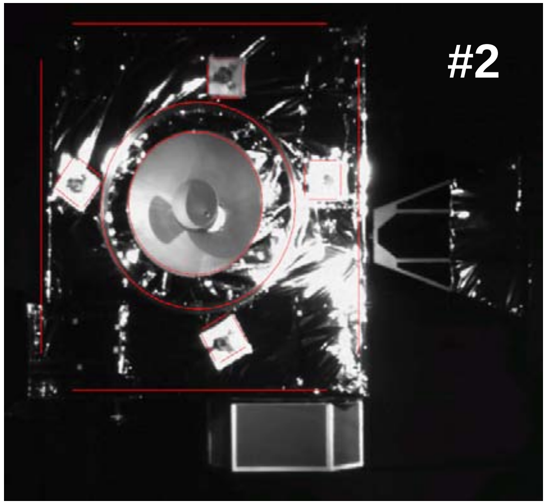
$$Q = \begin{bmatrix} d_x^2 & d_{xy}^2 & d_{xz}^2 & d_{xt}^2 \\ d_{xy}^2 & d_y^2 & d_{yz}^2 & d_{yt}^2 \\ d_{xz}^2 & d_{yz}^2 & d_z^2 & d_{zt}^2 \\ d_{xt}^2 & d_{yt}^2 & d_{zt}^2 & d_t^2 \end{bmatrix}$$

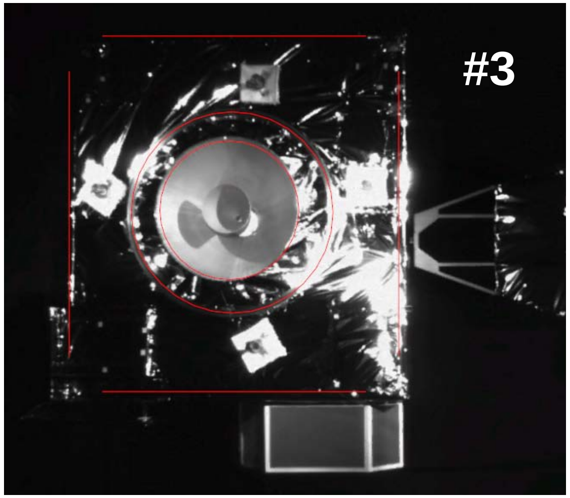
$$PDOP = \sqrt{d_x^2 + d_y^2 + d_z^2}$$

We should see the same trends in C^{-1} as those in HTHI

6DOF Models

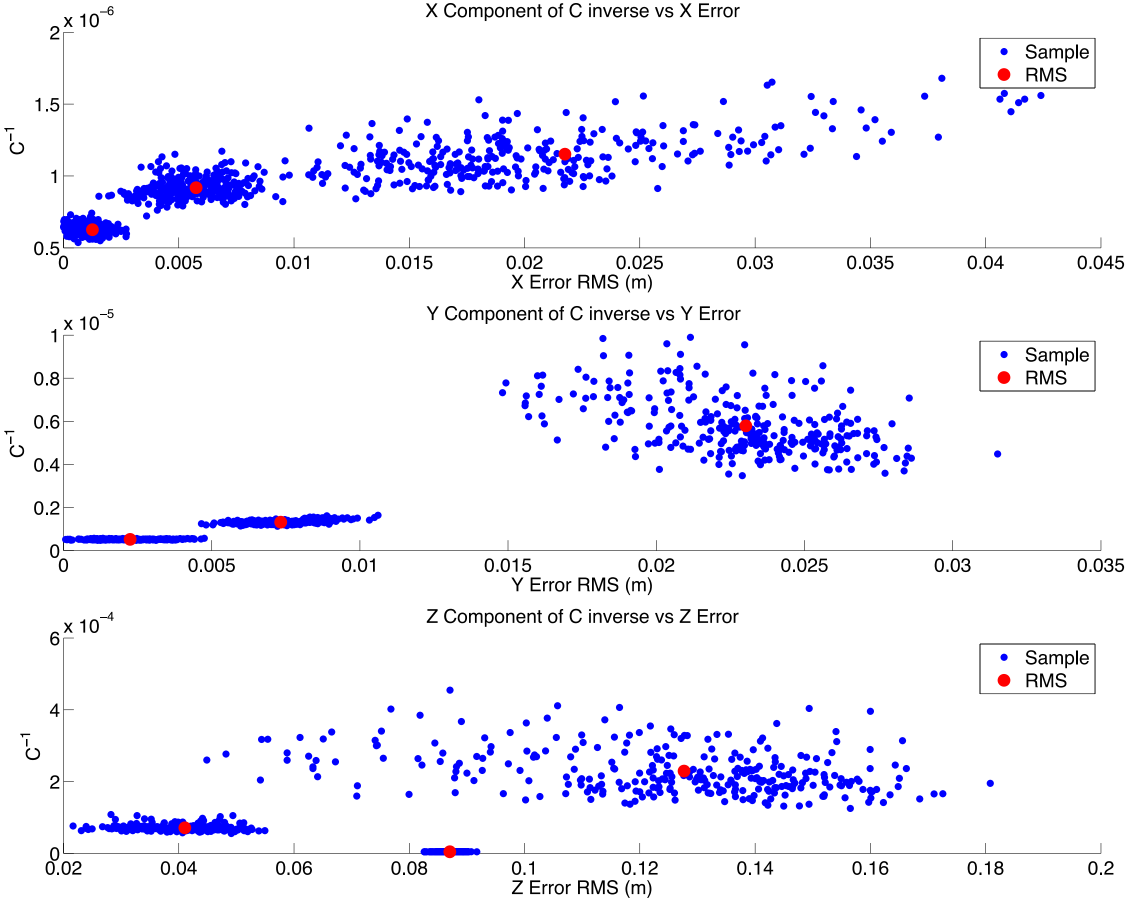






- GOES-12 FSAB Models
  - From left-to-right models should have less observability (less features)
- Trajectory involves range span from 4m to 2m

6DOF Problem: Trends



**Note: Model #1 is on left of each figure (#3 on right)*

- Each DOF portrays the **same linear relationship** between the observability and the estimation errors
- Verifies that the parameters from the simplified model are accurately comparable to the 6DOF model

Conclusions and Next Steps

- Results look very promising to continue with plans to derive a quality metric and tool for generating better models
- More work to be done in characterizing the linear relationship
  - Add more models to see if trend continues
  - Ballpark of “good vs. bad” numbers
  - How this ties into the RNF
- Will look at a test case and predict performance prior to looking at imagery, then compare to what metric predicted
  - Further fine-tune the predictions
- Need to generate a feature-by-feature representation of the plots to pick out which to keep/discard
- Perform Monte Carlo analysis using synthetic imagery to optimize the trajectory and/or model for a given scenario
  - Presumably by minimizing C^{-1} over the entire trajectory

Thank you for your time. Questions?

References

- Drummond, T.; Cipolla, R.; , "Real-time visual tracking of complex structures," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.24, no.7, pp.932-946, Jul 2002

BACK-UP SLIDES

Simplified Problem

State: $\begin{bmatrix} \vec{p} \\ \theta \end{bmatrix}$ **Measurements:** $\begin{bmatrix} \Delta\phi_1 \\ \Delta\phi_2 \\ \cdots \\ \Delta\phi_n \end{bmatrix} = \begin{bmatrix} \phi_{1,m} - \phi_{1,a} \\ \phi_{2,m} - \phi_{2,a} \\ \cdots \\ \phi_{n,m} - \phi_{n,a} \end{bmatrix}$ **Model:** $\begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \cdots \\ \vec{e}_n \end{bmatrix}$