Lidar Assisted Rendezvous and Docking

Ronald Fenton, Research Assistant
Electrical and Computer Engineering, Utah State University, Logan UT 84322

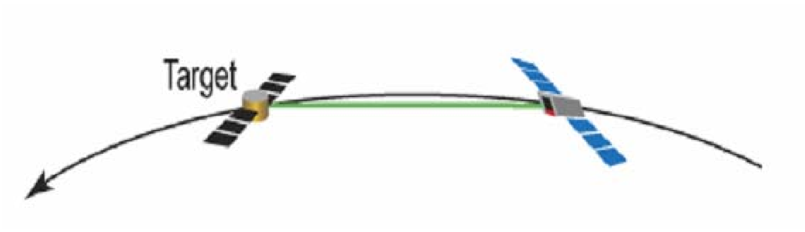
Rees Fullmer, Associate Professor
Mechanical and Aerospace Engineering, Utah State University, Logan UT 84322

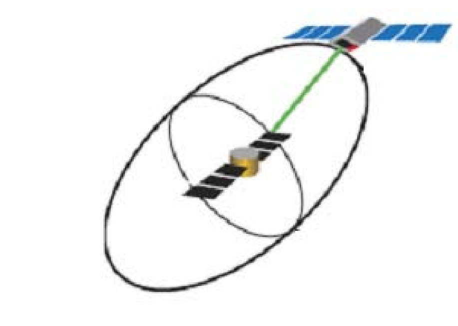
Robert Pack, Research Associate Professor
Center for Advanced Imaging Lidar, Utah State University, Logan UT 84322

Sponsored by: Space Dynamics Laboratory – Internal Research and Development

Spacecraft Relative Navigation Sensors for Rendezvous and Docking Problem

- Rendezvous and Docking:
  - Consists of a series of orbital maneuvers and controlled trajectories, which successively bring the chaser vehicle into vicinity of, and eventually dock with the target S\C.
  - Final approach and docking requires accurate relative position and attitude information between the spacecraft.
- Project Objective:
  - Investigate using lidar point cloud images to determine the real-time relative position and attitude measurements between an active spacecraft and a damaged or non-cooperative spacecraft.





CAIL Center for Advanced Imaging LADAR

UtahState UNIVERSITY

Lidar System Registration

- Lidar systems combine the capabilities of radar and optical systems to allow simultaneous measurement of range, reflectivity, velocity, temperature, azimuth, and elevation angle.

- Lidar operate at wavelengths in the ultraviolet, visible, near-,mid-, and far-infrared regions.

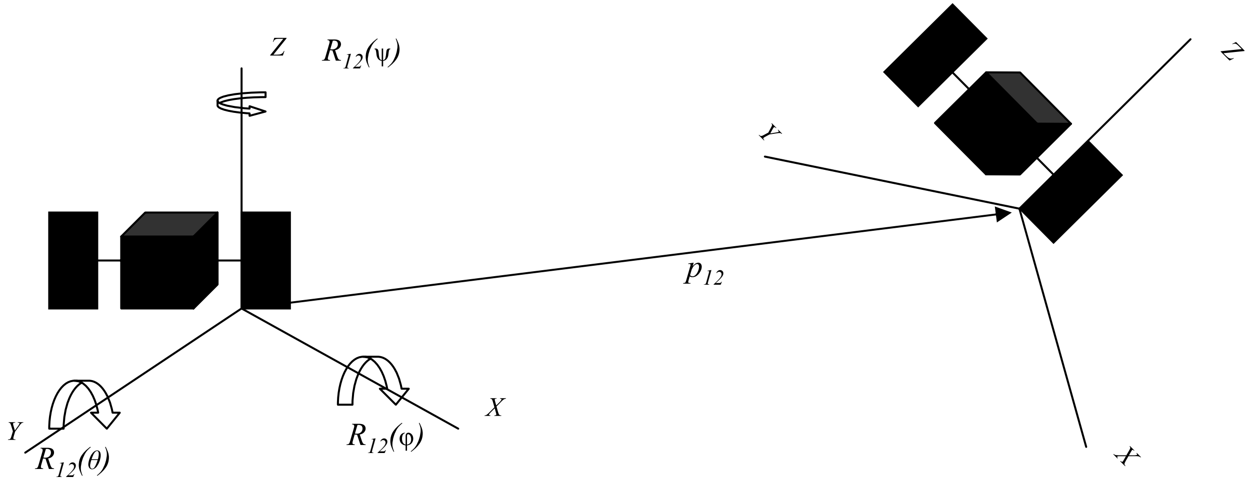
- A 2 degree-of-freedom scanning lidar can provide range data over a set of azimuth and elevation angles, creating a 3 dimensional range image or point cloud of the target.

- The 3 dimensional point cloud image can be compared to known solid models of the target to determine range and relative orientation. Determining the relative orientation and displacement vector is referred to as “Registration”

- Simulation testing requires a method for generating realistic point cloud data.

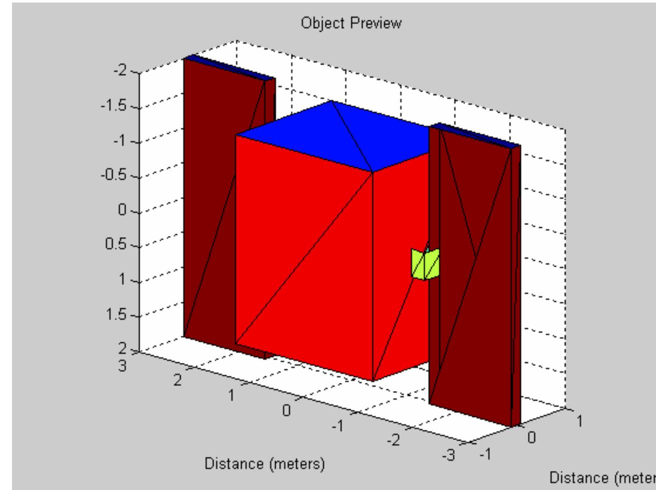
The Registration Parameters

- The relative position and orientation are determined by registration.
- Parameterized by:
  - the rotation matrix R_{12} or equivalently, the quaternion or Euler angles
  - the Cartesian translation $p_{12}(x,y,z)$

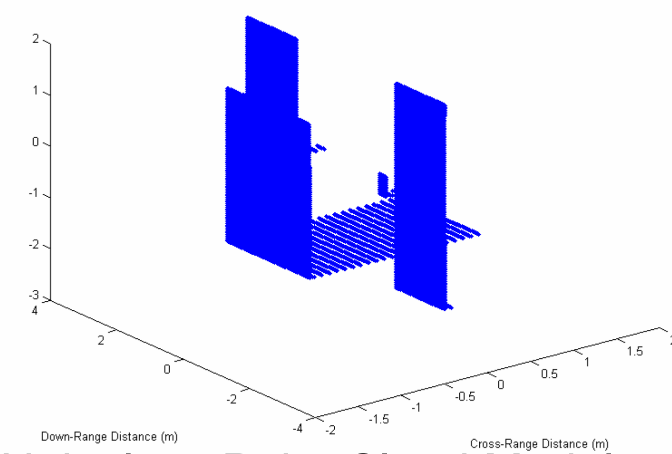


USU LADARSIM Ladar Simulator

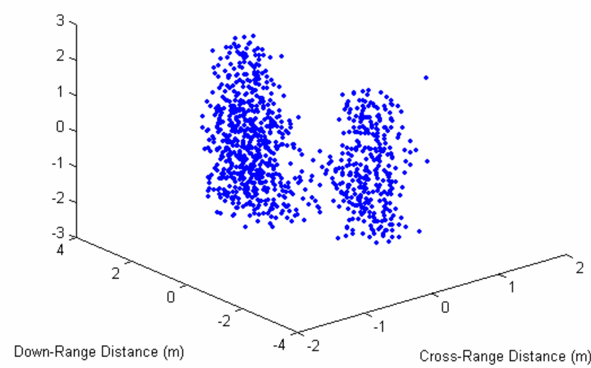
- USU LADARSIM was developed for NAVAIR and incorporates performance-related parameters associated with the laser, detector, signal processing, scanner dynamics, platform dynamics, navigation errors, and scene properties to provide general system analysis, error source modeling, and 3-D points clouds .



Original Solid Model



Noiseless Point Cloud Model

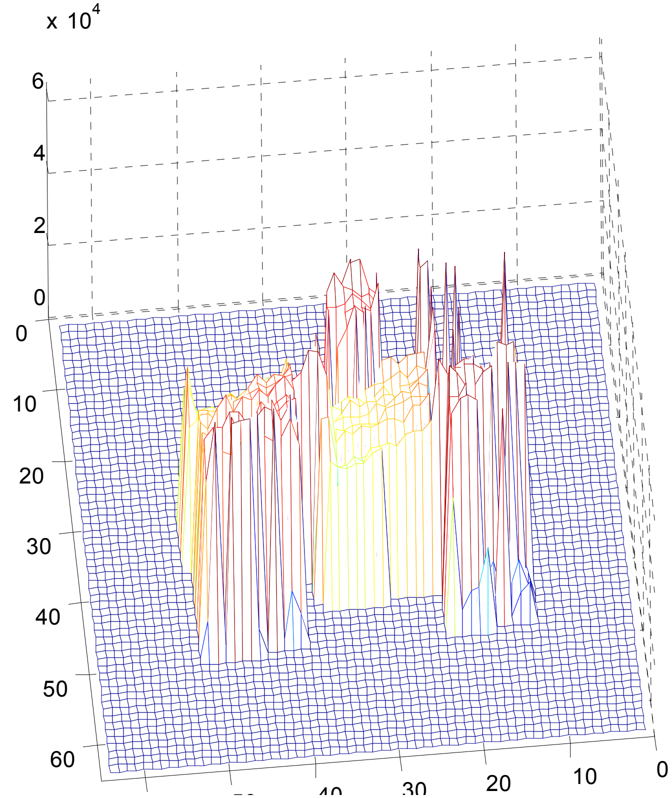


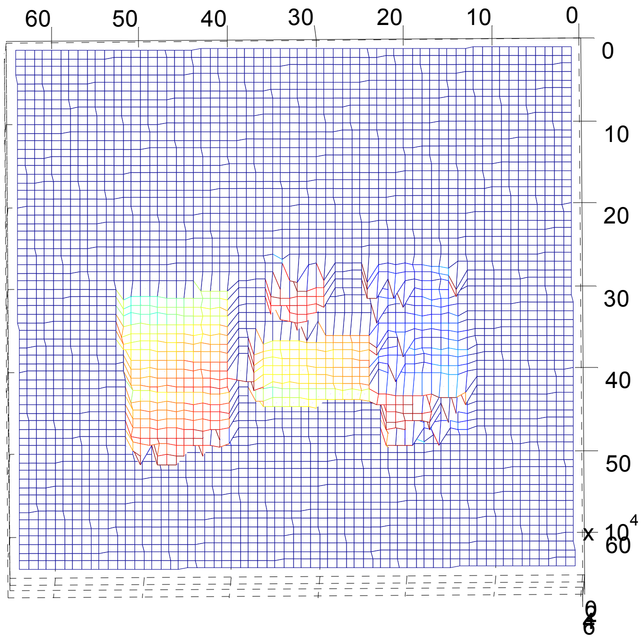
Noisy Point Cloud Model

Range Error
14 cm

Canesta Flash Lidar

- Experimental point cloud models will be obtained from a Canesta Model# DP205 55 degree field-of-view flash lidar imager for real world solid model simulations.

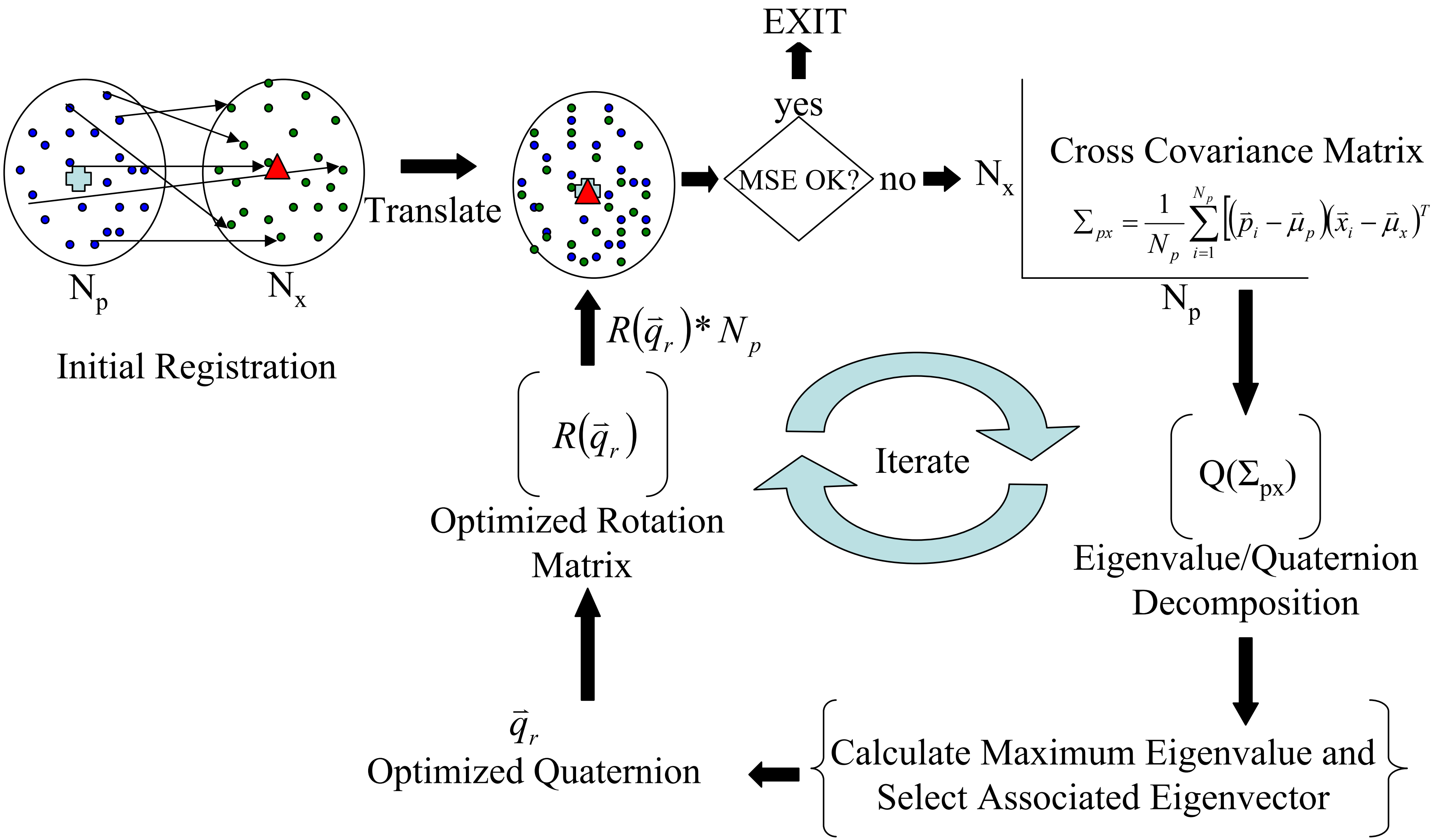




Iterative Closest Point (ICP) Algorithm

- Registration is implemented using the Iterative Closest Point Algorithm
- Originally developed by Chen and Medoni and Besl and Mckay, the ICP algorithm goes about taking two sequentially acquired range images and calculating the registration parameters between a pair of range images.
- The ICP algorithm can divided up into several different steps:
  - 1. Initialization
  - 2. Selection of Points
  - 3. Calculating point-to point or point-to-plane correspondences
  - 4. Calculate registration parameters $R_{12}(\psi,\theta,\varphi)$ and $p_{12}(x,y,z)$
  - 5. Minimizing an error metric

ICP Algorithm Overview



ICP Initialization Problem

- One major problem is calculating initial registration parameters that will lead to a final registration that is a <u>global minimum.</u>

- Initial alignment methods:
  - MACH filter -tracking scanner position and indexing surface features,
  - “spin-image” surface signatures,
  - computing principal axes of inertia of lidar scanned range images,
  - searching for corresponding points between images,
  - calculating image corners,
  - Testing all initial alignment possibilities, allows for the algorithm to successively search for the global minimum.

- Using the last method, an initial set of 312 unique initial alignments were constructed using the following values for the quaternion (q_0, q_1, q_2, q_3).

q_0 = {1, 0.5, 0},
q_1 = {1, 0.5, 0, -0.5, -1},
q_2 = {1, 0.5, 0, -0.5 -1},
q_3 = {1, 0.5, 0, -0.5 -1}.

Selection of Points

- The algorithm must next select the number of points to be used to begin point-to-point matching.

- The number of computations is proportional to the product of the number of points in each point cloud. Larger data sets greatly increase the registration computation time.

- The point selection strategy for 3-D range images can be divided into three possible strategies:
  - (1) use all the points provided in each range image;
  - (2) uniformly subsample a given percentage of the range image data;
  - (3) randomly sample a given percentage of the range image data.

Calculating Point-to-Point Correspondences

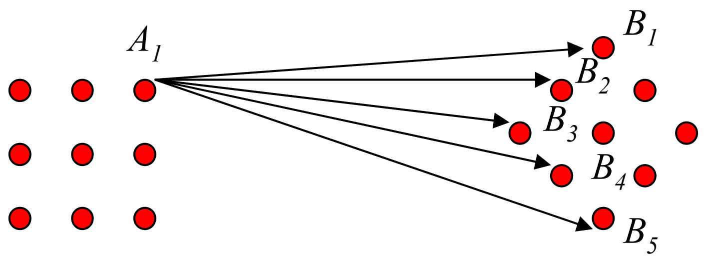
- After choosing an appropriate number of points, the ICP algorithm must match the points in the two sets of data $A = (A_1, A_2, \ldots A_n)$ and $B = (B_1, B_2, \ldots B_m)$.

- In order to match the two sets of points, a point-to-point distance criterion is used;

- The point-to-point distance calculation has been incorporated as follows:
  - Given $\vec{A}_i = (x_1, y_1, z_1)$ and $\vec{B}_i = (x_2, y_2, z_2)$

$$d(\vec{A}_i, \vec{B}_i) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

The calculation is done multiple times



Calculating Point-to-Plane Correspondences

- After choosing an appropriate number of points, the ICP algorithm must match the points in one data set $A = (A_1, A_2, \ldots A_n)$ to the planes of a 3D solid mesh.
- The point-to-plane distance calculation has been incorporated as follows:
  - Given $\vec{A}_i = (x_1, y_1, z_1)$ and a plane describe by three points one of which is $\vec{B}_i = (x_2, y_2, z_2)$

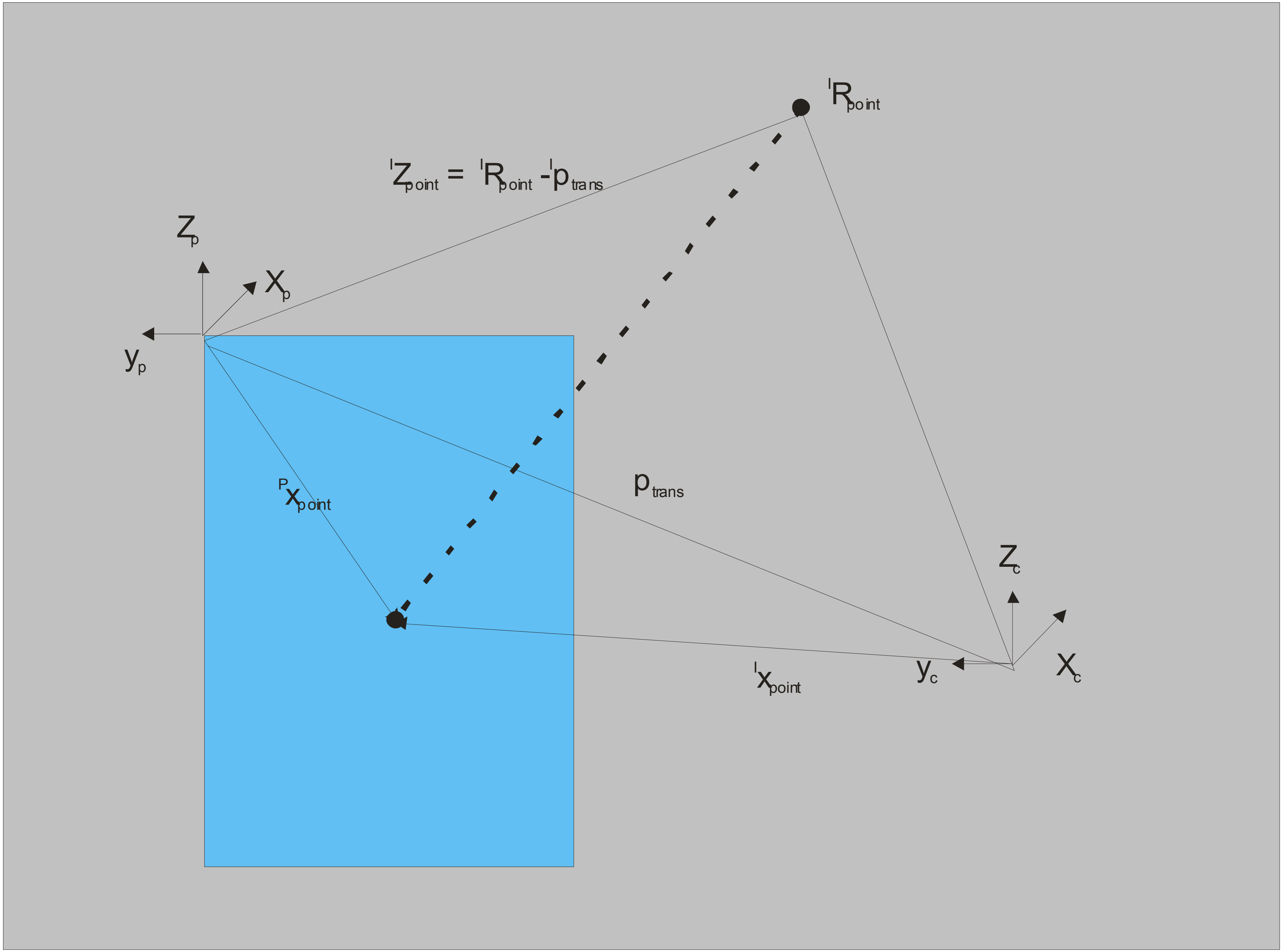
$$d(A_i, B_i) = \vec{n} \cdot \left(A_i - B_i\right)$$

- Once the point has been matched to the corresponding plane, the point needs to be projected onto that plane and realized in inertial coordinates.

$$\vec{z}_p = \frac{\left(\vec{A}_i - \vec{B}_i\right)}{\left|\vec{A}_i - \vec{B}_i\right|};\vec{x}_p = \frac{\left(\vec{A}_i - \vec{B}_i\right)\times\left(\vec{C}_i - \vec{B}_i\right)}{\left|\left(\vec{A}_i - \vec{B}_i\right)\times\left(\vec{C}_i - \vec{B}_i\right)\right|};\vec{y}_p = \vec{z}_p \times \vec{x}_p$$

$$T_{I\to P} = \begin{bmatrix} -x_p^T- \\ -y_p^T- \\ -z_p^T- \end{bmatrix} \longrightarrow {}^{P}Z_{POINT} = T_{I\to P}\,{}^{I}Z_{POINT} \longrightarrow {}^{P}X_{POINT} = \begin{bmatrix} 0 \\ e_y\,{}^{P}Z_{POINT} \\ e_z\,{}^{P}Z_{POINT} \end{bmatrix}$$

$${}^{I}X_{POINT} = {}^{I}\rho_{TRANS} + T_{P\to I}\,{}^{P}X_{POINT}$$



$^{I}R_{point}$
$^{I}Z_{point} = {}^{I}R_{point} - {}^{I}p_{trans}$
Z_p
X_p
y_p
$^{P}x_{point}$
p_{trans}
Z_c
y_c
X_c
$^{I}x_{point}$

Calculating Registration Parameters Eigenvalue/Quaternion Algorithm

The most commonly used methods in the ICP algorithm are either the quaternion-based or SVD-based methods.

The quaternion-based method is used in this study.

It calculates the quaternion equal to the eigenvector associated with the maximum eigenvalue of a 4x4 matrix constructed from the cross-covariance matrix between the point-to-point matched data sets.

Rigid body constraints are taken it account such that

$$|q| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1$$

Minimizing the Error Metric

- The final step in the ICP algorithm was to compute an error metric that the algorithm uses as a measure to the success of the calculated registration for each iteration.

- For most applications, a mean square error (MSE) or sum of the squared distance between corresponding points after the calculated registration was applied for point-to-point methods.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{y}_i - R_{12}\vec{x}_i - \vec{p}_{12}\right\|^2$$

- Similarly, a MSE between corresponding points and matched planes can be used.

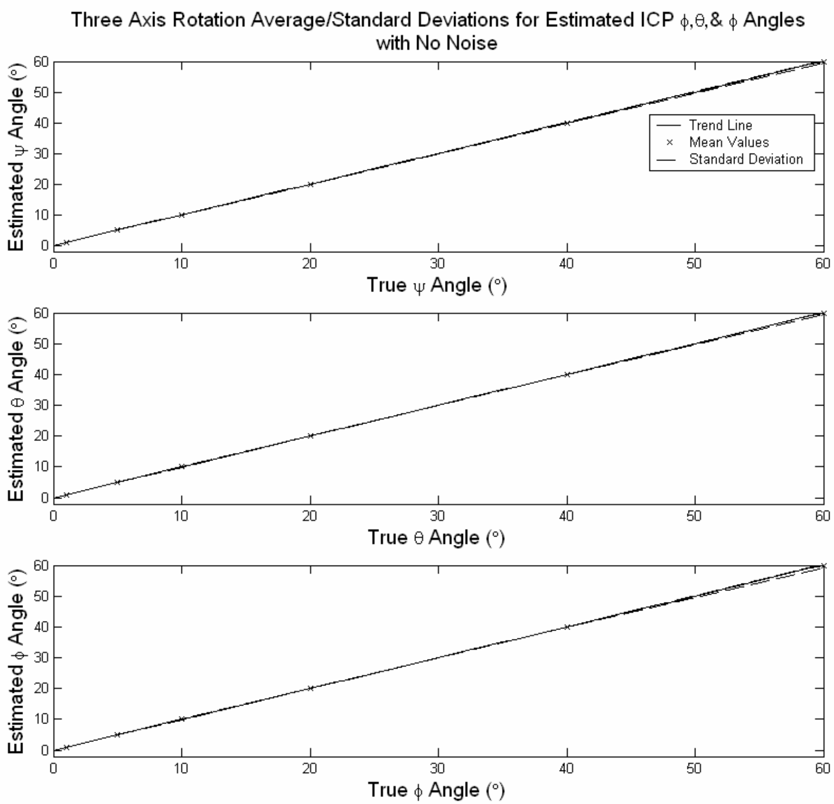
$$MSE = \sum_{i=1}^{N}\vec{n}\cdot\left(A_i - B_i\right)$$

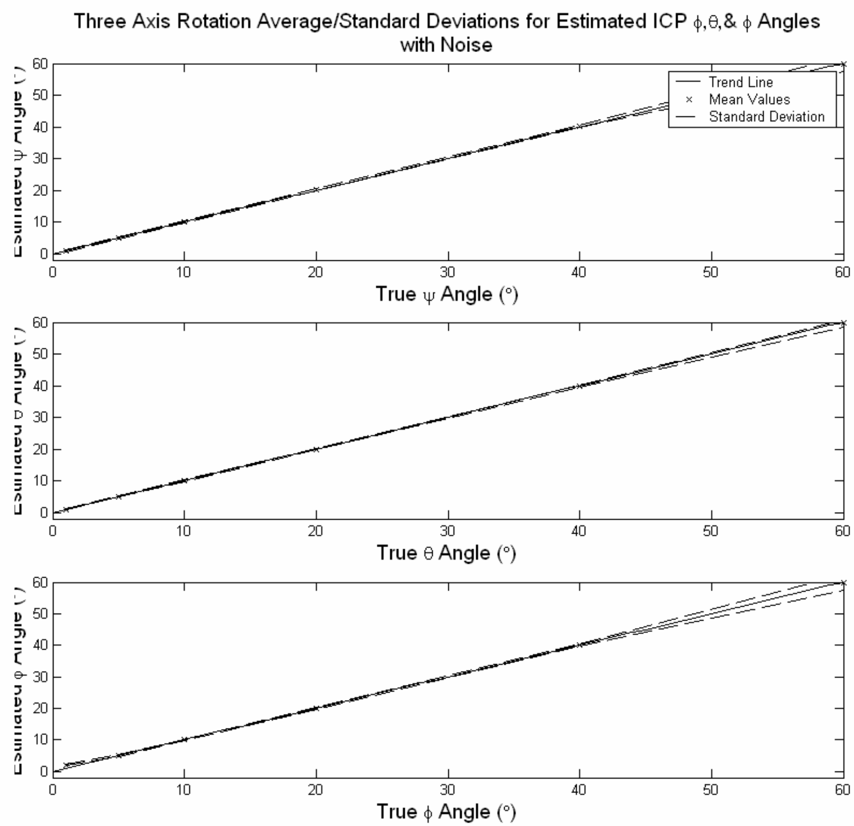
Three-Axis Rotation Simulation Results
Point-to-Point vs Point-to-Point

Three Axis Rotation	No Noise (º)	Noisy (º)
Maximum Mean Angular Error	1.79	2.54
Minimum Mean Angular Error	1.46	1.85
Average Mean Angular Error	1.69	2.21
1σ - Standard Deviatiation	1.32	1.82

Three Axis Rotation	No Noise (º)	Noisy (º)
Maximum Mean Angular Error	0.0546	0.8
Minimum Mean Angular Error	0.0522	0.349
Average Mean Angular Error	0.0532	0.462
1σ - Standard Deviatiation	0.03	0.325

Three Axis Rotation:
Error dependence on rotation angle
Point-to-Plane

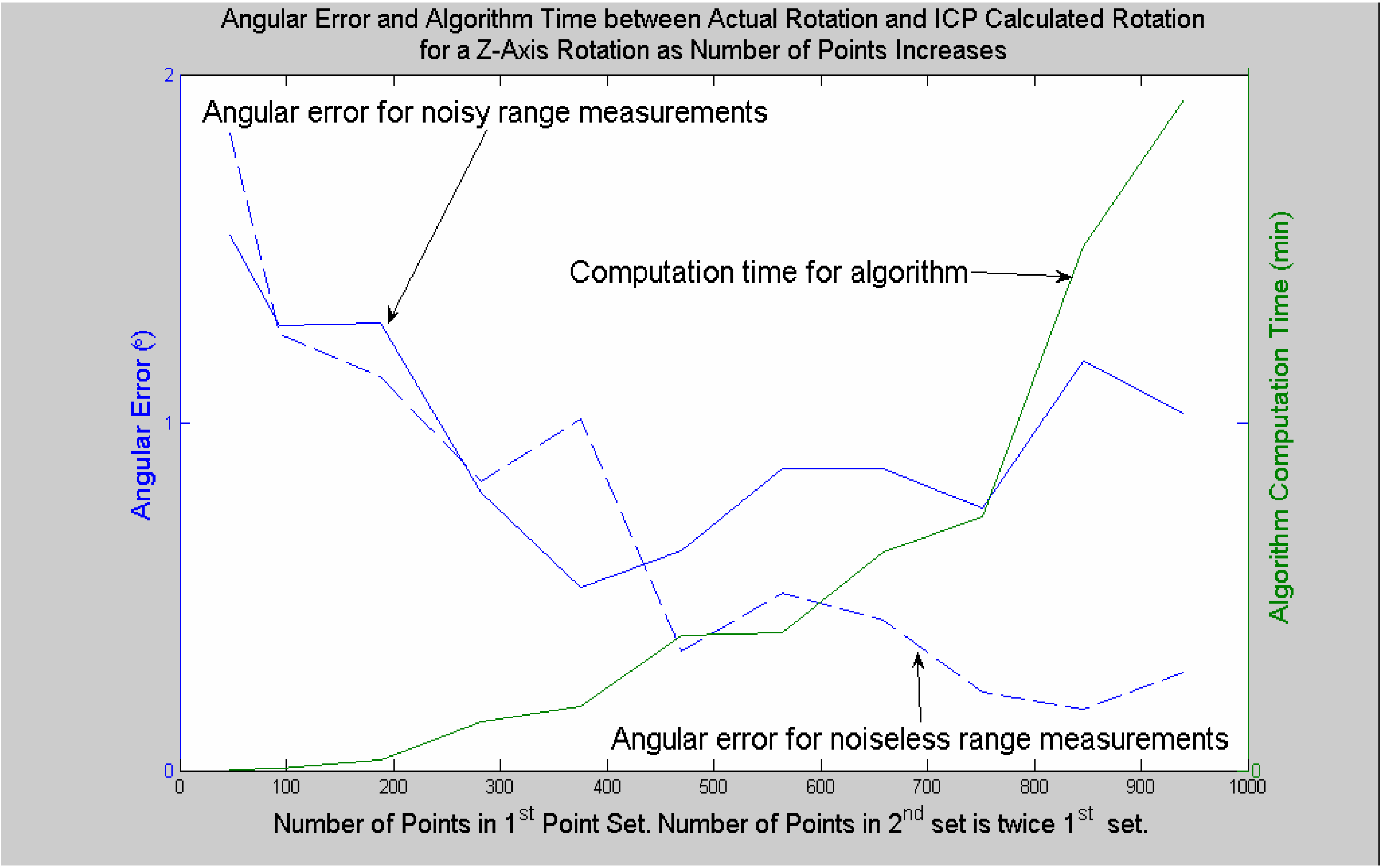




CAIL Center for Advanced Imaging LADAR

UtahState UNIVERSITY

Number of Points vs. Error and Computation Time



Conclusions

- Completed testing of accelerated and robust version of the point-to-point and point-to-plane ICP algorithm.
- Implemented ICP/Point-Plane Algorithm
  - Significant accuracy improvement when compared to ICP/Point-Point algorithm
    - 14 cm error: 0.325º 1-σ error
    - 0 cm error: 0.037º 1-σ error
  - Currently analyzing more realistic 2 cm error conditions.
  - Incorporating translational correction algorithm
  - Initialization is way too slow for real-time work

Future Works and Tasks Underway

- Investigating Real-Time initialization using Mach Filters, Spin Filters, or other possible options.
- Preparing for experimental data recording and analysis from Canesta system
  - Major data handling problem discovered and corrected
- Modification of LadarSim to incorporate moving spacecraft