Pose Performance of LIDAR-Based Navigation for Satellite Servicing

Jordan L. Sell,[*] Andrew Rhodes,[†] John O. Woods,[‡]
John A. Christian,[§] and Thomas Evans[¶]
West Virginia University, Morgantown, West Virginia, 26506, United States

Flash LIDARs are an important new sensing technology for relative navigation; these sensors have shown promising results during rendezvous and docking applications involving a cooperative vehicle. An area of recent interest is the application of this technology for pose estimation with non-cooperative client vehicles, in support of on-orbit satellite servicing activities. The capability for autonomous rendezvous with non-cooperative satellites will enable refueling and servicing of satellites (particularly those designed without servicing in mind), allowing these vehicles to continue operating rather than being retired. The research investigates numerous issues surrounding pose performance using LIDARs. Models for pose covariance are developed and compared against both simulation results and hardware test results. For the hardware tests, a SwissRanger ToF camera was used as a surrogate for Flash LIDARs.

I. Introduction

Increased interest in on-orbit satellite repair and refueling has led to a need for novel technologies for relative navigation with respect to a non-cooperative client vehicle. Building on past experience in spacecraft rendezvous and docking, NASA hopes to develop capabilities for on-orbit repair, maintenance, and refueling of legacy satellite systems — many of which were designed without thought for serviceability. These vehicles could routinely be repaired or refueled given modest improvements in autonomous navigation and robotic systems.[1] This manuscript addresses advances in sensing and data processing technologies that may help bring such capabilities closer to reality.

LIght Detection And Ranging (LIDAR) sensors are a popular choice as the primary relative navigation sensor for rendezvous and docking because they provide both bearing and range measurements with respect to the observed object. These data may be processed through an on-board filter to estimate the spacecraft's relative position and attitude. LIDARs use light round-trip time-of-flight (ToF) to estimate range according to the simple relationship

$$r = \frac{c\,t}{2} \tag{1}$$

where r is the range from the sensor to the object, c is the speed of light, and t is the light's traveled time-of-flight. Thus, for each measured point, the outputs from the LIDAR device are the projected (x, y) coordinates in the plane perpendicular to the boresight, the range r, and an intensity; the (x, y, z) for each point is transformed into (x, y, z) coordinates in the sensor frame, where z is aligned with the LIDAR boresight.

One LIDAR variant that has become popular in the last five years is the Flash LIDAR. Flash LIDARs simultaneously provide the range and bearing from the sensor to many points on the observed object,

[*] Graduate Research Assistant, Mechanical and Aerospace Engineering Department, and AIAA Student Member.
[†] Graduate Research Assistant, Mechanical and Aerospace Engineering Department.
[‡] Post-Doctoral Fellow, Mechanical and Aerospace Engineering Department.
[§] Assistant Professor, Mechanical and Aerospace Engineering Department, and AIAA Senior Member.
[¶] Research Assistant Professor, Mechanical and Aerospace Engineering Department.

generating a three-dimensional (3D) point cloud representation of the observed scene. Recent test flights and theoretical studies have demonstrated Flash LIDARs as a viable sensor option.[1–4] To date, however, most in-flight use and testing of Flash LIDARs has focused on relative navigation with respect to a cooperative target. Previous Flash LIDAR flight tests[3,5] and subsequent operational use by the SpaceX Dragon capsule[6] depended upon the existence of retro-reflectors on the observed object — usually the International Space Station (ISS) — to assist with the navigation process. These retro-reflectors work as control points for the estimation of relative position and attitude (collectively called *pose*), meaning that just these points — which are positioned at well-known locations on the observed object — are used in the estimation process.[4] Vehicles with such navigation aids are said to be *cooperative* objects.

More recent work has considered pose estimation with Flash LIDARs of objects without navigation aids (known as *non-cooperative* objects).[7,8] Examples of non-cooperative objects might include asteroids, as well as client spacecraft requiring on-orbit refueling. Indeed, solutions to the more general non-cooperative problem largely subsume those of the cooperative problem; however, much work remains in maturing and understanding the performance of these LIDAR-based techniques for non-cooperative relative navigation.

One of the outputs of a Flash LIDAR is a 3D point cloud. Much work has been done on processing point cloud data, and a great deal of these developments have been captured in the open-source Point Cloud Library.[9] Of particular importance is the problem of finding the transformation that maps one point cloud (the *measurements*) on top of another point cloud (a pre-existing *model* of the object). The resulting transformation — which consists of a translation and a rotation — is the pose estimate sought in the relative navigation problem. It should be noted that, depending on the application and the authors, this task is sometimes also referred to as *point cloud registration.*

The present paper reviews the Flash LIDAR measurement model and discusses the selected method for pose estimation, and briefly touches on a promising strategy for providing an initial estimate of pose. An overview of the pose estimation process is provided in Figure 1. The pose estimation problem includes three steps: (1) pose initialization, (2) matching of points and sampled surfaces between the measured LIDAR data and a corresponding digital model stored onboard the spacecraft computer, and (3) estimation of pose given a set of corresponding points. The approximate performance for such pose algorithms is also presented. Finally, this procedure is demonstrated via Monte Carlo simulation, as well as through laboratory testing completed at the West Virginia Robotics Technology Center (WVRTC).

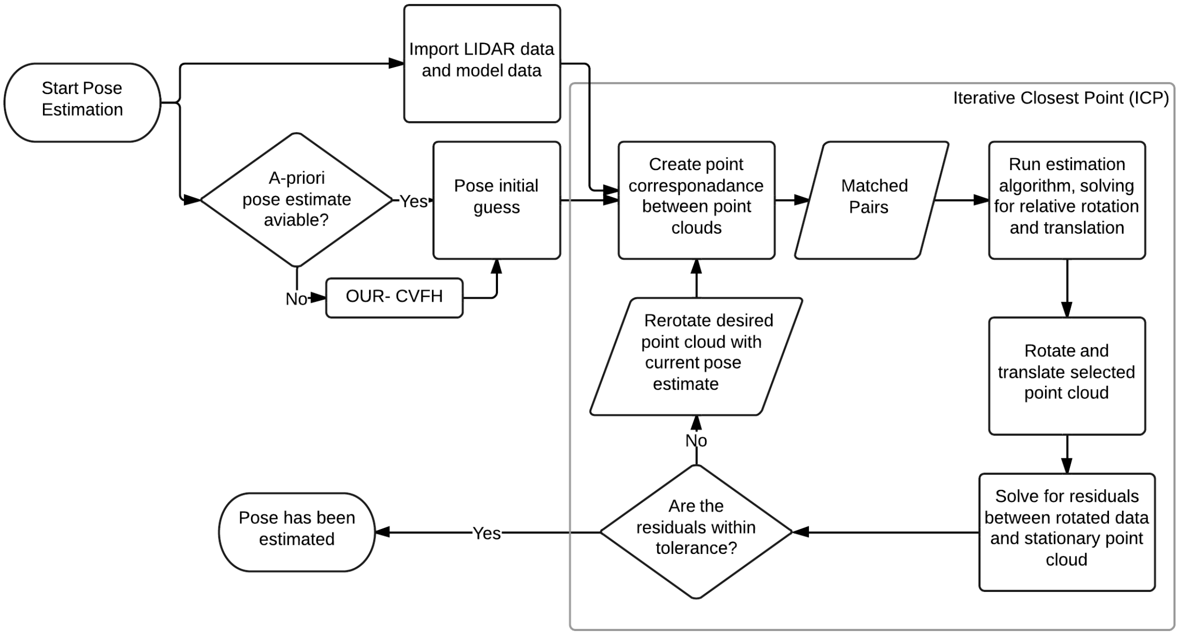

Start Pose Estimation
Import LIDAR data and model data
Iterative Closest Point (ICP)
A-priori pose estimate aviable?
Yes
Pose initial guess
No
OUR- CVFH
Create point corresponadance between point clouds
Matched Pairs
Run estimation algorithm, solving for relative rotation and translation
Rerotate desired point cloud with current pose estimate
Rotate and translate selected point cloud
No
Pose has been estimated
Yes
Are the residuals within tolerance?
Solve for residuals between rotated data and stationary point cloud

Figure 1: Flowchart of pose estimation process.

II. LIDAR Measurement Model

The basic non-linear measurement model of a LIDAR used in relative pose estimation is given by

$$\mathbf{y}_i = \mathbf{h}_i(\mathbf{T}, \mathbf{r}) = \mathbf{T}\mathbf{p}_i + \mathbf{r} \tag{2}$$

where $\mathbf{y}_i$ is the LIDAR measurement to the ith point in the sensor frame, $\mathbf{T}$ is the rotation matrix from the measured object's body frame to the sensor body frame, $\mathbf{p}_i$ is the location (relative to the object's origin) of the ith measured point, and $\mathbf{r}$ is the position of the sensor expressed in the target body frame.

In real life applications, the sensor measurements will be corrupted by noise,

$$\tilde{\mathbf{y}}_i = \mathbf{T}\mathbf{p}_i + \mathbf{r} + \boldsymbol{\nu}_i \,, \tag{3}$$

where $\boldsymbol{\nu}_i$ is the zero-mean Gaussian noise, $\boldsymbol{\nu}_i \sim \mathcal{N}\left(0, \mathbf{R}_i\right)$. If one assumes that the range error and bearing error from a LIDAR are small and uncorrelated, then it can be shown that the covariance for a single LIDAR measurement is described by

$$\mathbf{R}_i = \rho_i^2 \sigma_\phi^2 \mathbf{I}_{3\times3} + \left(\frac{\sigma_\rho^2}{\rho_i^2} - \sigma_\phi^2\right) \mathbf{y}_i \mathbf{y}_i^\mathsf{T} \,, \tag{4}$$

where σ_ϕ is the standard deviation of the the bearing error (in radians), σ_ρ is the standard deviation of the range error, and $\rho_i = \|\mathbf{y}_i\| = \sqrt{\mathbf{y}_i^\mathsf{T}\mathbf{y}_i}$ is the range from the sensor to the model's ith point.

III. Pose Estimation

Pose estimation using corresponding 3D point cloud data is a well-studied problem, for which a variety of techniques has been considered. A thorough survey was conducted by Eggert *et al.*[10] which compares the four most well-known and widely-used pose estimation strategies. Those methods analyzed and compared within the survey include singular value decomposition (SVD),[11,12] unit quaternions,[13] orthonormal matrices,[14] and dual quaternions.[15] All of these methods begin with least squares fitting, but differ in their approaches to solving for the six degrees of freedom (6 DOF, three axial rotation angles and 3D spatial translation vector). Correct one-to-one point correspondences are assumed by all methods prior to optimization of the pose estimate. While all of the four methods provide workable solutions to the problem at hand, the SVD method has been selected for its quick execution, robust handling of statistical outliers, and accuracy for both ideal and noisy measurements. This method has been a popular substitute for the unit quaternion method and offers comparable accuracy.[16]

To begin the pose estimation process, one must first examine the probability density function (PDF) describing the chances of obtaining any particular measurement $\tilde{\mathbf{y}}_i$, given the state $(\mathbf{T}, \mathbf{r})$, which takes the form of a multivariate Gaussian,

$$\mathrm{p}\left(\tilde{\mathbf{y}}_i \,|\, \mathbf{T}, \mathbf{r}\right) = C_i \exp\left[-\frac{1}{2}\left(\tilde{\mathbf{y}}_i - \mathrm{E}\left[\tilde{\mathbf{y}}_i\right]\right)^\mathsf{T} \mathbf{R}_i^{-1} \left(\tilde{\mathbf{y}}_i - \mathrm{E}\left[\tilde{\mathbf{y}}_i\right]\right)\right] \tag{5}$$

with constant

$$C_i = (2\pi)^{-m/2} \det\left(\mathbf{R}_i\right)^{-1/2} \,. \tag{6}$$

If the goal is maximum likelihood estimation of the pose, then the $\mathbf{T}$ and $\mathbf{r}$ that maximize the likelihood of obtaining all of the simultaneous Flash LIDAR measurements is desired,

$$\max J(\mathbf{T}, \mathbf{r}) = \mathrm{p}\left(\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \ldots, \tilde{\mathbf{y}}_n \,|\, \mathbf{T}, \mathbf{r}\right) \,. \tag{7}$$

If the error on each of the measurements is assumed to be uncorrelated, then this maximization problem may be rewritten as the familiar minimization problem of the form[17]

$$\min J(\mathbf{T}, \mathbf{r}) = \sum_{i=1}^{n} \frac{1}{2}\left(\tilde{\mathbf{y}}_i - \mathrm{E}\left[\tilde{\mathbf{y}}_i\right]\right)^\mathsf{T} \mathbf{R}_i^{-1} \left(\tilde{\mathbf{y}}_i - \mathrm{E}\left[\tilde{\mathbf{y}}_i\right]\right) \,. \tag{8}$$

For the sake of deriving the pose estimation algorithm, suppose for a moment that the measurements have a spherical covariance,

$$\mathbf{R}_i^{-1} = w_i \mathbf{I}_{3\times3} \,, \tag{9}$$

where $w_i = \frac{1}{\sigma_i^2}$ (and σ_i is the sensor's uncertainty at any point i in the point cloud). Additionally, the zero-mean additive noise means the measurements are unbiased,

$$\mathrm{E}\left[\tilde{\mathbf{y}}_i\right] = \mathbf{T}\mathbf{p}_i + \mathbf{r} + \mathrm{E}[\boldsymbol{\nu}_i] = \mathbf{y}_i \,. \tag{10}$$

Thus, the minimization problem of interest simplifies to

$$\min J(\mathbf{T}, \mathbf{r}) = \frac{1}{2}\sum_{i=1}^{n} w_i \left(\tilde{\mathbf{y}}_i - \mathbf{T}\mathbf{p}_i - \mathbf{r}\right)^{\mathsf{T}} \left(\tilde{\mathbf{y}}_i - \mathbf{T}\mathbf{p}_i - \mathbf{r}\right) \,. \tag{11}$$

The optimal pose may be found by applying the first differential condition.[18] Begin by taking the partial with respect to $\mathbf{r}$ and setting the result to zero,

$$\frac{\partial J}{\partial \mathbf{r}} = \sum_{i=1}^{n} w_i \,\left(\tilde{\mathbf{y}}_i - \mathbf{T}\mathbf{p}_i - \mathbf{r}\right) = 0 \,, \tag{12}$$

and rearranging gives the optimal translation vector,

$$\hat{\mathbf{r}} = \frac{\sum_{i=1}^{n} w_i \,\tilde{\mathbf{y}}_i}{\sum_{i=1}^{n} w_i} - \mathbf{T}\,\frac{\sum_{i=1}^{n} w_i \,\mathbf{p}_i}{\sum_{i=1}^{n} w_i} \,. \tag{13}$$

The expression can be simplified by recognizing that the two fractions represent the weighted centers of mass of the measured and model points,

$$\begin{aligned} \bar{\mathbf{y}} &= \frac{\sum_{i=1}^{n} w_i \,\tilde{\mathbf{y}}_i}{\sum_{i=1}^{n} w_i} \text{ and} \\ \bar{\mathbf{p}} &= \frac{\sum_{i=1}^{n} w_i \,\mathbf{p}_i}{\sum_{i=1}^{n} w_i} \,, \end{aligned} \tag{14}$$

respectively; and the coordinates may be expressed relative to the centroids, $\check{\mathbf{y}} = \tilde{\mathbf{y}}_i - \bar{\mathbf{y}}$ and $\check{\mathbf{p}} = \mathbf{p}_i - \bar{\mathbf{p}}$, respectively. The translation vector between the centers of mass is expressed thusly:

$$\hat{\mathbf{r}} = \bar{\mathbf{y}} - \mathbf{T}\bar{\mathbf{p}} \,. \tag{15}$$

To solve for the optimal rotation matrix, the optimal translation $\hat{\mathbf{r}}$ is substituted into the minimization expression (Eq. 11) for $\mathbf{r}$:

$$\min J(\mathbf{T}) = \frac{1}{2}\sum_{i=1}^{n} w_i \,\left(\tilde{\mathbf{y}}_i - \mathbf{T}\mathbf{p}_i - \hat{\mathbf{r}}\right)^{\mathsf{T}} \left(\tilde{\mathbf{y}}_i - \mathbf{T}\mathbf{p}_i - \hat{\mathbf{r}}\right) \,. \tag{16}$$

Substituting the weighted centers of mass defined in Eq. 14, and then rearranging, gives

$$\min J(\mathbf{T}) = \lambda_0 - \sum_{i=1}^{n} w_i \,\check{\mathbf{y}}_i^{\mathsf{T}} \,\mathbf{T}\,\check{\mathbf{p}}_i \tag{17}$$

where $\lambda_0 = \sum_{i=1}^{n} w_i(\check{\mathbf{y}}_i^{\mathsf{T}}\check{\mathbf{y}}_i + \check{\mathbf{p}}_i^{\mathsf{T}}\check{\mathbf{p}}_i)$. Since the second term of Eq. 17 is a scalar, the summation can be rewritten as a trace, whose operands may be cyclically permuted, resulting in the expression

$$\min J(\mathbf{T}) = \lambda_0 - \mathrm{tr}\left[\mathbf{T}\mathbf{B}^{\mathsf{T}}\right] \tag{18}$$

where $\mathbf{B}$ is a 3×3 matrix,

$$\mathbf{B} = \sum_{i=1}^{n} w_i \check{\mathbf{y}}_i \check{\mathbf{p}}_i^{\mathsf{T}} \,. \tag{19}$$

As seen in Haralick *et al.*,[12] the singular value decomposition of $\mathbf{B}$ is expressed as

$$\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}} \,, \tag{20}$$

where the columns of $\mathbf{U}$ are orthonormal eigenvectors of $\mathbf{B}\mathbf{B}^\mathsf{T}$ and the columns of $\mathbf{V}$ are orthonormal eigenvectors of $\mathbf{B}^\mathsf{T}\mathbf{B}$. $\mathbf{D}$ is a diagonal matrix containing the singular values. Using this decomposition of $\mathbf{B}$, it is now possible to solve for the optimal rotation matrix,

$$\hat{\mathbf{T}} = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U})\det(\mathbf{V}) \end{bmatrix}\mathbf{V}^\mathsf{T}\,, \tag{21}$$

where the $\det(\mathbf{U})\det(\mathbf{V})$ term assures that the resulting rotation matrix is right-handed. The optimal translation may now be found by incorporating the newly found optimal rotation matrix, *e.g.*,

$$\hat{\mathbf{r}} = \bar{\mathbf{y}} - \hat{\mathbf{T}}\bar{\mathbf{p}}\,. \tag{22}$$

Once the optimal relative rotation matrix and translation are acquired, the pose estimation problem is completed.

IV. Computation of Pose via Iterative Closest Point

Pose estimation, sometimes called registration, is the process of aligning two objects on the basis of their shared features. Stated precisely, pose is the transformation which maps the location of features in a known frame (the model) to the observed location of those same features in an unknown frame (the sensor data). The iterative closest point (ICP) strategy[16] has become a standard method for registration of three-dimensional objects, whose surfaces may be sampled to produce point clouds.

Given an initial pose estimate, ICP seeks to iteratively improve the alignment between a sensor-produced point cloud and the reference model cloud. For each point in the sensor cloud, the nearest point in the model is determined, defining a correspondence. The sum of the squared correspondence distances is iteratively minimized until ICP converges upon a minimum.

A. General Pose Initialization

A key problem with ICP is its tendency, during the course of iteration, to become 'stuck' at locally minimized sum-of-squared-distances which are not the absolute (correct) minima. ICP's narrow basin of convergence is typically addressed by providing the algorithm with a reasonable initial guess of the pose, often the output of ICP from the previous time point.

The very first pose initialization is obtained by some other means, such as using the known poses for the client body and for the sensor at arrival on orbit. Such techniques require a high degree of knowledge about the client's attitude, which may not be available for a tumbling satellite (for example, a spacecraft which has lost attitude control, or an asteroid). Regardless of the status of the client body, a filter which relies exclusively on previous filter states is less robust than would be desired.

B. Oriented, Unique, and Repeatable Clustered Viewpoint Feature Histograms

A promising strategy for pose initialization involves the use of OUR-CVFH, which stands for Oriented, Unique, and Repeatable Clustered Viewpoint Feature Histograms.[19] The strategy is related to Fast Point Feature Histograms,[20] which was previously explored for pose estimation during approach to ISS.[8] The authors are exploring the use of this technique for initialization of ICP in ongoing research.

The OUR-CVFH recognition pipeline, which is part of the PCL,[9] involves the identification of clusters of points with similar surrounding surface features. The clusters are each described by at least one, but no more than eight, 308-bin histograms. Training of OUR-CVFH involves the identification of clusters on a set of training views. By matching the clusters in a test LIDAR point cloud to those in a specific training view (which can be accomplished by a k nearest neighbors search), a set of candidate matches is obtained. The best match is typically the closest training view to the test view, which provides a 6 DOF pose estimate.

Figure 2 shows the top candidate initial pose guess from the OUR-CVFH pipeline on ISS's Zarya module, using a simulated LIDAR image produced at 100 meters by the LIDARgl OpenGL simulator (developed in the WVU Applied Space Exploration Lab). This estimate was calculated in 2.1 seconds on a 2011 Macbook Air.

The OUR-CVFH method is noteworthy not only for its speedy 6 DOF pose approximation, but for its ability to recognize objects — even when those objects are partially occluded.[19] Its pose estimate is at least

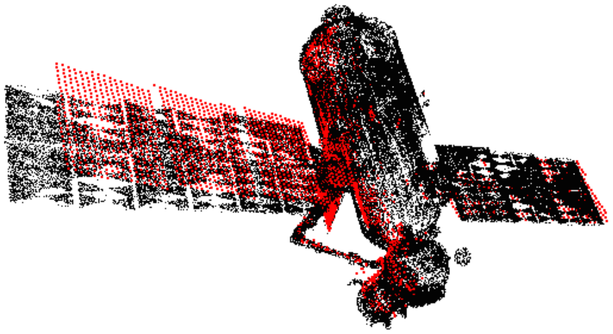


Figure 2: Pose initialization estimate for ISS module Zarya using the OUR-CVFH pipeline at a range of 100 meters. Red points represent the LIDAR-derived point cloud; black points indicate the composite training cloud.

as good as the angle between training views; with sufficient training, albeit on a scale untenable for modern flight processors, an exact pose could be calculated without any help from ICP.

Work is under way on characterizing the performance of the OUR-CVFH pipeline at various ranges. However, the underlying strategy suggests that OUR-CVFH may fail to recover client body attitude when sensor readings include few points (*e.g.*, early in the approach). Furthermore, at very close ranges, OUR-CVFH is limited by the fidelity of the 3D model used for training. At any range, degeneracy of views (such as for rotationally symmetric objects) is problematic. Filters which rely on the OUR-CVFH pipeline for pose re-initialization will need to account for these factors. Although the method presented in this manuscript does not make use of OUR-CVFH for its pose initialization, future work will demonstrate this technique.

V. Development of Analytic Pose Covariance Relations

The optimal estimate of the pose of a spacecraft is found by minimizing the weighted sum of the measurement residuals. Ideally, this weighting would be the inverse of the measurement covariance. To compute pose using SVD, the weighting matrix is assumed to be spherical (see Eq. 9), but Eq. 4 clearly shows that a LIDAR measurement doesn't generally have a spherical covariance. This assumption makes the resulting pose solution a weighted least squares estimate, but not necessarily a maximum likelihood estimate. Therefore, the analytic state covariance for the described situation is given as[17]

$$\mathbf{P} = \left[\sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{W}_i \mathbf{H}_i\right]^{-1} \sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{W}_i \mathbf{R}_i \mathbf{W}_i \mathbf{H}_i \left[\sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{W}_i \mathbf{H}_i\right]^{-1} \tag{23}$$

where, if $\mathbf{W}_i = \frac{1}{\sigma_i^2}\mathbf{I}_{3\times3}$, then

$$\mathbf{P} = \left[\sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{H}_i\right]^{-1} \sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{R}_i \mathbf{H}_i \left[\sum_{i=1}^{n} \mathbf{H}_i^{\mathsf{T}} \mathbf{H}_i\right]^{-1} . \tag{24}$$

It should be noted that this covariance estimate is still a bit optimistic. The construction of the covariance matrix $\mathbf{R}$ assumes an observed point is tied to a specific point in the model. While ICP will force this correspondence to be the case, most LIDAR measurements of non-cooperative objects will belong to a surface and, therefore, only contain information in the direction of the surface normal to first order. As a result, it is necessary to observe multiple surfaces facing sufficiently independent directions for the full 6 DOF pose to be observable from a single LIDAR image. When point-to-point matching is enforced between the model and the measurements by ICP, the measurement noise will result in incorrect point correspondence — an error source not captured in the above covariance equations. Consequently, this analytic covariance will underestimate the actual pose covariance in practice. Therefore, future work will examine point-to-plane covariance in more detail.

VI. Monte Carlo Validation of Pose Estimation

In order to examine and test the algorithms being implemented within the pose estimation program, Monte Carlo simulations were run and the results analyzed. As a way of demonstrating the above comments about point-to-point matching, simulations were run using both perfect matching (rarely achieved in practice) and imperfect matching (more similar to reality).

These simulations assume that the LIDAR is observing a simple box with dimensions of $0.305 \times 0.231 \times 0.114\,\mathrm{m}$. The model point cloud, normally created through a CAD model, is then imported and provides a grid of equally space points on the model. Each Monte Carlo run creates a new set of simulated LIDAR measurements (a 3D point cloud in this case) with noise described by Eq. 4. Additionally, a random pose initialization error is applied at the beginning of each run.

The two data sets (measured and model) are then processed through a custom point-to-point ICP implementation making use of the MATLAB `KDTreeSearcher` class. The matching function works by first creating a kD tree with the model data. The matching function then runs a standard nearest neighbor search to match each points from the measured point cloud to a candidate point in the model data set. The matched points from the model set are sent into the estimation routine, along with the measurement data, to calculate a rotation matrix and the position vector. This process repeats until the residuals of the measurements are within a set tolerance. The residuals are calculated between the estimated transform and the known actual transform. After the tolerance criteria have been met the residuals are stored and the Monte Carlo simulation begins another run. The stored residuals are used to calculate the covariance of the estimation process over ten thousand runs and compared with the analytic results.

A. Monte Carlo Analysis of Pose Performance with Perfect Matching

The first case analyzed uses model data comprised of points randomly sampled from the surface of a simulated box. The measured data are created from a subset of these points that have been translated, rotated, and then corrupted with noise. Because the measured points were pulled directly from the model point cloud, and since the measurement error is substantially smaller than the spacing between the sampled points, most of the point correspondences will be correct when the ICP algorithm is finished. The perfect matches eliminate the error in the point correspondence, representing an idealized scenario that rarely occurs in practice for non-cooperative navigation.

The results of this 10,000-run Monte Carlo analysis can be seen in Figure 3. The residuals along each axis are shown in the histograms, with each bar describing the frequency at which that error occurred during Monte Carlo analysis. The dashed line shows the numeric covariance from ICP and the solid line shows the calculated analytic covariance provided through Eq. 24.

B. Monte Carlo Analysis of Pose Performance with Imperfect Matching

After ensuring that the ICP implementation worked for measured points with perfect matches in the model cloud, a second set of simulations was run for the more realistic scenario where points do not have definite matches. As before, the measured point cloud was created from points randomly distributed on the box surface. The model point cloud, however, now consists of equally spaced points distributed over the surface of the box. The spacing between these points was chosen to be very small compared to the sensor noise. As such, the randomly generated surface points will be matched to a point on the box surface that is different from the actual surface point from which the measurement was created. The results of such a Monte Carlo analysis are shown in Figure 4. As expected, the analytic covariance is overly optimistic and has a slightly smaller standard deviation than the actual numeric results.

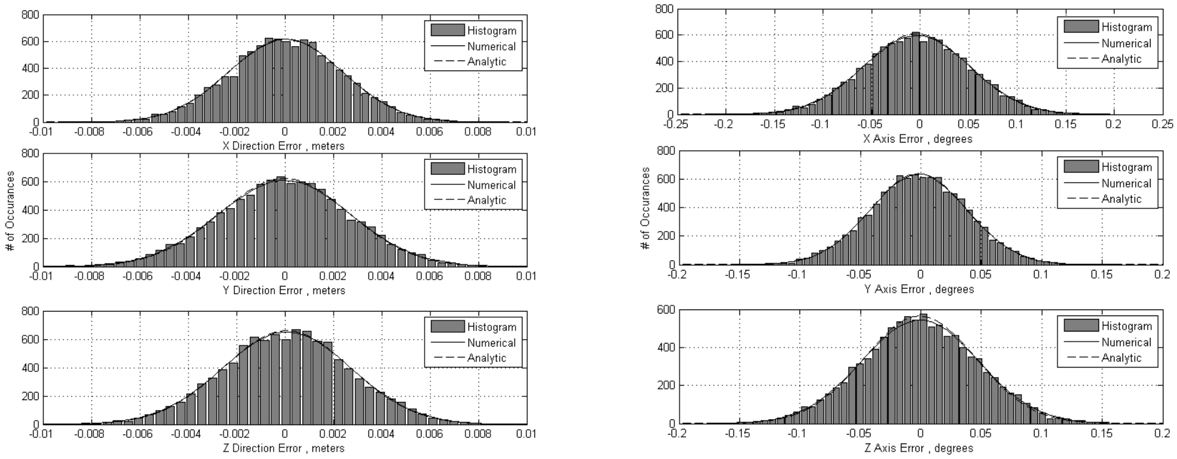
VII. Experimental Validation of Pose Estimation Using a SwissRanger

In addition to the Monte Carlo analysis described in the previous section, it is equally as important is to assess pose performance when processing actual sensor data. A SwissRanger 4000 ToF camera was selected as a surrogate Flash LIDAR for experimental validation of the analytic pose covariance. The SwissRanger and Flash LIDAR produce analogous information (intensity image and range image) when viewing an object, although using different methods. Data from a SwissRanger was collected in a controlled laboratory environment. The following summarizes the test facility, experimental set-up, and experimental results.

A. WVRTC Testing Facility

The West Virginia Robotic Technology Center (WVRTC) at West Virginia University (WVU) is a development and technology test bed facility supporting Goddard Space Flight Center's Satellite Servicing Capabilities Office (SSCO).The WVRTC was established in 2009 and is located off-campus in the Robert H. Mollohan Research Center in Fairmont, West Virginia. The Center houses multiple robotic platforms with an objective of supporting full-scale testing of satellite servicing technology and operational procedures.[21]

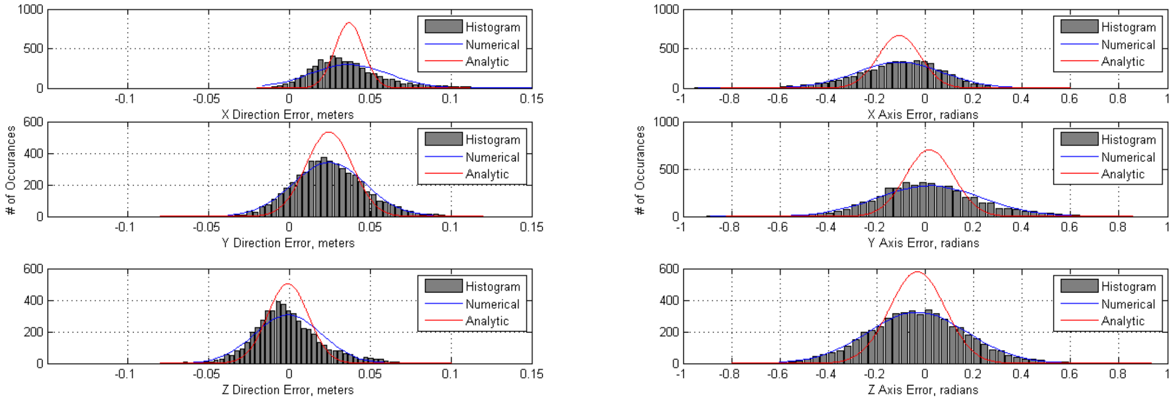
The experimental validation test procedures using the SwissRanger system were performed in the Satellite Servicing Robotic Workstation (SSRW) of the WVRTC, shown in Figure 5. This station houses three industrial robot manipulators with 7 DOF motion capability. The positional accuracy of the robot manip-



(a) Translation error histogram.

(b) Rotation error histogram.

Figure 3: Numeric (dashed) and analytic (solid) PDFs conform closely for errors in rotation (a) and translation (b) for point clouds which match perfectly. The PDFs are overlaid on a histogram from a 10,000-run Monte Carlo simulation.



(a) Translation error histogram.

(b) Rotation error histogram.

Figure 4: The analytic covariance is slightly optimistic because it neglects errors introduced from imperfect matching of measured points to model points.

ulator equipped with the SwissRanger system mounted to the manipulator end effector locations serves as the source of the truth data used for comparison to the sensor-based pose determination.

The SSRW robot manipulators are the Yaskawa Motoman SIA50D and SIA10D models. The superior dexterity of a 7 DOF robot manipulator enables the manipulator to re-orient itself relative to the object and environment — a characteristic critical for proposed satellite servicing objectives and for test evaluations such as that presented in this work. In addition, a robot manipulator with advanced repeatability capabilities, based on robot joint kinematic evaluation, is essential for accurate data acquisition. The repeatability performance parameter and additional specifications of the SIA50D model used in this test are summarized in Table 1.

Table 1: Test robot manipulator specifications and performance parameters.

Robot Model	DOF	Payload	Workspace Volume	Max Velocity	Repeatability
Motoman SIA50D	7	50 kg	$12.74\,\mathrm{m}^3$	200 $^{\mathrm{deg}}/_{\mathrm{s}}$ (wrist twist)	±0.1 mm

B. Testing Procedure

In order to collect experimental data, the SwissRanger was mounted on a Motoman robotic arm end effector with a custom attachment bracket (Figure 6).

The SwissRanger's coordinate frame was created using a Leica laser tracker (AT 901-MR). Leica measurements were collected on three sides of the camera and a frame was affixed to the center of the SwissRanger's

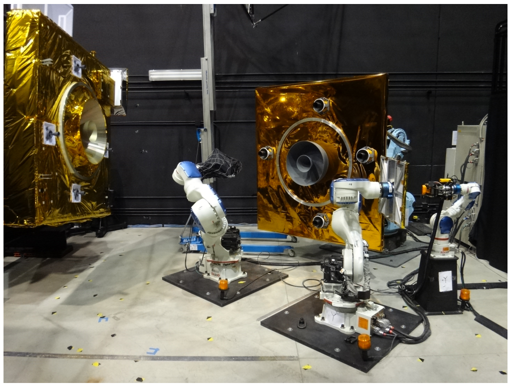
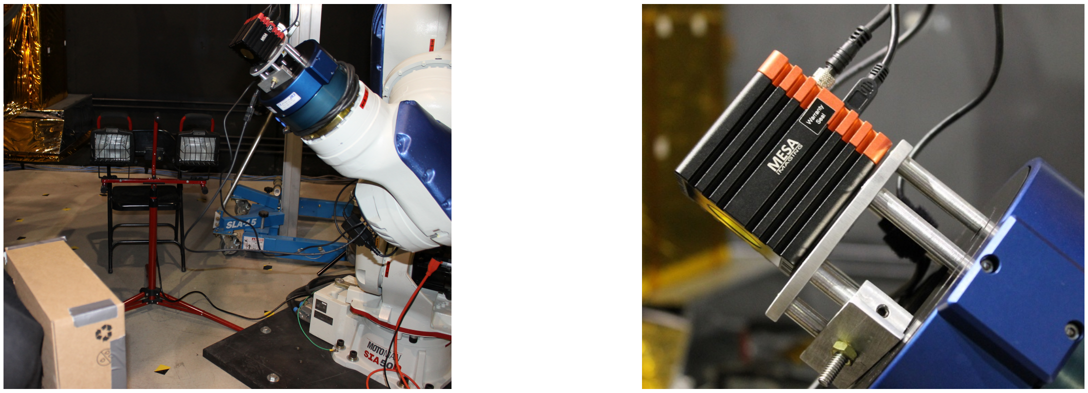


Figure 5: The Satellite Servicing Robotic Workstation at the WVRTC.



(a) SwissRanger attached to Motoman SIA50D.

(b) SwissRanger mounted on end effector.

Figure 6: Complete experimental test set-up.

receive optics (with the z^+-direction going out of the camera along the boresight). A similar procedure with the Leica tracker was used to established body-fixed frames on the various objects that were observed (e.g. the box shown in this paper). These frames, along with recorded robot positions, are used to establish a "truth" for comparison with the pose measured/estimated by the SwissRanger.

The observed objects were individually fixed to a duvetyne fabric-covered test platform. The black duvetyne fabric works as a light absorption surface and is used to emulate the space environment as best as possible in the laboratory environment.

For each observed object, both static and dynamic tests were performed. The static tests are useful for assessing pose performance, while the dynamic tests demonstrate how performance changes with viewing geometry. For the dynamic tests, a number of simulated rendezvous trajectories were performed. These pre-programmed trajectories followed straight line approach paths and were approximately 30 seconds in duration. During each one of these tests, SwissRanger data was collected at a rate of 30 Hz. Each SwissRanger frame is stored in an individual time-stamped file that includes a 3D point cloud (x, y, z coordinates in the camera frame), range image, and intensity image. Simultaneously, the rotation angles and position of the Motoman SIA50D end effector were recorded into a robot position log.

C. SiwssRanger Data Pre-processing

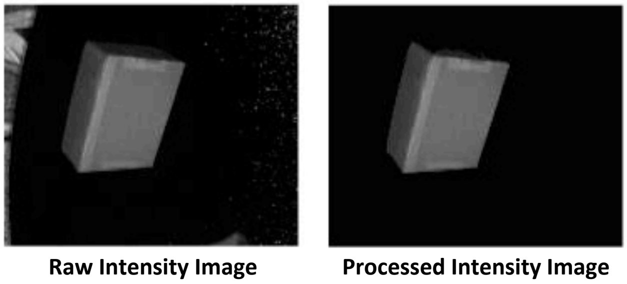
Each SwissRanger data file containes the x, y, z coordinates, range, and intensity for each of the 25,344 (144×176) pixels on the focal plane array. As was mentioned before, black duvetyne fabric was used to emulate a space background for these tests. Practical limitations, however, precluded the entire SwissRanger FOV to be covered by fabric at the beginning of the test (when the sensor is farthest away from the observed object). This resulted in testing artifacts around the image edges. To simulate the type of image that would be encountered in space, these testing artifacts need to be removed. This was achieved through a combination of intensity-based and range-based filters. An example of this is shown in Figure 7

D. Empirical Characterization of Pose Performance with Static Test Data

A series of static tests were performed to assess the typical pose performance that may be achieved when viewing a simple box with the SwissRanger. The SwissRanger and the observed object were placed in a static configuration and two thousand images were collected of the unchanging scene. These two thousand images were then separately imported and processed through the pose estimation algorithms described above. The resulting distribution of pose solutions describes the pose error due to sensor noise. The results can be seen in Figure 8 and Table 2.

E. Approach Simulation Processing

Next, pose estimation was performed on the dynamic approach data to demonstrate the utility of LIDAR for non-cooperative rendezvous. The particular case shown here has a total duration of 28 seconds. Each image taken during this approach was filtered to remove testing artifacts and processed through the ICP-based pose estimation algorithm. A few example images from the approach are shown in Figure 9. The pose



Raw Intensity Image

Processed Intensity Image

Figure 7: Example LIDAR intensity image containing testing artifacts (left) that are removed through simple intensity and range checks to create an artifact-free image (right).

estimates created using SwissRanger data were compared to the known values from the robot position logs. The results are shown in Figure 10.

As seen in Figures 10–11 the ICP was able to accurately estimate the pose of the camera relative to the the target body during an approach, with small errors. The covariance of the system was calculated using Eq. 24 and encompasses most of the errors seen in the estimation. While the noise is of approximately the correct magnitude, there a pose bias that is not accounted for in the covariance shown in these plots. This bias, which is likely due to poor observations on one of the box faces, has a magnitude of about 1 cm in position and 1° in attitude.

VIII. Conclusion

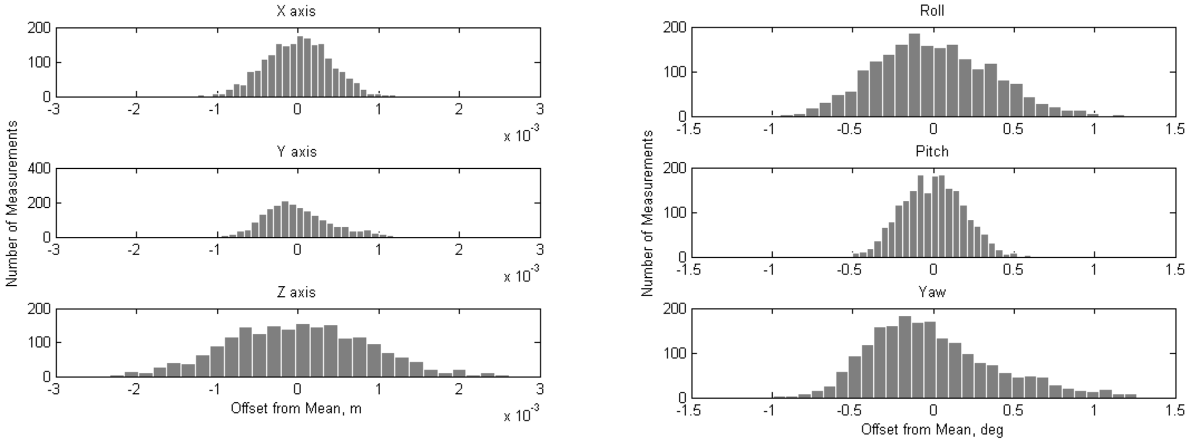
This manuscript examines the effectiveness of Flash LIDAR sensors for autonomous rendezvous and docking of non-cooperative satellites. The report makes use of the standard iterative closest point (ICP) algorithm for obtaining an optimal pose estimate, and demonstrates pose determination along an approach recorded using a SwissRanger 4000 ToF camera. A derivation is presented for analytic pose covariance computation, which is numerically validated using a Monte Carlo simulation. Additional forward work is necessary to accurately capture the affects of imperfect matching between the measured point cloud and the model point cloud within an analytic covariance framework.

A key obstacle with ICP — its narrow basin of convergence — is discussed, and the OUR-CVFH pose initialization strategy (which requires no initial guess, but produces an inexact pose) is presented briefly as a solution. Future work will further detail the use of OUR-CVFH for filter initialization and re-initialization, and will demonstrate the non-cooperative pose estimation process in real-time.

This analysis offers positive indications for the applicability of Flash LIDARs in relative navigation with

Table 2: Static pose distribution statistical values.

	Mean	Std. Dev.	Skewness	Kurtosis
X, m	-0.0023	0.000402	-0.0535	2.8555
Y, m	-0.0146	0.000421	0.5822	3.2828
Z, m	1.1848	0.000884	0.0493	2.8739
Pitch, deg	76.1756	0.3622	0.2347	2.8550
Yaw, deg	40.6986	0.1911	0.1064	2.8515
Roll, deg	-9.1522	0.3994	0.6781	3.2362



(a) Position deviation.

(b) Rotation deviation.

Figure 8: Distribution of static pose estimation.

non-cooperative targets. The ability to estimate pose with the data produced by a LIDAR sensor has been well-examined and is a proven technology for cooperative applications, but hasn't yet been flown in missions involving non-cooperative vehicles. Such non-cooperative vehicles (*e.g.*, legacy satellites) were built without consideration for serviceability, refueling, or docking, and the ability to accomplish these tasks autonomously would greatly improve the life expectancy of expensive space technologies.

Acknowledgments

The authors would like to thank Patrick Lewis of the WVRTC for help in collecting experimental test data with the SwissRanger in the WVRTC Fairmont facility. This work was made possible through contract NNG10CR16C in support of the NASA GSFC Satellite Servicing Capabilities Office (SSCO).

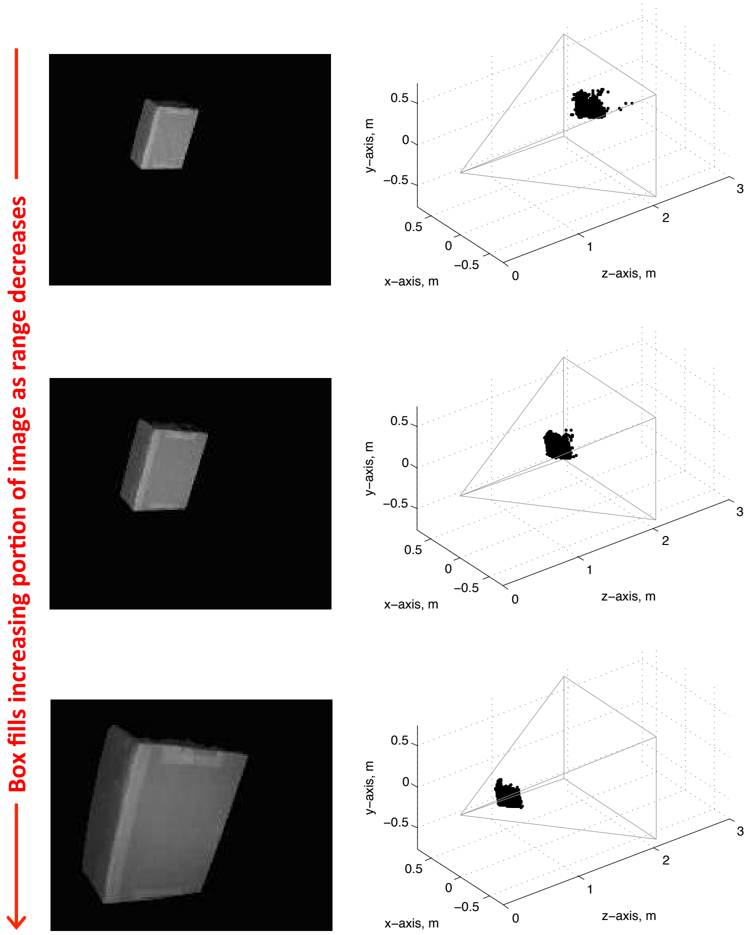
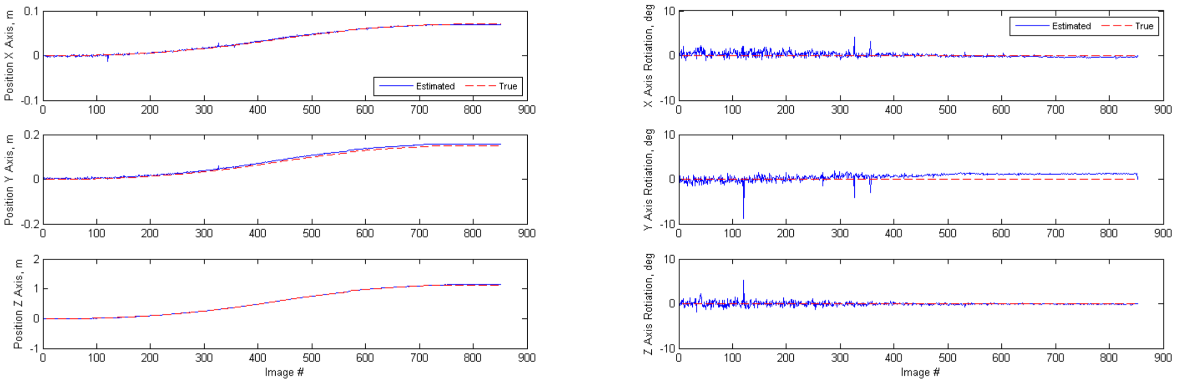


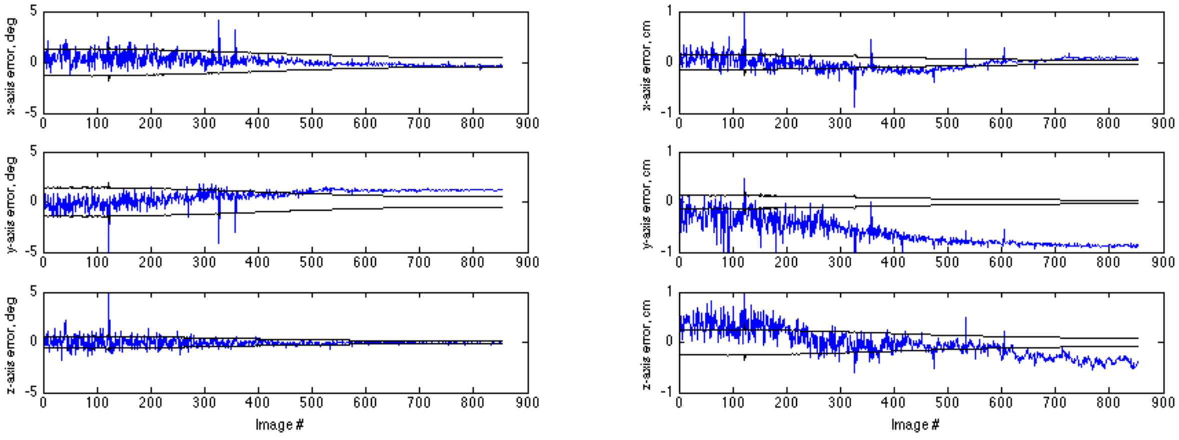
Figure 9: Example intensity images and 3D point clouds from SwissRanger during rendezvous test. Intensity images on the left show results after pre-processing has removed test artifacts. The gray pyramid in the right-hand 3D image shows the SwissRanger's field of view.



(a) Position change from initial.

(b) Rotation change from initial.

Figure 10: True and estimated approach tracking results.



(a) Rotation error.

(b) Translation error.

Figure 11: Error between estimated pose (from SwissRanger measurements and ICP algorithm) and true pose (from robot log files). The blue line is the error and the black line is the 3σ covariance.

References

[1] Naasz, B. J. and Moreau, M., "Autonomous RPOD Challenges for the Coming Decade," No. AAS 12-065, Breckenridge, CO, February 2012.

[2] Amzajardian, F., Petway, L., Hines, G., Robuck, V., and Reisse, R., "Lidar Sensors for Autonomous Landing and Hazard Avoidance," *AIAA SPACE Conference 2013*, No. 10.2514/6.2013-5312, San diego, CA, September 2013.

[3] Christian, J. A. and Cryan, S., "A Survey of LIDAR Technology and its Use in Spacecraft Relative Navigation," No. 10.2514/6.2013-4641, August 2013.

[4] Christian, J., Robinson, S., D'Souza, C., and Ruiz, J., "Cooperative Relative Navigation of Spacecraft Using Flash Light Detection and Ranging Sensors," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 452–465.

[5] Christian, J. A., Hinkel, H., D'Souza, C. N., Maguire, S., and Patangan, M., "The Sensor Test for Orion RelNav Risk Mitigation (STORRM) Development Test Objective," *Guidance, Navigation, and Control conference*, No. 2011-6260, AIAA, August 2011.

[6] Carreau, M., "Dragon Successfully Berthed to Station," *Aviation Week and Space Technology*, 25 May 2012.

[7] Galante, J., Van Eepoel, J., Strube, M., Gill, N., Gonzalez, M., Hyslop, A., and Patrick, B., "Pose Measurement Performance of the Argon Relative Navigation Sensor Suite in Simulated Flight Conditions," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, MN, 13-16 August 2012.

[8] Rohrschneider, R., Tandy, W., Bladt, J., and Gravseth, I., "Simulation Results of Rendezvous and Docking with the International Space Station Using Only 3D Range Images," *35th Annual AAS Guidance and Control Conference*, Breckenridge, CO, 3-8 Feb 2012.

[9] Rusu, R. and Cousins, S., "3D is here: Point Cloud Library (PCL)," *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 9-13 May 2011.

[10] Eggert, D. W., Lorusso, A., and Fisher, R. B., "Estimating 3-D rigid body transformations: a comparision of four major algorithms," *Machine Vision and Applications* , 1997, pp. 272–290.

[11] Arun, K. S., Huang, T. S., and Blostien, S. D., "Least-Squares Fitting of Two 3-D Point Sets," *The Journal of Navigation*, Vol. 9, 1987, pp. 698–700.

[12] Haralick, R., Joo, H., Lee, H., Zhuang, X., Vaidya, V., and Kim, M., "Pose Estimation from Corresponing Point Data," *IEEE Transactions On Sysmtems, Man and Cybernetics*, Vol. 19, No. 6, 1989.

[13] Horn, B. K. P., "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, Vol. 4, 1987, pp. 629–642.

[14] Horn, B., Hilden, H., and Negahdaripour, S., "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America*, Vol. 5, 1988, pp. 1127–1135.

[15] Walker, M. W., Shao, L., and Volz, R., "Estimating 3-D location parameters using dual number quaternions." *CVGIP: Image Understanding*, Vol. 54, 1991, pp. 358–367.

[16] Besl, P. J. and McKay, N. D., "A Method for Registration of 3-D Shapes." *IEEE Transactions of Pettern Analysis and Machine Intelligence*, Vol. 14, 1992, pp. 239–256.

[17] Crassidis, J. and Junkins, J., *Optical Estimation of Dynamic Systems*, CRC Press, Boca Raton, FL, 2004.

[18] Hull, D., *Optimal Control Theory for Applications*, Springer, New York, NY, 2003.

[19] Aldoma, A., Tombari, F., Rusu, R. B., and Vincze, M., "OUR-CVFH — Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation," *Pattern Recognition*, edited by A. Pinz, T. Pock, H. Bischof, and F. Leberl, Springer Berlin Heidelberg, 2012, pp. 113–122.

[20] Rusu, R. B., Blodow, N., and Beetz, M., "Fast Point Feature Histograms (FPFH) for 3D registration," *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 3212–3217.

[21] Evans, T., Christian, J., Marani, G., and Lewis, P., "Testing Facility for Autonomous Robotics and GNC Systems at West Virginia University," *AAS Guidance & Control Conference*, Breckenridge, CO, February 2014.