Received: 8 June 2021 | Revised: 27 June 2022 | Accepted: 23 July 2022

DOI: 10.33012/navi.562

ORIGINAL ARTICLE

Accurate Covariance Estimation for Pose Data From Iterative Closest Point Algorithm

Rick H. Yuan | **Clark N. Taylor** | **Scott L. Nykl**

Electrical and Computer Engineering, Air Force Institute of Technology, Ohio, USA

Correspondence
Clark Taylor
Email: clark.taylor.3@au.af.edu/ clark.n.taylor@gmail.com

Abstract

One of the fundamental problems of robotics and navigation is the estimation of the relative pose of an external object with respect to the observer. A common method for computing the relative pose is the iterative closest point (ICP) algorithm, where a reference point cloud of a known object is registered against a sensed point cloud to determine relative pose. To use this computed pose information in downstream processing algorithms, it is necessary to estimate the uncertainty of the ICP output, typically represented as a covariance matrix. In this paper, a novel method for estimating uncertainty from sensed data is introduced.

Keywords

covariance estimation, ICP, iterative closest point, uncertainty estimation

1 | INTRODUCTION

One of the key algorithms used when working with 3D point clouds is the iterative closest point (ICP) algorithm introduced by Chen and Medioni (1992) and Besl and McKay (1992). Since then, the ICP algorithm has been used in many robotics applications such as search and rescue, powerplant inspection, shoreline monitoring, and autonomous driving (Pomerleau et al., 2015). It is also used in 2D or 3D self-localization methods such as simultaneous localization and mapping (SLAM; Mendes et al., 2016). Input data for the ICP algorithm can come from lidar, monocular, or stereo vision sensors.

The ICP algorithm consists of iterating through the following two steps:

1. For every point in the measured point cloud, find the closest point in the reference point cloud.
2. Compute the best 6D pose (location and orientation) to align each measured point with its closest reference point. Note that finding the best 6D pose to align points with a 1-to-1 correspondence can be solved in closed form using quaternions or the singular value decomposition (SVD) of a 3x3 matrix (e.g., Arun et al. [1987], Horn [1987], and Horn et al. [1988]).

By iteratively executing this process, the ICP algorithm outputs the best-fit 6d pose, as well as the correspondence between the sensed and reference points. This best-fit rotation and translation estimate is what can be used by downstream algorithms to complete their tasks.

To use a position and rotation estimate in a navigation framework, knowing the accuracy of the derived estimate can be as important as the estimate itself. For example, many robotics applications use the output of ICP to form a pose graph, where the 6d transform between two point clouds is used to estimate the movement of the robot. To estimate the robot trajectory over time, a trajectory that is the best match (probabilistically) to the measured 6d transforms is found. What makes the best match depends on the uncertainty estimated for the ICP output. Unfortunately, as mentioned in Censi et al. (2008), most ICP algorithms do not return estimates of their own uncertainty. Therefore, in this paper, we look to improve the estimation of the uncertainty obtained when running the ICP algorithm. We do not propose modifying the ICP algorithm, itself, but simply the estimation of its uncertainty after ICP has converged to a solution. Therefore, Section 1.2 only covers methods for estimating uncertainty, and not the ICP algorithm itself.[1]

The motivating application for this paper is two aircraft attempting to fly in formation. There will be some safety margin built into the desired (relative) location of the aicraft to ensure a collision does not occur. However, if the ICP uncertainty is high, then the safety margin can be violated even without external disturbances due to errors in the ICP algorithm. This, combined with external disturbances, can quickly lead to catastrophic failures. Therefore, modeling the safety margin using an accurately estimated uncertainty is vital to ensuring safe operation. The focus of this paper is computing an accurate estimate of the uncertainty of the ICP output pose.

1.1 | Notation

For clarity, we utilize the following notation in this paper:

- Scalar: denoted by a non-boldface letter (e.g., k or N)
- Vector: denoted by a **boldface** lowercase letter (e.g., $\mathbf{n}$)
- Matrix: denoted by a **boldface** uppercase letter (e.g., $\mathbf{K}$)
- Set: denoted by curly brackets (e.g., a set of vectors $\{\mathbf{p}_r\}$); the elements of the set are denoted as $\mathbf{p}_{r,i}$, where i is the index of that element.
- Identity matrix: denoted by $\mathbf{I}$ followed by a subscript denoting its size (e.g., $\mathbf{I}_6$ is a 6×6 identity matrix)

1.2 | Prior Work

To understand how uncertainty has been computed in the past, it is necessary to precisely define the objective of Step 2 in the ICP algorithm. The second step of the ICP algorithm is based on finding the relative pose that minimizes the sum of squared distances between the reference points and the sensed points. This step optimizes the following cost function (Markley & Mortari, 2000):

$$\arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \| \mathbf{p}_{s,i} - \mathbf{R}\mathbf{p}_{r,\phi(i)} - \mathbf{t} \|^2 \tag{1}$$

[1] The literature on efficiently executing ICP, navigating using ICP, applying ICP-based algorithms to different robotics scenarios, and many other aspects of ICP is extensive and beyond the scope of this paper, though a recent survey of some such topics can be found in Kolhatkar and Wagle (2021).

where $\mathbf{R}$ and $\mathbf{t}$ are the relative rotation and translation between the reference and sensed points, $\mathbf{p}_{s,i}$ is the i-th sensed point, $\phi(\cdot)$ is a function that takes in an index to a point in the sensed cloud and returns an index to a matching (closest) point in the reference cloud, and $\mathbf{p}_{r,\phi(i)}$ is the position of the $\phi(i)$-th point in the reference cloud.

As the solution to Step 2 is a least-squared problem, there are well-known methods for computing the covariance for the result. We term this the *Jacobian method* and describe it in the next section. We then describe some modifications that have been made in the literature to obtain more accurate uncertainty estimates. In simulation, the true covariance of this result can be obtained through a Monte-Carlo method.

1.2.1 | Jacobian Method

The second step of the ICP algorithm is based on finding the relative pose that minimizes the sum of squared distances between the reference points and the sensed points defined in Equation (1). To understand the uncertainty resulting from this step, we will represent this step as a least-squares equation using normal form, yielding:

$$\mathbf{J}\mathbf{x} = \mathbf{z}$$
$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_{1,x} \\ \mathbf{z}_{1,y} \\ \mathbf{z}_{1,z} \\ \vdots \\ \mathbf{z}_{n,z} \end{bmatrix} \tag{2}$$

where $\mathbf{z}_i = \mathbf{p}_{s,i} - \mathbf{R}\mathbf{p}_{r,\phi(i)} - \mathbf{t}$ is a 3-vector and consists of an x, y, and z element, and $\mathbf{x}$ is a vector of changes to the relative pose expressed as:

$$\mathbf{x} = [\Delta t_1 \; \Delta t_2 \; \Delta t_3 \; \Delta r_1 \; \Delta r_2 \; \Delta r_3]^T \tag{3}$$

where t_j is the j-th element of the three-element translation vector, r_j is the rotation around the j-th axis, and $\mathbf{J}$ is the differential change in $\mathbf{z}$ with respect to t_j and r_j. Specifically:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial z_{1,x}}{\partial t_1} & \frac{\partial z_{1,x}}{\partial t_2} & \frac{\partial z_{1,x}}{\partial t_3} & \frac{\partial z_{1,x}}{\partial r_1} & \frac{\partial z_{1,x}}{\partial r_2} & \frac{\partial z_{1,x}}{\partial r_3} \\ \frac{\partial z_{1,y}}{\partial t_1} & \frac{\partial z_{1,y}}{\partial t_2} & \frac{\partial z_{1,y}}{\partial t_3} & \frac{\partial z_{1,y}}{\partial r_1} & \frac{\partial z_{1,y}}{\partial r_2} & \frac{\partial z_{1,y}}{\partial r_3} \\ \frac{\partial z_{1,z}}{\partial t_1} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \frac{\partial z_{n,z}}{\partial t_1} & \frac{\partial z_{n,z}}{\partial t_2} & \frac{\partial z_{n,z}}{\partial t_3} & \frac{\partial z_{n,z}}{\partial r_1} & \frac{\partial z_{n,z}}{\partial r_2} & \frac{\partial z_{n,z}}{\partial r_3} \end{bmatrix} \tag{4}$$

The weighted least-squares problem solution can be solved as:

$$\mathbf{x} = (\mathbf{J}^T \mathbf{W}\mathbf{J})^{-1} \mathbf{J}^T \mathbf{W}\mathbf{z} \tag{5}$$

where $\mathbf{W}$ is the weighting matrix. For data with a known covariance, $cov(\mathbf{z})$, $\mathbf{W} = cov(\mathbf{z})^{-1}$, which also means $\mathbf{W}$ is symmetric. To find the covariance of $\mathbf{x}$:

$$E[\mathbf{x}\mathbf{x}^T] = E\left[(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T\mathbf{J}(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\right] \tag{6}$$

$$= (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}E[\mathbf{z}\mathbf{z}^T]\mathbf{W}^T\mathbf{J}(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1} \tag{7}$$

$$= (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{W}^{-1}\mathbf{W}^T\mathbf{J}(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1} \tag{8}$$

$$= (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{J}(\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1} \tag{9}$$

$$= (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1} \tag{10}$$

Note that, when computing the covariance using the Jacobian method, we are assuming that the first step of the ICP algorithm perfectly matches every sensed point to the correct point in the reference model. With real data, however, there are two sources of error that are not considered by the Jacobian method. First, because the reference model consists of a set of points, there may not be a point at the exact location that the sensed point represents. Second, even if there was a reference point at the exact noise-free source of the sensed point, Step 1 of the ICP algorithm may match the sensed point to the wrong reference point. If we were to think of the reference model as a set of points lying on a surface, both of these noise sources introduced lose information along the surface, but preserve information orthogonal to the surface.

1.2.2 | *Advanced Methods*

In Brossard et al. (2020), Censi (2007), Landry et al. (2019), Mendes et al. (2016), and Prakhya et al. (2015), more advanced methods of estimating ICP covariance are introduced. Censi (2007) lays the groundwork for this area of research by introducing a closed form estimation of ICP's covariance that considers the additional error sources described above and demonstrates it on a 2D scan matching problem. Prakhya et al. (2015) extends Censi's method into a 3D application. Mendes et al. (2016) utilizes this technique to apply ICP to a ground robot SLAM application. Brossard et al. (2020) examines the sources of error named by Censi: wrong convergence, underconstrained situations, and sensor noise. The most comparable work to ours was found in Prakhya et al. (2015). We describe how this method (termed the *Prakhya method* after the first author) solves for ICP covariance in the following subsection.

Prakhya et al. (2015) implements a closed form method of estimating the uncertainty of an ICP fit. The key insight of this method is finding a method to include the ICP cost function (Step 1 of ICP) within the Jacobian computation. The output covariance is calculated through Equations (11) and (12):

$$C = \sum_{i=1}^{n} \| \mathbf{p}_{s,i} - \mathbf{R}\mathbf{p}_{r,\phi(i)} - \mathbf{t} \|^2 \tag{11}$$

$$cov(\mathbf{x}) = \left(\frac{\partial^2 C}{\partial x^2}\right)^{-1}\left(\frac{\partial^2 C}{\partial z \partial x}\right) cov(\mathbf{z}) \left(\frac{\partial^2 C}{\partial z \partial x}\right)^{T}\left(\frac{\partial^2 C}{\partial x^2}\right)^{-1} \tag{12}$$

where $\mathbf{x}$ is the output state vector, $cov(\mathbf{x})$ is the covariance of the output pose, C is the cost function optimized by ICP, and $\mathbf{z}$ is the input point vector.

Derivation: Let F be the ICP function where $\mathbf{x} = F(\mathbf{z})$. The first-order Taylor series approximation at $\mathbf{z} = \mathbf{z}_0$ is then defined as:

$$\mathbf{x} = F(\mathbf{z}) \approx F(\mathbf{z}_0) + \frac{\partial F}{\partial z_0}(\mathbf{z} - \mathbf{z}_0) \tag{13}$$

$$\mathbf{x} = F(\mathbf{z}) \approx F(\mathbf{z}_0) + \frac{\partial F}{\partial z_0}(\mathbf{z}) - \frac{\partial F}{\partial z_0}(\mathbf{z}_0) \tag{14}$$

To take the covariance of this form, the constants $F(\mathbf{z}_0)$ and $\frac{\partial F}{\partial z_0}(\mathbf{z}_0)$ can be ignored. Substituting into the expression, $cov(\mathbf{Bz} + \mathbf{c}) = \mathbf{B}cov(\mathbf{z})\mathbf{B}^T$, the covariance can now be defined as:

$$cov(\mathbf{x}) \approx \frac{\partial F}{\partial z_0} cov(\mathbf{z}) \left(\frac{\partial F}{\partial z_0} \right)^T \tag{15}$$

F, however, is not in closed form and it is difficult to compute the partial derivative of F with respect to $\mathbf{z}_0$. Censi (2007) solves this problem with the implicit function theorem:

$$\frac{\partial F}{\partial z_0} = -\left(\frac{\partial^2 C}{\partial x^2} \right)^{-1} \left(\frac{\partial^2 C}{\partial z \partial x} \right) \tag{16}$$

Thus, substituting Equation (16) into Equation (15) yields the solution in Equation (12).

While this approach leads to significantly more accurate uncertainty results, the work introduced in this paper has two significant advantages over prior work. First, prior work has assumed the input covariance from the sensors is known a priori. This requires careful characterization of the sensors and may be violated as temperature or other environmental conditions change. We remove the requirement of a priori characterization in this work. Second, the approach proposed in this paper is computationally much faster than the Prakhya approach. Furthermore, we believe it to be conceptually easier to understand, enabling greater understanding of the uncertainty estimation problem for the ICP algorithm.

Another work that bears some similarity to our approach is Guehring (2001). In this work, Guehring propose modifying (weighting) the ICP least-squares solution by what we call *the point-to-point method* in this paper. Unfortunately, it does not show how this weighting can be used to compute an uncertainty (covariance) estimate for the final ICP results, nor does it show how this weighting affects the covariance estimate. Therefore, we do not directly compare against this approach in our results section.

1.3 | Motivation/Test Scenario

This research is motivated by the problem of automated aerial refueling (AAR). Aerial refueling is the process of transferring fuel from one aircraft (the tanker) to the receiving aircraft while in flight and is a key enabler of Air Forces around the world. Currently, a human is used to guide the boom (a large pipe that protrudes from the back of the tanker) into the correct spot on the receiving aircraft.

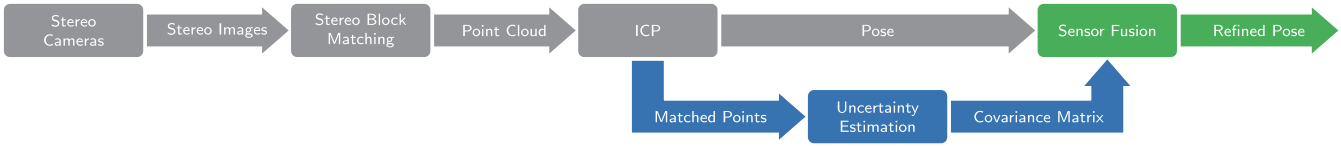

FIGURE 1 Stereo vision pipeline for automated aerial refueling (AAR)

AAR proposes to automate the task of guiding the boom and requires a very accurate relative pose between the tanker and receiving aircraft. To accurately estimate relative pose, our research group has proposed using stereo computer vision in Anderson et al. (2021), where two cameras are mounted to the underside of the tanker aircraft. Stereo block matching is applied to the two images from the cameras to generate a point cloud representing the object in the visual field. Then, the ICP algorithm is used to fit a known model of the aircraft against the visually generated point cloud to estimate the output pose. This process is outlined in Figure 1. Noise may be introduced at multiple levels throughout this process; there may be pixel error in the cameras or the wrong features may be matched in stereo block matching, for example. To guarantee safe operation of the AAR application, the uncertainty in the system must be carefully evaluated and characterized. The covariance estimation methods proposed in this paper were tested on a complete simulation environment of the AAR application, enabling us to identify and evaluate the uncertainty due to different sources of error in the pipeline.

In the remainder of this paper, we introduce our new technique for estimating covariance information of the ICP algorithm in Section 2. We present some preliminary results of our technique in Section 3, and conclude the paper in Section 4.

2 | METHODOLOGY

To enable accurate estimation of the ICP uncertainty while considering the point matching (first) step of the ICP algorithm, we propose a Kalman filter update-based technique. In this section, we first introduce the Kalman filter-based approach, while subsections fill in details necessary for the implementation of this approach.

There are three key attributes of the Kalman filter that we use to properly estimate the covariance of the ICP algorithm, namely:

1. The Kalman filter is a least-squared estimator. Typically, the Kalman filter is used to perform a least-squares estimate using both dynamics and measurement updates. In this case, however, we will be doing just a multi-measurement update step (i.e., no dynamics for this system). This attribute shows that a Kalman filter can be used to properly estimate the covariance of a least-squares solution.
2. As shown in Sorenson (1966), as long as the measurements are not correlated, different measurements from the same time step can be applied in any order to the Kalman filter and the same result will be achieved. This enables us to create a computationally efficient algorithm that applies the measurement from only one point match at a time. Even though only the current estimate and covariance are stored, the final covariance will accurately reflect the impact of all point matches in the ICP algorithm.

3. The **H** or measurement matrix of the Kalman filter can be used to express in which direction a particular measurement has information. By properly choosing the **H** matrix, we can create a covariance matrix that is an accurate representation of the ICP uncertainty. This concept of a measurement's direction is key to properly estimating the uncertainty of the ICP algorithm and will be discussed in detail later in this section.

Having motivated our choice of a Kalman filter, we show our specific implementation for this problem in Algorithm 1. Note that this algorithm is implemented after the ICP algorithm has completed execution, meaning we have both the best relative pose estimate (**R, t**) between the point clouds and a mapping, $\phi(i)$, that relates points in the sensed cloud to points in the reference cloud. Also note that the δ_r values are incremental rotations around the 1st, 2nd, and 3rd axes.

In Algorithm 1, we start with a large initial value for the covariance (theoretically, infinity) and, with each measurement reduce the uncertainty depending on the new information given by the measurement. The unique aspects of our approach include:

1. The measurement covariance is estimated from the data rather than being an input. The actual calculation is shown on Line 4 in Algorithm 1 and is further discussed in Section 2.1.
2. Each measurement only reduces the covariance in a particular direction determined by **H**. The key working principle is that the measurement model, **H**, is constructed so that each sensed point only provides information (a reduction in covariance) in the direction denoted by the vector, **n**. The underlying assumption is that if there were any error orthogonal to the normal vector, that error would cause the sensed point to be registered to the next point over. Thus, all the information orthogonal to the normal vector is hidden from the ICP algorithm while the information parallel to the normal is preserved. Sections 2.2 and 2.3 describe two approaches to computing the normal vector, **n**.

2.1 | Measurement Noise Estimation

Any approach to calculate the output noise must have a good understanding of the input noise. Note that the papers cited in Section 1.2.2 assumed that the input noise was known a priori. In our application, however, we are generating point clouds from a stereo vision setup. An inherent property of point clouds generated from stereo cameras is that the noise is strongly and non-linearly dependent on range. Thus, it is necessary to re-compute the input noise of the sensor for each image pair.

In a previous automated aerial refueling paper by Johnson et al. (2017), this error was characterized every 0.5 m from 30- to 100-m away from the camera. To utilize this method, however, creates a heavy characterization requirement. Nor can it take into account in-flight scenarios such as rain or out-of-focus cameras that may cause a change in the input uncertainty.

Therefore, instead of attempting to pre-characterize the sensor noise, we propose estimating the input noise from the data itself. Specifically, we can estimate the measurement noise, σ_m^2, as the mean squared error of the distances between the matched points as shown on Line 4 of Algorithm 1. Assuming the ICP algorithm

ALGORITHM 1
Kalman Filter-Based ICP Covariance Estimation

Inputs : $\{\mathbf{p}_s\}, \{\mathbf{p}_r\}, \mathbf{R}, \mathbf{t}, \phi()$
Outputs: P, the covariance matrix for ICP

//Initialize $\mathbf{P}$ *to a large value*;
$\mathbf{P} = 1e6 * \mathbf{I}_6$;
//Estimate the noise on the measurements;
$\sigma_m^2 = \frac{1}{N}\sum_{i=1}^{N} \| \mathbf{p}_{s,i} - \mathbf{R}\mathbf{p}_{r,\phi(i)} - \mathbf{t} \|^2$
//Take each measurement (sensed point) individually and update $\mathbf{P}$;
for $i \leftarrow 1$ **to** N **do**
  $\mathbf{n} = \text{ComputeNormal}(\mathbf{P}_{s,i}, \{\mathbf{P}_r\}, \mathbf{R}, \mathbf{t}, \phi(i))$;
  //Make a temporary variable $\mathbf{v}$;
  $\mathbf{v} = \mathbf{R}\mathbf{p}_{r,\phi(i)}$;
  $\delta\mathbf{r}1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \mathbf{v}$;
  $\delta\mathbf{r}2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \mathbf{v}$;
  $\delta\mathbf{r}3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{v}$;
  $\mathbf{H} = [\mathbf{n}_x\, \mathbf{n}_y\, \mathbf{n}_z\, \delta\mathbf{r}1 \cdot \mathbf{n}\ \delta\mathbf{r}2 \cdot \mathbf{n}\ \delta\mathbf{r}3 \cdot \mathbf{n}]$;
  $S = \mathbf{H}\mathbf{P}\mathbf{H}^T + \sigma_m^2$;
  $\mathbf{K} = \mathbf{P}\mathbf{H}^T S^{-1}$;
  $\mathbf{P} = (\mathbf{I}_6 - \mathbf{K}\mathbf{H})\mathbf{P}$;
end
return **P;**

has converged onto the correct global minimum, σ_m^2 should accurately reflect the variance of the sensor noise orthogonal to the local surface.

Note that, for this calculation to be valid, there are a few assumptions that should be satisfied. First, the ICP algorithm needs to converge to the right solution. Similar to the other advanced approaches described in Section 1.2.2, the proposed algorithm does not account for the possibility of the ICP algorithm converging to an incorrect solution. Second, we assume that the sensor noise is evenly distributed in all directions, following a spherical Gaussian distribution that is identical for each sensed point. Third, we assume all the sensed points are independent of each other. In Section 3, we present results demonstrating the efficacy of the proposed approach and its robustness to different imaging conditions.

2.2 | Computing n from Point-to-Point Constraints

To execute the Kalman filter-based covariance estimation approach outlined in Algorithm 1, the *ComputeNormal* function must be defined. In this subsection, we describe a simple approach—the point-to-point method—to estimating the

normal that leads to improved covariance results compared with the Jacobian method.

The point-to-point method, illustrated in Figure 2, computes the normal vector as the difference between the sensed point (i) and its corresponding reference point ($\phi(i)$). Because this vector should be normalized, it is computed as:

$$\mathbf{n} = \frac{\mathbf{p}_{s,i} - \mathbf{p}_{r,\phi(i)}}{\| \mathbf{p}_{s,i} - \mathbf{p}_{r,\phi(i)} \|} \tag{17}$$

While this method works well in general, we also found that when the noise is small and/or the reference points are spread far apart, the normal vectors generated by this method can be significantly different than the true normal vectors of the surface defined by the set of reference points. For example, consider the example shown in Figure 3. The true normal of the reference surface should be pointed straight up as the reference points are in a horizontal line. Due to the small error in the true normal direction compared with the spacing between the reference points, however, the computed normal in this case will be significantly different from the true normal.

To overcome this weakness of the point-to-point approach, we introduce another approach for estimating normal vectors—point-to-plane—in the following subsection.

2.3 | Point-to-Plane Approach for n Estimation

The basic idea behind the point-to-plane approach is illustrated in Figure 4. The reference surface is locally approximated by forming planes between reference points that are close together. Then, rather than simply computing the unit vector to the closest reference point, the normal is computed to the closest plane. Note that there are several variants of the ICP algorithm that utilize a point-to-plane technique to try and improve the robustness of the ICP algorithm itself, e.g., Low (2004) and Segal et al. (2009).

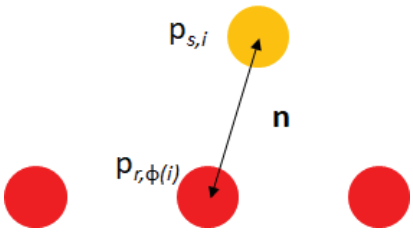

FIGURE 2 An example **n** vector from the point-to-point method

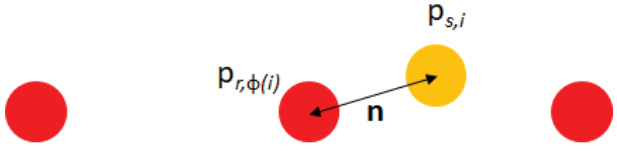

FIGURE 3 An example point-to-point scenario that generates a normal vector with significant error

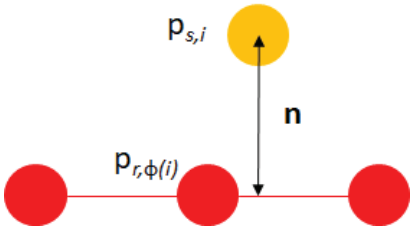

FIGURE 4 A simplified illustration of the point-to-plane approach for computing **n**

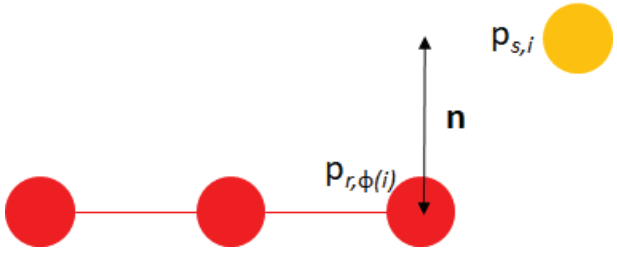

FIGURE 5 Point-to-plane when the sensed point is beyond the end of the reference points

In this application, **n** was found using a brute-force method. From the reference point cloud, the set {c} is populated from the eight closest neighbors in the reference model to $P_{r,\phi(i)}$. For each combination of two points in set {c}, a plane is defined between $P_{r,\phi(i)}$ and the two points, leading to different candidate planes. This produces a set of unit normal vectors local to $P_{r,\phi(i)}$. The plane normal vector with the greatest absolute dot product against the point-to-point normal vector is selected as the new **n** value.

While the point-to-plane method overcomes the problem discussed at the end of Section 2.2, there are two difficulties with this approach that can be observed. First, the reference model is assumed to be smooth and continuous. If the reference points have large discontinuities, then defining what the correct normal is is problematic. Second, information from points sensed past the edge of the reference model will not be properly accounted for. As shown in Figure 5, even though the presence of a point beyond the end conveys information orthogonal to the normal vector of the plane (i.e., the whole set of red points should shift right), the point-to-plane method will not consider that information. Therefore, the covariance information in the direction of the plane will be overestimated.

3 | RESULTS

To demonstrate the effectiveness of our new technique for estimating the uncertainty of an ICP-based registration, we conducted several different experiments as outlined in this section. In Section 3.1, we demonstrate the effectiveness of our technique for estimating covariance with a simulated box object where the sensed point cloud consists of points on the box with noise added. In Section 3.2, we extend the simulation to a full aerial refueling scenario. In this simulation, the points are derived from a complete stereo block matching process, including generating images of the object, simulating a realistic process for obtaining the sensed point clouds. Finally, we compare our method with the Prakhya et al. (2015) method (the most comparable method in the literature) in Section 3.3. While the covariance estimates are almost indistinguishable from Prakhya et al. (2015), we demonstrate

significant computational savings in our method. Furthermore, we present results showing our method is more robust to changes in the sensing scenario.

3.1 | 1x2x3 Box

In this section, several different ICP covariance estimation methods are run on a relatively simple shape: a 1x2x3 box. The box has dimensions of length 1 in the x direction, 2 in the y direction and 3 in the z direction. The faces orthogonal to the x-axis are the largest, so x is expected to have the lowest ICP covariance while the sides orthogonal to the z-axis are the smallest, indicating that the z translation's covariance should be the highest.

Sensed points are sampled from the mathematical definition of the shape and then corrupted with isotropic Gaussian noise. The magnitude of this Gaussian noise is swept over a logarithmic range to demonstrate behavior at different noise levels. For the Jacobian method, this known noise level is fed directly into Equation (5), while our method estimates the covariance of the inputs from the sensed data. At each noise level, 100 runs of the ICP algorithm are sampled to generate a Monte-Carlo covariance to compare the prediction against. Each run in the Monte-Carlo simulation has an independently sampled set of sensed points corrupted with independently generated noise.

In Figure 6, we show the uncertainty results that come from the Monte-Carlo (MC) runs using the Jacobian method from Section 1. Note that the MC runs (solid plots) show significant differences between the x, y, and z translation as we discussed earlier, but the Jacobian-based covariance estimates are almost the same for all three axes. (Unfortunately, they are so similar that only one axis' results can be viewed on the plot.) While not shown, a similar pattern is exhibited for the rotational angles. This demonstrates the fundamental weakness of computing covariance without explicitly considering the first step of the ICP algorithm.

In Figures 7 and 8, we show the results of the point-to-point and point-to-plane methods from Section 2, respectively. The point-to-point method performs much better than the Jacobian method. However, it still suffers from the error case shown in Figure 3, causing it to assume information was generated in more directions

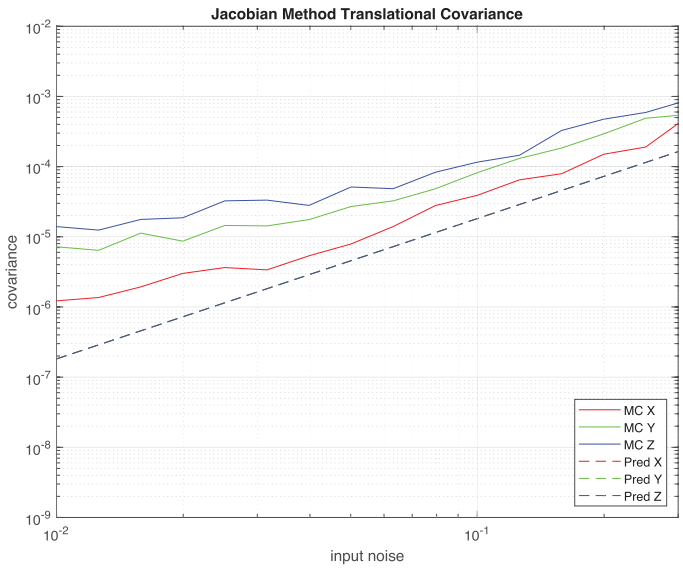

FIGURE 6 Results of Jacobian Method on a 1x2x3 box

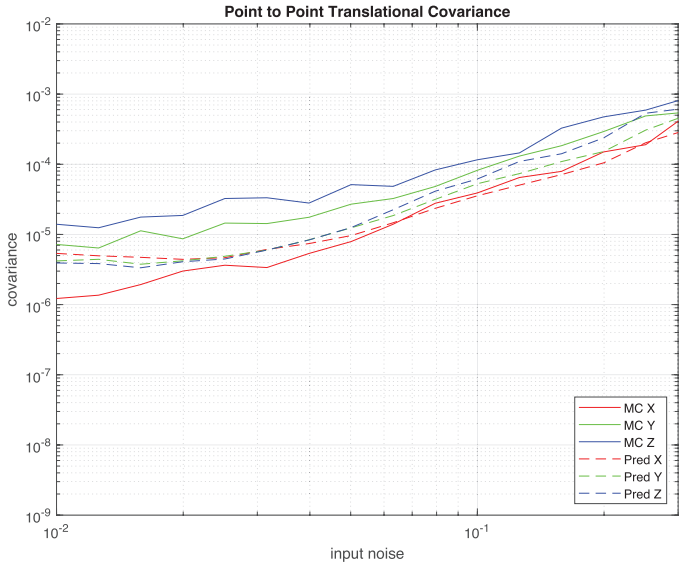

FIGURE 7 Results of point-to-point method on a 1x2x3 box

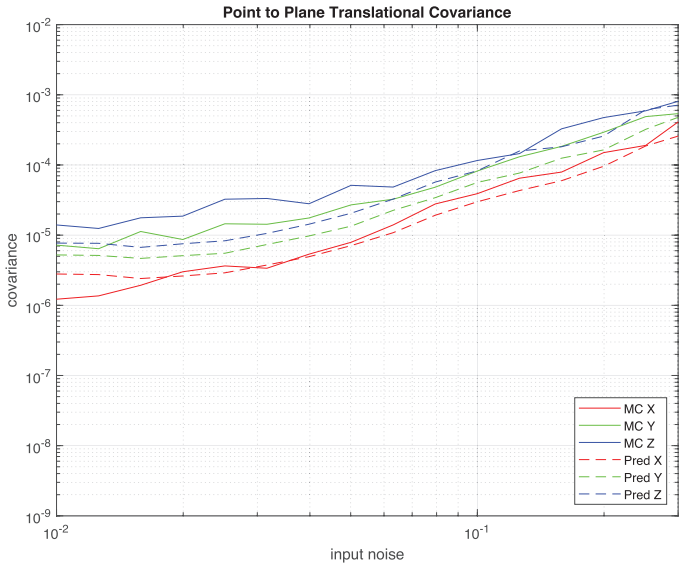

FIGURE 8 Results of point-to-plane method on a 1x2x3 box

than the box shape actually allows for. In aggregate, these optimistic errors combine in the extended Kalman filter (EKF) to result in an output covariance estimate that is more optimistic than the truth (Monte-Carlo estimate).

Finally, the performance of the point-to-plane method is shown in Figure 8. By explicitly limiting the information generated by each sensed point to the vector orthogonal to the planes generated by the reference points, errors due to point mismatches and finding points between reference points can be overcome. In Figure 8, specifically note that (a) there is now significant differentiation in the predicted accuracy between points that corresponds with the size of the orthogonal planes (x has the smallest covariance, z the largest) and (b) the plot lines of MC vs predicted are significantly closer together than for either the Jacobian or point-to-point techniques. Note that the results are slightly less accurate at the lower end of the input noise range due to the noise becoming lower than the sample rate of the reference model.

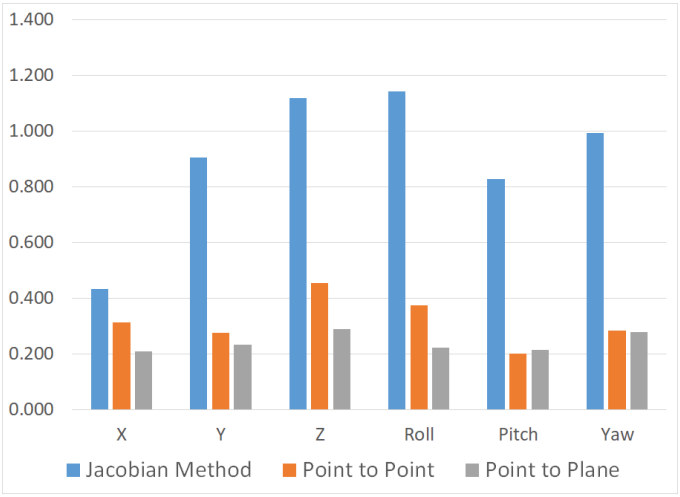

FIGURE 9 RMSLE plot on a 1x2x3 box

To more concisely describe the closeness of plots together, we will use a new metric, the root-mean-square log error (RMSLE) for future plots. The RMSLE between predicted and MC run covariance values can be computed as:

$$RMSLE(d) = \sqrt{1/n \sum_{i=1}^{n} \left(\log_{10}(MC_{d,d,i}) - \log_{10}(P_{d,d,i}) \right)^2} \tag{18}$$

where d is the index corresponding with elements to be evaluated (e.g., X→1, roll→4, etc.), i indexes across the different noise levels of the run, MC is the Monte-Carlo covariance, and $\boldsymbol{P}$ is the predicted covariance. In this metric, a 1 would signify a 10x variance error and a 0.3 would signify a 2x variance error. Figure 9 shows the RMSLE of the three predictions on the 1x2x3 box. Using this view, it is easy identify the significant improvement in using either the point-to-point or point-to-plane method compared with the Jacobian method, and that the point-to-plane method is the overall best covariance estimator in this scenario.

3.2 | Simulated Aerial Approach

While the previous section demonstrated that our method provided significant advantages over the Jacobian method, the way the sensed points were generated was not very realistic. To gain a more realistic evaluation of performance, in this test, the point-to-point and point-to-plane methods are applied to a simulated aerial approach of the receiver aircraft to the tanker. The simulation includes the entire vision pipeline: Cameras in the virtual world are mounted to the underside of the tanker and collect images of the receiver aircraft model; the stereo images are passed through stereo block matching to create a sensed point cloud, and the output of the stereo block matching is then fed to the ICP algorithm. Figure 10 shows an example scenario where the tanker and receiver are illustrated, together with simulated imagery in the bottom-left and -right corner of the image. The yellow dots represent the sensed point cloud generated by this system, while the red dots are the reference point cloud.

The receiver aircraft is initially placed 25 meters away from the camera and is moved back in 1-meter increments to a distance of 45 meters. At each position, 100 trials of ICP are run on the stereo images to create a Monte-Carlo truth covariance.

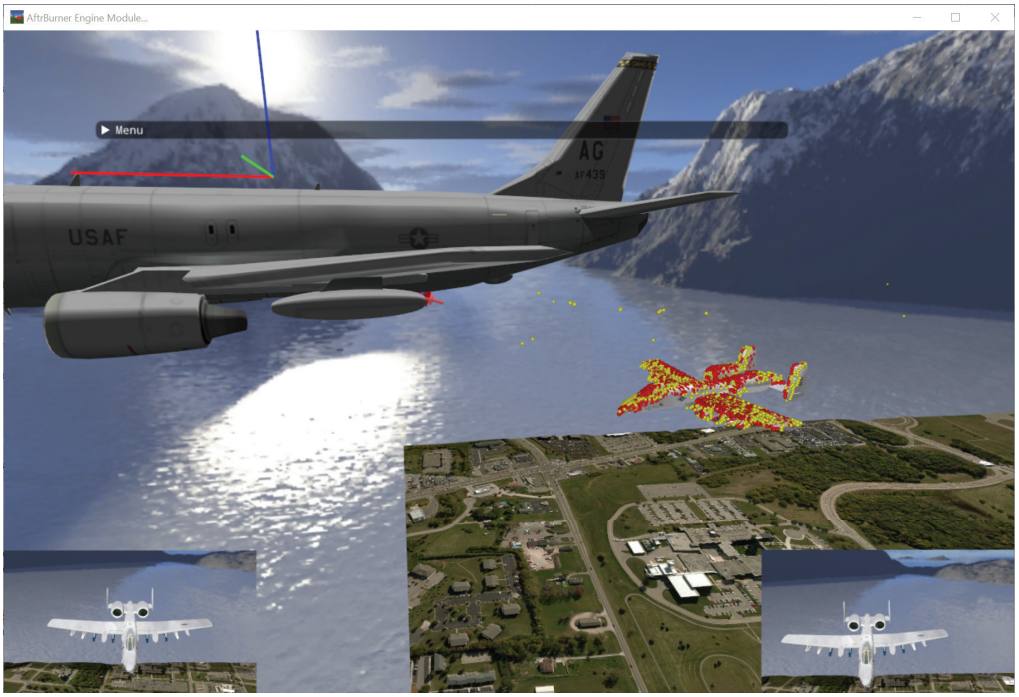

FIGURE 10 The ICP algorithm performed on simulated receiver aircraft

To inject randomness into the simulation, a secondary point light source is moved to a random location around the tanker for each trial. In this way, the point cloud generated with each Monte-Carlo run is randomized without substantially changing the simulation parameters.

The results of this experiment are shown in Figure 11, where the RMSLE of each method is plotted. Unfortunately, no single method appears to be the best across all axes at all times. However, we still believe the point-to-plane method is best overall as:

- It has no extreme errors. Note that the y-axis is on the log scale, so when the Jacobian method is off by >1.2, this corresponds to errors of more than an order of magnitude! The point-to-plane method, on the other hand, is usually less than 0.6 (a 4x variance error, or 2x standard deviation) and only exceeds this value once by a small amount.
- In areas where the point-to-plane method is worse than the alternatives, the differences are generally small (and lower on the log scale) than some of the differences in the other direction. For example, when compared with the point-to-point method, the x and yaw axes both see very significant improvements in the point-to-plane methods, while the roll and z-axis results have smaller differences between the point-to-point and point-to-plane approaches.
- As we show in the next section, the point-to-plane method has essentially identical performance results when compared with the state-of-the-art technique introduced in Prakhya et al. (2015).

3.3 | Prakhya vs our Method

In this section, Prakhya's covariance estimation method (Prakhya et al., 2015)—the most applicable method from prior literature—was compared against our Kalman filter approach. This test was performed on a 1x2x3 box as discussed in the

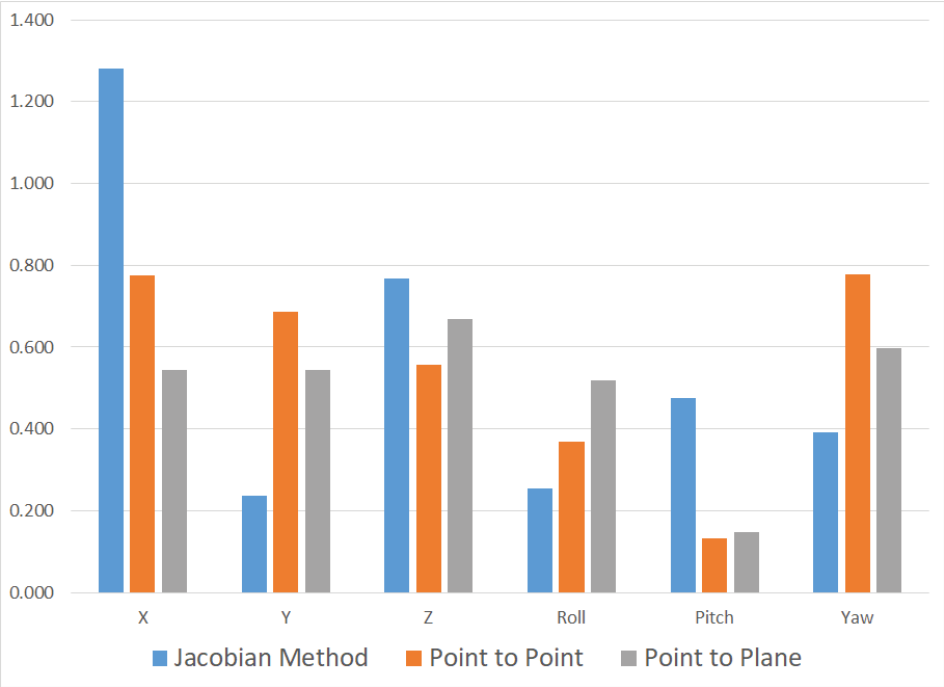

FIGURE 11 RMSLE across various distances and lighting conditions in full AAR simulation

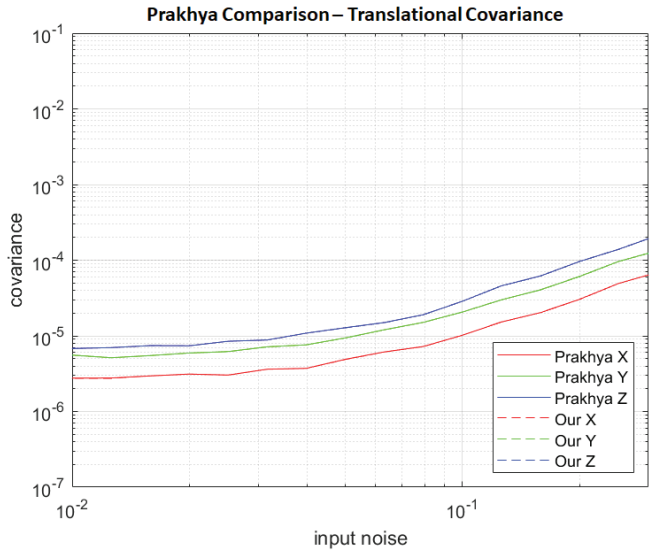

FIGURE 12 Translational comparison for Prakhya vs our method on a 1x2x3 box

prior section, and both methods were in their point-to-plane configuration. The same method for computing normal vectors and the same input noise values were used for both algorithms.

The results show that the two estimation techniques have virtually identical predictions for the translational covariance states. The rotational states are also very similar but diverge slightly at higher input noise levels.

While the accuracy estimates are very similar, computationally, the requirements for these algorithms turn out to be completely different. In Figure 14, we show the runtime required to run the Prakhya algorithm vs our approach. Both methods

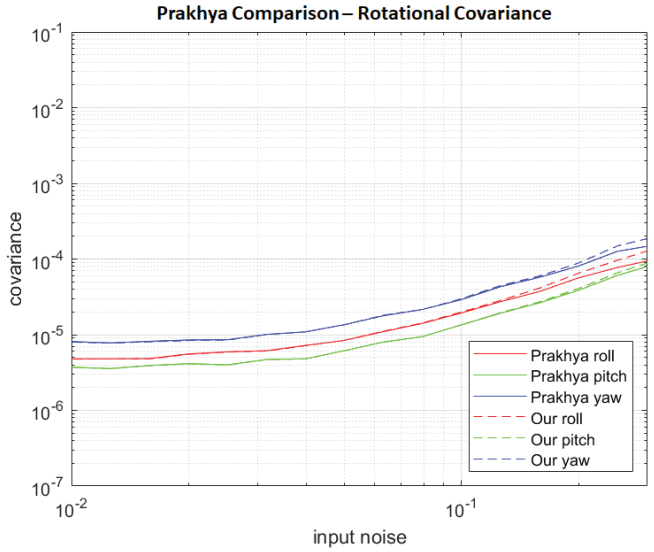

FIGURE 13 Rotational comparison for Prakhya vs our method on a 1x2x3 box

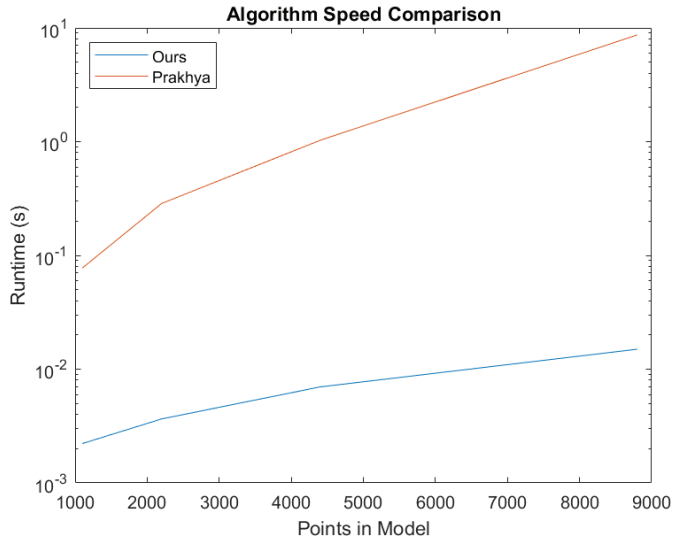

FIGURE 14 Speed comparison for Prakhya vs our method on a 1x2x3 box

were implemented in C++ code and on the same computer. As shown, the runtime of our approach is approximately two orders of magnitude faster than the Prakhya approach. This is primarily due to the operations on large matrices required to implement the Prakhya approach versus the the sequential processing of each point in our approach. Note that this time comparison is purely for the covariance estimation step. ICP point correspondence still takes the majority of the time in the overall loop.

3.3.1 | Robustness to Sensing Conditions

In addition to being more computationally efficient, the method proposed in this paper is more robust to different sensing conditions than prior work. Because the Prakhya method relies on a fixed input noise covariance matrix that is computed

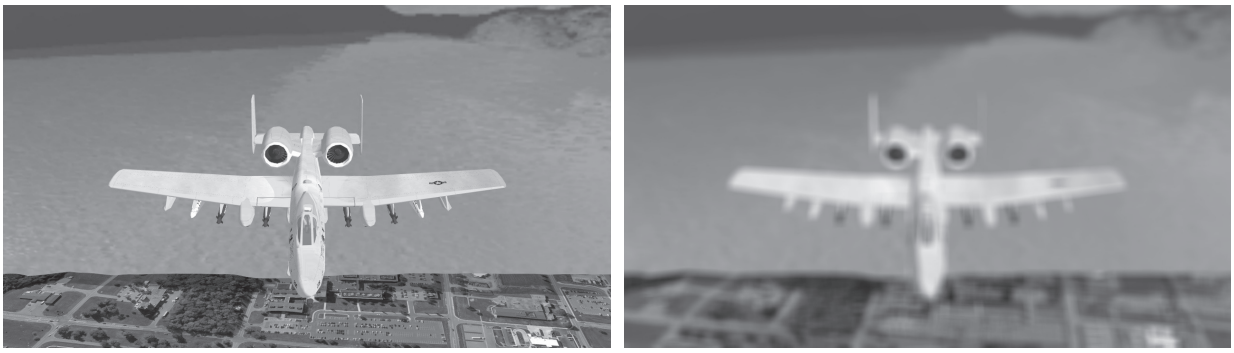

FIGURE 15 Original image vs degraded image

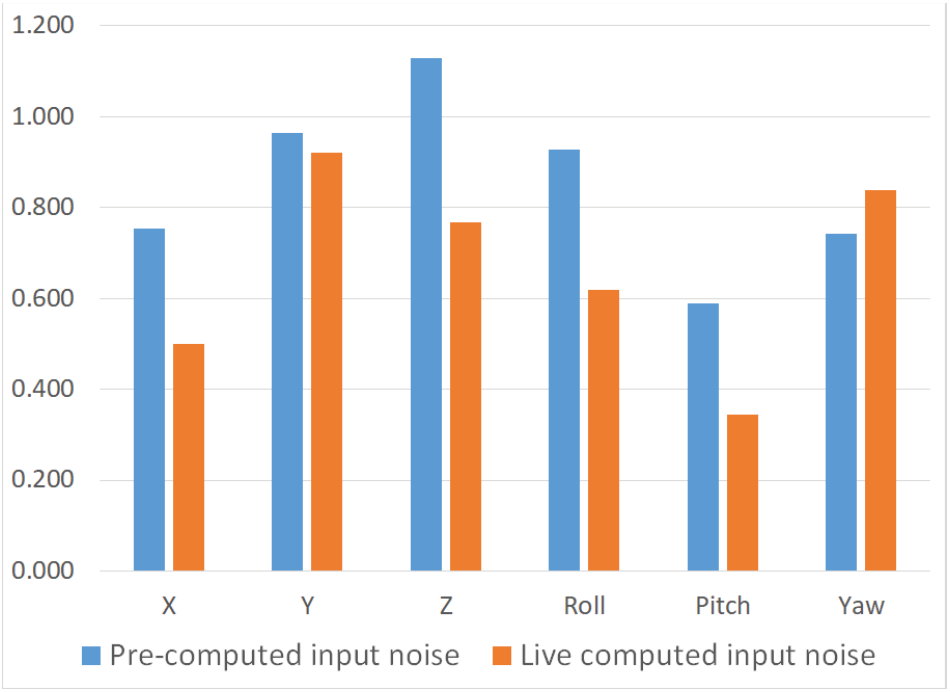

FIGURE 16 Comparative results when sensor has degraded

a priori, its accuracy may be adversely affected by changes or degradation to the sensing scenario. Our method, on the other hand, computes the input noise values online, as described in Section 2.1, making it more robust to changing sensing conditions. To demonstrate this robustness, a 21x21 box filter was applied to simulate a degraded sensor by blurring the input images. Example sensor inputs are shown in Figure 15. A real-life analogue of this effect would be fog accumulating on the camera lenses or the camera's focus adjusting slightly (e.g., due to vibration).

In Figure 16, we plot the errors from two different techniques for dealing with the blurred sensor input: (a) we utilize precomputed input noise values collected from the nominal (unblurred) AAR run, and (b) with live computation of the input noise values as described in Section 2.1. Figure 16 shows that, for the majority of the states, computing the input covariance matrix live allowed our algorithm to increase the accuracy of the output predicted covariance matrix. This test shows that computing the input covariance online increased the robustness of the covariance estimate while reducing the calibration requirement of the computer vision pose estimation system, representing a significant improvement over prior work.

4 | CONCLUSION

This paper introduces a novel method to estimate the covariance matrix of an ICP pose estimation problem. This approach uses the EKF update step to consolidate the information generated by each point in the sensed point cloud with each point being treated as a scalar measurement against its local surface normal vector. While this paper does not address some known shortcomings of prior methods, particularly the assumption that the ICP algorithm converged to the best solution, this method can accurately estimate the covariance matrix of the 6d pose more rapidly and with more intuitive appeal than prior methods.

This method was tested against Monte-Carlo simulations of a basic shape as well as on a fully simulated stereo vision application of automated aerial refueling. Results show that, of the two methods introduced, the point-to-plane method consistently is the better predictor of ICP covariance. Compared against the existing method by Prakhya et al. (2015), this approach produced near-identical covariance estimations while achieving a runtime improvement of over two orders of magnitude.

In this approach, the input sensor covariance is recreated from error statistics of the measured point cloud so that no information about the sensor setup is needed prior to computation. This creates a self-tuning effect, reducing the characterization load, and making the system robust to sensor degradation

For future work, a full sensor fusion application of automated aerial refueling can be implemented using the covariance matrix from this method. More generally, this ICP covariance estimation method can be directly applied to point clouds generated from other sources such as lidar, structured light, or 3D scanners.

REFERENCES

Anderson, J., Miller, J., Wu, X., Nykl, S., Taylor, C., & Watkinson, W. (2021, August). Real-time automated aerial refueling with stereo vision: overcoming GNSS-denied environments in or near combat areas. *Inside GNSS*. https://insidegnss.com/real-time-automated-aerial-refueling-with-stereo-vision-overcoming-gnss-denied-environments-in-or-near-combat-areas

Arun, K. S., Huang, T. S., & Blostein, S. D. (1987). Least-squares fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9*(5), 698–700. Retrieved from https://doi.org/10.1109/tpami.1987.4767965

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 14(2), 239–256. https://doi.org/10.1109/34.121791

Brossard, M., Bonnabel, S., & Barrau, A. (2020). A new approach to 3D ICP covariance estimation. *IEEE Robotics and Automation Letters, 5*(2), 744–751. https://doi.org/10.1109/lra.2020.2965391

Censi, A. (2007). An accurate closed-form estimate of ICP's covariance. *Proc. 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, 3167–3172. http://doi.org/10.1109/ROBOT.2007.363961

Censi, A., Marchionni, L., & Oriolo, G. (2008). Simultaneous maximum-likelihood calibration of odometry and sensor parameters. *2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA. https://doi.org/10.1109/ROBOT.2008.4543516

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing, 10*(3), 145–155. https://doi.org/10.1016/0262-8856(92)90066-C

Guehring, J. (2001). Reliable 3D surface acquisition, registration and validation using statistical error models. *Proc. of the 3rd International Conference on 3D Digital Imaging and Modeling,* Quebec City, Canada. https://doi.org/10.1109/IM.2001.924440

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A, 4*(4), 629–642. https://doi.org/10.1364/JOSAA.4.000629

Horn, B. K. P., Hilden, H. M., & Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A,* 5(7), 1127–1135. https://doi.org/10.1364/JOSAA.4.000629

Johnson, D. T., Nykl, S. L., & Raquet, J. F. (2017). Combining stereo vision and inertial navigation for automated aerial refueling. *Journal of Guidance, Control, and Dynamics, 40*(9), 2250–2259. https://doi.org/10.2514/1.G002648

Kolhatkar, C., & Wagle, K. (2021). Review of SLAM algorithms for indoor mobile robot with lidar and RGB-D camera technology. In M. N. Favorskaya, S. Mekhilef, R. K. Pandey, & N. Singh (Eds.), *Innovations in electrical and electronic engineering* (pp. 397–409). https://doi.org/10.1007/978-981-15-4692-1_30

Landry, D., Pomerleau, F., & Giguere, P. (2019). CELLO-3D: Estimating the covariance of ICP in the real world. *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada. https://doi.org/10.1109/icra.2019.8793516

Low, K. -L. (2004). *Linear least-squares optimization for point-to-plane ICP surface registration* (Technical Report No. TR04-004). https://www.comp.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf

Markley, F. L., & Mortari, D. (2000). Quaternion attitude estimation using vector observations. *The Journal of the Astronautical Sciences, 48*(2), 359–380. https://doi.org/10.1007/bf03546284

Mendes, E., Koch, P., & Lacroix, S. (2016). ICP-based pose-graph SLAM. *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Lausanne, Switzerland. https://doi.org/10.1109/ssrr.2016.7784298

Pomerleau, F., Colas, F., & Siegwart, R. (2015). *A review of point cloud registration algorithms for mobile robotics*. Now Foundations and Trends. https://doi.org/10.1561/9781680830255

Prakhya, S. M., Bingbing, L., Rui, Y., & Lin, W. (2015). A closed-form estimate of 3D ICP covariance. *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, Tokyo, Japan. https://doi.org/10.1109/mva.2015.7153246

Segal, A., Haehnel, D., & Thrun, S. (2009). *Generalized-ICP*. http://www.roboticsproceedings.org/rss05/p21.pdf

Sorenson, H. W. (1966). Kalman filtering techniques. *Advances in Control Systems, 3,* 219–292. Elsevier. https://www.sciencedirect.com/science/article/abs/pii/B9781483167169500102

How to cite this article: Yuan, H. R., Taylor, C. N., & Nykl, S. L. (2023). Accurate covariance estimation for pose data from iterative closest point algorithm. *NAVIGATION, 70*(2). https://doi.org/10.33012/navi.562