A Closed-form Estimate of 3D ICP Covariance

Prakhya Sai Manoj, Liu Bingbing†, Yan Rui† and Weisi Lin*
School of Computer Engineering, Nanyang Technological University, Singapore.
†Institute for Infocomm Research - A*STAR, Singapore.
saimanoj001@e.ntu.edu.sg, {bliu, ryan}@i2r.a-star.edu.sg, wslin@ntu.edu.sg

Abstract

We present a closed-form solution to estimate the covariance of the resultant transformation provided by the Iterative Closest Point (ICP) algorithm for 3D point cloud registration. We extend an existing work [1] that estimates ICP's covariance in 2D with point to plane error metric to 3D with point to point and point to plane error metrics. Moreover, we do not make any assumption on the noise present in the sensor data and have no constraints on the estimated rigid transformation. The source code of our implementation is made publicly available, which can be adapted to work for ICP with different error metrics with minor changes. Our preliminary results show that ICP's covariance is lower at a global minimum than at a local minima.

1 Introduction

Iterative Closest Point (ICP) algorithm as initially proposed in [2] has been applied to various applications ranging from scan matching, odometry estimation [3], 3D mapping in robotics to point cloud registration and reconstruction in 3D computer vision. Given two point clouds, ICP algorithm estimates the rigid transformation between them. ICP algorithm establishes correspondences between the source and the target point clouds, finds a transformation that minimizes an error metric/objective function and transforms the source point cloud based on the transformation estimate. It iterates over these steps till the estimated transformation or the residual error between the source and the target point clouds is smaller than a set threshold. In short, ICP estimates the homogeneous transformation matrix $\mathbf{H} = [\mathbf{R}|\mathbf{T}]$ consisting of a rotation matrix $\mathbf{R}$ and a translation matrix $\mathbf{T}$ that aligns the source point cloud to the target point cloud. ICP has many variants [4] that employ different error metrics such as point to point error minimization, point to plane error minimization, correspondence estimation, rejection methods or the way in which samples are weighed etc.

In [1], Andrea Censi has proposed a method to calculate ICP's covariance and showed experimentally that their method performs better than the existing methods. The actual implementation of their method is only for 2D scan matching with point to plane error metric. The author represented 2D points in a polar representation as $[r, \theta]$ and only considered the noise in r coordinate but not in θ, which is an approximation. This might hold in the case of 2D but it may not be true in 3D. Another closely related work [5] uses Mb-ICP (Metric based ICP) and estimates its covariance. However, Mb-ICP's [5] formulation cannot be extended to 3D directly, i.e., 3D Mb-ICP and its covariance estimation can only be applied to mobile robots that have negligible rotations in roll and pitch angles. Hence it cannot be generalized to estimate the transformation between two point clouds that have six degrees of freedom. Another recent work [6], provides theoretical analysis of ICP by linearizing its cost function and employing the Hessian to calculate the covariance of ICP.

In this work, we extend the ICP's covariance formulation as proposed in [1] to work for 3D point cloud registration. We do not make any assumption and consider the possibility of noise being present in all the three variables in 3D. Thus it can be applied to data acquired by devices with varying sensor noises. Moreover, our implementation is a generalized method for 3D ICP's covariance estimation without any constraints on rotation and translation.

*This research is partially supported by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

2 Covariance of a Minimization Algorithm

In this section, we discuss how to calculate the covariance of any algorithm that minimizes an objective function. The below mentioned theorem can be found in Andrea Censi's work [1], but here we provide a detailed explanation on how this equation was arrived at.

Theorem : Let $\mathbf{z}$ be the input/measurements and $\mathbf{x}$ be the output of an algorithm/function A that operates on/minimizes an objective function J i.e., $\mathbf{x} = A(\mathbf{z}) = argmin_{\mathbf{x}} J(\mathbf{z}, \mathbf{x})$. Then the approximate value of the covariance of $\mathbf{x}$ will be

$$cov(\mathbf{x}) \approx \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1} \left(\frac{\partial^2 J}{\partial \mathbf{z} \partial \mathbf{x}}\right) cov(\mathbf{z}) \left(\frac{\partial^2 J}{\partial \mathbf{z} \partial \mathbf{x}}\right)^T \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1} \quad (1)$$

Proof: $A(\mathbf{z})$ can be considered as a function with input $\mathbf{z}$ and output $\mathbf{x}$. $A(\mathbf{z})$ can be approximated using Taylor series expansion at a value $\mathbf{z} = \mathbf{z}_o$ as

$$\mathbf{x} = (A(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_o}) \approx A(\mathbf{z}_o) + \frac{\partial A}{\partial \mathbf{z}_o}(\mathbf{z} - \mathbf{z}_o)$$

$$\mathbf{x} = (A(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_o}) \approx A(\mathbf{z}_o) + \frac{\partial A}{\partial \mathbf{z}_o}(\mathbf{z}) - \frac{\partial A}{\partial \mathbf{z}_o}(\mathbf{z}_o) \quad (2)$$

where $\frac{\partial A}{\partial z_0} = \frac{\partial A}{\partial z}\big|_{z=z_0}$. The covariance of an algebraic expression of the form $\mathbf{B}\mathbf{z} + c$, where c is a constant, is equal to $\mathbf{B} cov(\mathbf{z}) \mathbf{B}^T$. Applying this property in Eqn.2, where $A(\mathbf{z}_o)$ and $\frac{\partial A}{\partial \mathbf{z}_o}(\mathbf{z}_o)$ are constants, results in

$$cov(\mathbf{x}) = cov\left(A(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_o}\right) \approx \frac{\partial A}{\partial \mathbf{z}_o} cov(\mathbf{z}) \frac{\partial A}{\partial \mathbf{z}_o}^T \quad (3)$$

Substituting the implicit function theorem (mentioned in the Appendix of [1]) as shown in Eq.4 to Eqn.3,

$$\frac{\partial A}{\partial \mathbf{z}_o} = -\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}\left(\frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}}\right) \tag{4}$$

results in the final expression as stated in the theorem.

$$cov(\mathbf{x}) = cov\left(A(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_o}\right)$$

$$\approx \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}\left(\frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}}\right) cov(\mathbf{z}) \left(\frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}}\right)^T \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}$$

In the above equation, $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}$ is a symmetric matrix(shown later), thus not taking the transpose would not make any difference. This general theorem can be applied to estimate the covariance of any algorithm that minimizes an objective function.

3 Estimating ICP's Covariance for 3D Point to Point Error Metric

Consider two point clouds P and Q and the rigid transformation between them is estimated using the ICP algorithm. The method used to establish the correspondences does not matter. We represent the number of correspondences used in the final iteration of ICP between P and Q by n. We estimate the covariance of ICP's transformation estimate that minimizes a point to point metric based objective function as shown in Eqn.5.

$$J = \textstyle\sum_{i=1}^{n} \|\mathbf{RP_i} + \mathbf{T} - \mathbf{Q_i}\|^{\mathbf{2}} \tag{5}$$

In Eqn.5, $\{\mathbf{P_i}, \mathbf{Q_i}\}$ are the n correspondences used in the last iteration of ICP and $[\mathbf{R}|\mathbf{T}]$ is the homogeneous transformation estimated by ICP algorithm. For further analysis, we will represent the homogeneous transformation $[\mathbf{R}|\mathbf{T}]$ estimated by ICP as $\mathbf{x} = [x\ y\ z\ a\ b\ c]$, where $x\ y\ z$ components represent the translation and $a\ b\ c$ represent the rotation in yaw, pitch and roll respectively. Our aim is to estimate the covariance of $\mathbf{x}$, which represents the transformation estimate returned by the ICP algorithm.

From Eqn.1, it can be seen that there is a need to calculate three terms, $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}$, $\left(\frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}}\right)$ and $cov(\mathbf{z})$, where J represents the objective function as shown in Eqn.5, $\mathbf{x} = [x\ y\ z\ a\ b\ c]$ is the ICP's transformation estimate and $\mathbf{z}$ represents the n sets of correspondences $\{\mathbf{P_i}, \mathbf{Q_i}\}$, where $\mathbf{P_i} = [P_{ix}\ P_{iy}\ P_{iz}]^T$ and $\mathbf{Q_i} = [Q_{ix}\ Q_{iy}\ Q_{iz}]^T$. We will now discuss how to calculate these three terms.

3.1 Calculation of $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)$

In 3D, $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)$ is a 6×6 matrix as shown in Eqn.6.

$$\begin{bmatrix}
\left(\frac{\partial^2 J}{\partial x^2}\right) & \left(\frac{\partial^2 J}{\partial y\partial x}\right) & \left(\frac{\partial^2 J}{\partial z\partial x}\right) & \left(\frac{\partial^2 J}{\partial a\partial x}\right) & \left(\frac{\partial^2 J}{\partial b\partial x}\right) & \left(\frac{\partial^2 J}{\partial c\partial x}\right) \\
\left(\frac{\partial^2 J}{\partial x\partial y}\right) & \left(\frac{\partial^2 J}{\partial y^2}\right) & \left(\frac{\partial^2 J}{\partial z\partial y}\right) & \left(\frac{\partial^2 J}{\partial a\partial y}\right) & \left(\frac{\partial^2 J}{\partial b\partial y}\right) & \left(\frac{\partial^2 J}{\partial c\partial y}\right) \\
\left(\frac{\partial^2 J}{\partial x\partial z}\right) & \left(\frac{\partial^2 J}{\partial y\partial z}\right) & \left(\frac{\partial^2 J}{\partial z^2}\right) & \left(\frac{\partial^2 J}{\partial a\partial z}\right) & \left(\frac{\partial^2 J}{\partial b\partial z}\right) & \left(\frac{\partial^2 J}{\partial c\partial z}\right) \\
\left(\frac{\partial^2 J}{\partial x\partial a}\right) & \left(\frac{\partial^2 J}{\partial y\partial a}\right) & \left(\frac{\partial^2 J}{\partial z\partial a}\right) & \left(\frac{\partial^2 J}{\partial a^2}\right) & \left(\frac{\partial^2 J}{\partial b\partial a}\right) & \left(\frac{\partial^2 J}{\partial c\partial a}\right) \\
\left(\frac{\partial^2 J}{\partial x\partial b}\right) & \left(\frac{\partial^2 J}{\partial y\partial b}\right) & \left(\frac{\partial^2 J}{\partial z\partial b}\right) & \left(\frac{\partial^2 J}{\partial a\partial b}\right) & \left(\frac{\partial^2 J}{\partial b^2}\right) & \left(\frac{\partial^2 J}{\partial c\partial b}\right) \\
\left(\frac{\partial^2 J}{\partial x\partial c}\right) & \left(\frac{\partial^2 J}{\partial y\partial c}\right) & \left(\frac{\partial^2 J}{\partial z\partial c}\right) & \left(\frac{\partial^2 J}{\partial a\partial c}\right) & \left(\frac{\partial^2 J}{\partial b\partial c}\right) & \left(\frac{\partial^2 J}{\partial c^2}\right)
\end{bmatrix} \tag{6}$$

Looking at the above matrix, one cannot presume that it will be a symmetric matrix. After estimating the closed form expressions for each of the terms, this matrix turned out to be a symmetric one. Now, we will see how to calculate the closed form expressions for some of the terms in $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)$.

We represent the objective function J, $J = \sum_{i=1}^{n} \|\mathbf{RP_i} + \mathbf{T} - \mathbf{Q_i}\|^{\mathbf{2}}$ as $J = \sum_{i=1}^{n} F^2$ and $F = \|G\|$, which implies that $G = \mathbf{RP_i} + \mathbf{T} - \mathbf{Q_i}$. For each set of correspondences $\{\mathbf{P_i}, \mathbf{Q_i}\}$, we estimate $\left(\frac{\partial^2 J_i}{\partial \mathbf{x}^2}\right)$ in which J_i depends on the value of i^{th} correspondences. After getting the values of $\left(\frac{\partial^2 J_i}{\partial \mathbf{x}^2}\right)$ for each set of correspondences, we sum them to form the final estimate of $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)$ i.e., $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right) = \sum_{i=1}^{n}\left(\frac{\partial^2 J_i}{\partial \mathbf{x}^2}\right)$. To calculate $\left(\frac{\partial^2 J_i}{\partial \mathbf{x}^2}\right)$, which is a 6×6 matrix, we start by computing each of its components as shown below.

$$\begin{aligned}
\frac{\partial J_i}{\partial x} &= 2 \cdot F \cdot \frac{\partial F}{\partial x} \\
\frac{\partial F}{\partial x} &= \frac{J_G(x)^T \cdot G}{\|G\|}
\end{aligned} \tag{7}$$

In Eqn.7, $J_G(x)$ represents the Jacobian of G with respect to x and it involves the calculation of the derivative of a norm, which is shown at https://sites.google.com/site/icpcovariance/appendix

$$\frac{\partial J_i}{\partial x} = 2 \cdot F \cdot \frac{J_G(x)^T \cdot G}{\|G\|} = 2 \cdot J_G(x)^T \cdot G$$

$$J_G(x) = \frac{\partial G}{\partial x} = \frac{\partial}{\partial x}(\mathbf{RP_i} + \mathbf{T} - \mathbf{Q_i})$$

$$J_G(x) = \frac{\partial}{\partial x}\left(\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33}\end{bmatrix}\begin{bmatrix} P_{ix} \\ P_{iy} \\ P_{iz}\end{bmatrix} + \begin{bmatrix} x \\ y \\ z\end{bmatrix} - \begin{bmatrix} Q_{ix} \\ Q_{iy} \\ Q_{iz}\end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0\end{bmatrix} \tag{8}$$

$$\frac{\partial J_i}{\partial x} = 2 \cdot [1 \quad 0 \quad 0] \cdot \left[[\mathbf{RP_i} + \mathbf{T}] - \begin{bmatrix} Q_{ix} \\ Q_{iy} \\ Q_{iz}\end{bmatrix}\right] \tag{9}$$

$$\frac{\partial J_i}{\partial x} = 2 \cdot (r_{11} \cdot P_{ix} + r_{12} \cdot P_{iy} + r_{13} \cdot P_{iz} + x - Q_{ix}) \tag{10}$$

Here, Eqn.10 is a closed form expression and many other terms like $\left(\frac{\partial^2 J_i}{\partial x^2}\right)$, $\left(\frac{\partial^2 J_i}{\partial y\partial x}\right)$ etc., can be calculated from this term. Moving forward, the calculation of $\left(\frac{\partial^2 J_i}{\partial a^2}\right)$ and related terms become more complex as the variables a, b, c with respect to which we partially derive, appear in the rotation matrix $\mathbf{R}$. In Eqn.11, we show the expression for $\left(\frac{\partial^2 J_i}{\partial a^2}\right)$ in terms of jacobians.

$$\left(\frac{\partial^2 J_i}{\partial a^2}\right) = 2 \cdot \left(\frac{\partial}{\partial a}\left(J_G(a)^T\right) \cdot G + J_G(a)^T \cdot \frac{\partial G}{\partial a}\right) \tag{11}$$

where $J_G(a)$ is the jacobian of G with respect to a and $J_G(a) = \frac{\partial G}{\partial a} = \frac{\partial \mathbf{R}}{\partial a}$ (as other terms are independent of a). Matrices R and $\frac{\partial \mathbf{R}}{\partial a}$ are given in the Appendix for your reference. From Eqn.11, it can be seen that the calculations become more intense and thus we employed symbolic toolbox in MatLab to calculate the jacobians. In a way, MatLab helped us to verify the values of previously mentioned terms that become constants or zeros. In this way, each term of the 6×6 matrix $\left(\frac{\partial^2 J_i}{\partial \mathbf{x}^2}\right)$ can be calculated and finally summed up to get $\left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)$.

3.2 Calculation of $\left(\frac{\partial^2 J}{\partial \mathbf{z} \partial \mathbf{x}}\right)$

To find the covariance from Eqn.1, we need to evaluate one more term $\left(\frac{\partial^2 J}{\partial \mathbf{z} \partial \mathbf{x}}\right)$ whose matrix size is $6 \times 6n$, where n is the number of correspondences $\{\mathbf{P_i}, \mathbf{Q_i}\}$ and is shown below.

$$\begin{bmatrix} \frac{\partial^2 J}{\partial P_{ix} \partial x} & \frac{\partial^2 J}{\partial P_{iy} \partial x} & \frac{\partial^2 J}{\partial P_{iz} \partial x} & \\ \frac{\partial^2 J}{\partial P_{ix} \partial y} & \frac{\partial^2 J}{\partial P_{iy} \partial y} & \frac{\partial^2 J}{\partial P_{iz} \partial y} & \\ \frac{\partial^2 J}{\partial P_{ix} \partial z} & \frac{\partial^2 J}{\partial P_{iy} \partial z} & \frac{\partial^2 J}{\partial P_{iz} \partial z} & ...(in\ terms\ of\ \mathbf{Q_i})... \\ \frac{\partial^2 J}{\partial P_{ix} \partial a} & \frac{\partial^2 J}{\partial P_{iy} \partial a} & \frac{\partial^2 J}{\partial P_{iz} \partial a} & ...for\ n\ correspondences... \\ \frac{\partial^2 J}{\partial P_{ix} \partial b} & \frac{\partial^2 J}{\partial P_{iy} \partial b} & \frac{\partial^2 J}{\partial P_{iz} \partial b} & \\ \frac{\partial^2 J}{\partial P_{ix} \partial c} & \frac{\partial^2 J}{\partial P_{iy} \partial c} & \frac{\partial^2 J}{\partial P_{iz} \partial c} & \end{bmatrix}$$

In the same way as attempted before, each term in this matrix can be calculated or estimated using MatLab software. The exact closed form expressions can be generated from the MatLab scripts provided in the source code.

3.3 Estimation of $cov(\mathbf{z})$

This is to be set in accordance with the noise present in the acquired sensor readings. It will be a matrix of size $6n \times 6n$, where n is the number of established correspondences in the last iteration. In the case where the sensor measurements $\mathbf{z}$ are uncorrelated, $cov(\mathbf{z})$ will be a diagonal matrix as shown in Eqn.12.

$$cov(\mathbf{z}) = \begin{bmatrix} cov(P_1) & 0 & \cdots & 0 \\ 0 & cov(Q_1) & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & cov(Q_n) \end{bmatrix} \quad (12)$$

4 Implementation Details

The source code of ICP's covariance for 3D point to point and point to plane error metrics is available at **https://sites.google.com/site/icpcovariance/**

We have developed a modular framework for ICP's covariance estimation. The first module is a MatLab script that deals with the calculation of the closed form expressions for the jacobians based on the objective function. The second module is a C++ package that calculates the ICP's covariance based on Eqn.1. The closed form expressions generated from the MatLab script are to be copied in to the C++ code. The C++ code takes in two point clouds, finds the transformation using ICP, and estimates the resultant transformation's covariance.

If the input point clouds are not closely aligned to each other, then there is no guarantee that ICP will converge to a global minimum. Hence, in the source code, we employ feature matching technique to bring the source and target point clouds close enough so that ICP can converge to a global minimum. Essentially, we extract uniform keypoints on both the source and target point clouds, extract SHOT [7] feature descriptors, find nearest neighbour matches of those feature descriptors and finally estimate the consensual transformation using RANSAC.

Once the source and target point clouds are aligned close enough via feature matching technique, ICP algorithm takes over for accurate point cloud registration, where the correspondences are the nearest neighbours between the 3D point clouds and the objective function is either point to point or point to plane error metric. It should be noted that only the correspondences in the last iteration are used to estimate the covariance of ICP. As the number of correspondences, n, increases, the covariance computation becomes more intensive. Hence the number of correspondences to be used for covariance computation can be limited to a certain threshold to efficiently estimate the ICP's covariance.

In order to adapt the ICP's covariance implementation to different variants, the objective function in both the MatLab script and C++ code should be changed accordingly and the calculated jacobians should be copied to the C++ package. The provided source code has both point to point and point to plane error metric based ICP's covariance estimation routines.

5 ICP's Covariance for Point to Plane Error Metric

Here, we extend the point to point error metric based ICP's covariance estimation to point to plane error metric. As mentioned before the important changes to be made are the objective functions used in both the C++ and MatLab scripts and the Jacobians estimated from the MatLab scripts are to be loaded into the C++ script. The objective function employed for point to plane [8] error metric is shown below.

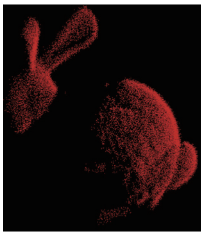
$$J = \sum_{i=1}^{n} [(\mathbf{RP_i} + \mathbf{T} - \mathbf{Q_i}).\mathbf{n_i}]^{\mathbf{2}} \quad (13)$$

where $\mathbf{n_i}$ is the unit surface normal vector at $\mathbf{Q_i}$ and all other variables mean the same as in Eqn.5. Please refer to the source code for the closed-form expressions.

6 Experimental Results

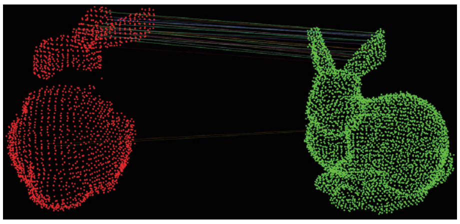
Our preliminary experimental evaluation shows that the ICP's covariance is lower at the global minimum than at a local minima. This may highlight the possibility of exploiting ICP's covariance in guiding the ICP to converge to the global minimum by branching from a local minima. To quantitatively show the covariance of ICP at a local minima and the global minimum, consider the data and the model point clouds as shown in Fig. 1(a,b), where the partial data point cloud has to be registered to a complete model point cloud. These data and model point clouds are also provided with the source code.

As mentioned before, to closely align the data and model point clouds, we employ the feature matching technique. We extract uniform keypoints on both the data and model point clouds and match them via SHOT feature descriptors [7]. Later, the false feature correspondences are removed via RANSAC and the final consensual transformation is estimated. The established positive correspondences are visualized in Fig. 1(c). The data point cloud is transformed based on this estimated transformation and ICP is executed on these two closely aligned point clouds. The resultant point cloud after successful registration of data and

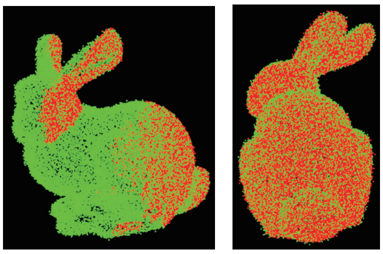


(a) Data

(b) Model



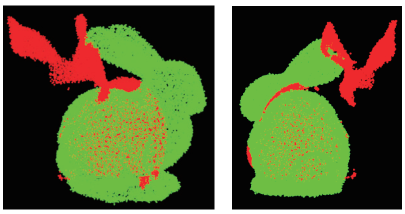
(c) Resultant Keypoint correspondences after feature matching technique



(d) ICP converging to a global minimum

$$\begin{bmatrix} 1.7e-07 & 1.4e-08 & -3.3e-08 & -5.8e-08 & 5.9e-09 & -9.2e-08 \\ 1.5e-08 & 2.1e-07 & 3.8e-09 & -1.1e-07 & 3.6e-08 & -1.0e-09 \\ -3.1e-08 & 4.0e-09 & 2.2e-07 & -7.4e-09 & 1.6e-07 & 5.9e-08 \\ -3.6e-08 & -1.1e-07 & -1.0e-08 & 5.8e-07 & -3.1e-07 & 3.2e-07 \\ -4.0e-08 & 3.5e-08 & 1.7e-07 & -3.0e-07 & 1.1e-06 & -6.0e-07 \\ -7.9e-09 & 1.5e-09 & 4.7e-08 & 3.0e-07 & -6.2e-07 & 1.0e-06 \end{bmatrix}$$

(e) ICP's covariance at the global minimum for the case (d)



(f) ICP converges to a local minima on the given data and model point clouds when the feature matching technique is not employed.

$$\begin{bmatrix} 3.7e-05 & 1.1e-05 & 1.2e-05 & -1.8e-05 & -5.0e-05 & 4.8e-05 \\ -3.6e-06 & 2.0e-05 & 2.5e-06 & 8.9e-06 & 4.2e-06 & 1.9e-05 \\ 3.4e-06 & 4.9e-06 & 1.4e-05 & -1.4e-06 & -4.5e-06 & 1.9e-05 \\ -1.4e-05 & 3.8e-06 & -3.8e-06 & 0.00011 & -1.1e-05 & -2.1e-05 \\ -5.5e-05 & -1.7e-05 & -1.8e-05 & -5.3e-06 & 0.00015 & -5.6e-05 \\ -5.8e-06 & 2.1e-05 & 1.2e-05 & -4.0e-06 & 2.1e-05 & 8.4e-05 \end{bmatrix}$$

(g) ICP's covariance at a local minima for the case (f)

Figure 1: Visualizing the local and global convergence and covariance of ICP algorithm on given data and model point clouds with point to plane error metric.

model point clouds via point to plane ICP variant is shown in Fig. 1(d).

However, without the feature matching technique, ICP does not converge to a global minimum on the provided data and model point clouds as there is a large orientation difference between them. Hence, in this case, where feature matching is not employed, ICP converges to a local minima, in other words, ICP fails and the registered point cloud is shown in Fig. 1(f). Feature matching is only necessary when there is large orientation difference between the data and model point clouds, otherwise ICP can possibly converge to global minimum.

In Fig. 1(e), we show the ICP covariance matrix at the global minimum and in Fig. 1(g), the ICP covariance matrix at a local minima for point to plane error metric based ICP is shown. It can be noticed from Fig. 1(e) and Fig. 1(g) that the ICP's covariance is higher at a local minima that at the global minimum.

7 Conclusion and Future Work

We discussed an existing formulation [1] for ICP's covariance estimation and extended it from 2D to 3D without making any assumptions. The source code of the implementation for ICP's covariance estimation for 3D point to point and point to plane error metric is made publicly available. One important point to note is that only the correspondences in the final iteration are used for covariance estimation in the provided implementation and also in [1]. In short, there is no information that is being considered on how the ICP algorithm has reached the final step. This can be a key direction to explore in the future. Our preliminary results show that ICP's covariance is lower at a global minima as compared to local minima but there is a need to perform extensive evaluation to confirm this. This highlights a possibility of exploring if the ICP's covariance can be used to guide the ICP algorithm to the global minimum from a local minima.

References

[1] A. Censi, "An Accurate Closed-form Estimate of ICP's Covariance," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2007.

[2] P. J. Besl and N. D. McKay, "Method for Registration of 3-D shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992.

[3] S. M. Prakhya, B. Liu, W. Lin, and U. Qayyum, "Sparse Depth Odometry : 3D Keypoint based Pose Estimation from Dense Depth Data ," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015.

[4] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *Third IEEE International Conference on 3-D Digital Imaging and Modeling*, 2001.

[5] S. Steinmann and F. Pomerleau, "Analysis of Scan Matching Methods for Indoor 3D Mapping."

[6] M. Barczyk, S. Bonnabel, and F. Goulette, "Observability, Covariance and Uncertainty of ICP Scan Matching," *arXiv preprint arXiv:1410.7632*, 2014.

[7] S. Salti, F. Tombari, and L. Di Stefano, "SHOT: Unique Signatures of Histograms for Surface and Texture Description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[8] K.-L. Low, "Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration," *Chapel Hill, University of North Carolina*, 2004.