IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 September 2025, accepted 26 September 2025, date of publication 3 October 2025, date of current version 9 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3617328

RESEARCH ARTICLE

Vision-Based Relative Pose Estimation of Non-Cooperative Spacecraft Using Time-of-Flight Camera

SANGDO PARK, MINSIK OH, HEOKJUNE YOU, AND DONGWON JUNG, (Member, IEEE)
Department of Smart Drone Convergence, Korea Aerospace University, Goyang-si 10540, South Korea

Corresponding author: Dongwon Jung (djung@kau.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by Korea Government [Ministry of Science and ICT (MSIT)], in 2024 (Space Debris Capture Mechanism Design and Validation Using Lab-Scale Experimental Platform), under Project NRF-2022M1A3B8073175.

ABSTRACT This paper deals with a vision-based pose estimation of a target using a time-of-flight (ToF) camera. In particular, a relative pose estimation algorithm for non-cooperative targets that lack markers or accessible sensor is proposed. The proposed method, called the CSAC-ICP algorithm, combines a feature-based coarse registration method, known as Corner Random Sample Consensus (CSAC), with the well-known Iterative Closest Point (ICP) algorithm to ensure both accuracy and robustness. Here, 3D corner points are adopted as primary features. The corner points are obtained by first converting 3D point clouds into 2D images from which line features are extracted and then re-projected into 3D lines, enabling 3D corner points computation. This reduces the computation on a large number of 3D points, ensuring real-time performance. Furthermore, an extended Kalman filter (EKF) is integrated with the proposed pose estimation algorithm in a loosely coupled manner, not only to continuously provide the target pose information for a moving target, but also to reduce measurement noises due to the motion. Because feature-based coarse registration takes advantage of the global optimization strategy, the proposed method can avoid the inherent pitfall of the standalone ICP method, in which the optimization process may fall into a local minimum and lead to incorrect pose calculation under certain conditions. Moreover, the proposed pose estimation scheme minimizes the dependency on the prior knowledge about the target object's shape by taking advantage of the RANSAC-based registration. Numerical simulation is set up to validate the functionality of the algorithm using simulated 3D point clouds, confirming the accuracy performance under controlled target motion. In addition, the proposed algorithm is implemented and tested on an actual hardware platform, which demonstrates the feasibility of real-time relative pose estimation for non-cooperative targets within the space environment. Moreover, the proposed pose estimation scheme reduces reliance on prior knowledge of the target object's geometry by using RANSAC-based registration. The simulation results show that the proposed CSAC-ICP algorithm achieves 30% faster computation and 50% higher accuracy compared to the standalone ICP method. Experimental validation further confirms that 3D corner features can be extracted at a rate of 30 frames per second. In addition, the pose estimation achieves an accuracy of up to 3 degrees for rotational motion and within 0.02 meters for translational motion along all three axes, demonstrating the feasibility and superior performance of the proposed algorithm in real-world scenarios.

INDEX TERMS Relative pose estimation, time-of-flight camera, multiplicative extended Kalman filter, iterative closest point.

The associate editor coordinating the review of this manuscript and approving it for publication was Patrizia Livreri.

I. INTRODUCTION

On-orbit servicing (OOS) includes Autonomous Rendezvous and Docking (AR&D), proximity operations of multiple spacecraft, and Active Debris Removal (ADR). For these operations, accurate acquisition of relative pose information between a service (chaser) spacecraft and the targets is crucial for the success of the missions [1].

In conventional AR&D missions, relative pose estimation was facilitated with cooperative targets. A target is classified as cooperative if its geometry or dynamic information is known, or if it is equipped with devices that can provide direct/indirect pose information to the chaser spacecraft. In the ETS-VII mission, which was the first AR&D procedure [2], the chaser and the target exchanged information from the Global Positioning System (GPS) through a communication link to determine the relative position. Meanwhile, the target was also equipped with camera-type sensors and recognizable markers to facilitate the measurement of its relative attitude by the chaser. Similarly, the Demonstration of Autonomous Rendezvous Technology (DART) mission also applied GPS communication between the chaser and the target, with the target featuring retroreflectors specifically designed to enhance the performance of the Advanced Vision Guidance System (AVGS) on the chaser [3]. For the Orbital Express mission and the European Space Agency's ATV-1 mission, inter-satellite communication and IR markers were used to accurately estimate the relative pose between cooperative targets and chasers [4], [5]. The success of AR&D missions with cooperative targets looks promising. However, in a real world mission environment, there could be cases where the service spacecraft operates on non-cooperative targets, which faces technical challenges. This is because non-cooperative objects typically lack prior knowledge about the target's geometry, and moreover, they are inactive and unlikely to be equipped with active systems capable of providing its motion states to the service spacecraft [6].

For this reason, relative pose estimation becomes an indispensable technology for successful on-orbit servicing missions. And many researchers have addressed this challenge using vision-based navigation approaches [7]. In particular, passive cameras have been widely adopted for relative pose estimation due to advances in data processing techniques, which facilitate ease of use and rapid image processing [8], [9], [10], [11], [12], [13], [14], [15]. In this regard, the authors of [8] have proposed a pose tracking algorithm that applies an M-estimator to images from a monocular camera. The initial pose estimate was obtained by matching the features extracted using the Scale-Invariant Feature Transform (SIFT) [16] with those of a target model. Unlike SIFT features, other feature types experience a decline in quality caused by variations in lighting conditions, target geometry changes, which reduces the effectiveness of feature matching. To address this issue, a convex hull [17] was applied to 2D images to extract convexity defect features. These features were then matched with a known 3D model of the target to infer 2D-3D correspondences for pose initialization. In contrast, a stereo camera can provide direct depth information eliminating scale ambiguities of the target object and enable a more accurate and robust pose estimation [10]. In [18], Features from the Accelerated Segment Test (FAST) are adopted as the main feature, subsequently employed to determine the relative pose using epipolar geometry. However, since stereo cameras calculate a depth map based on the principle of triangulation, the accuracy of pose estimation using stereo cameras is highly dependent on precise calibration to determine the camera baseline [19]. The authors of [11] applied a Visual Inertial Odometry (VIO) technique by using LED markers as features and performing feature matching. To enhance the reliability of target center estimation, they computed the average of all possible combinations of four markers selected from a total of six. In [12], the authors adopted a visual-Simultaneous Localization and Mapping (SLAM) approach. To address the difficulty of feature extraction in low-texture scenes captured by a monocular camera, they incorporated the Line Segment Detector (LSD) [20] and hidden parameters, achieving a 63.6% performance improvement over conventional ORB-SLAM [21]. The studies in [13] and [14] assumed the use of a monocular camera to observe a Positive Sensitive Detector (PSD) and generated measurements based on line-of-sight vectors. In [14], the authors employed a Dual Quaternion Multiplicative Extended Kalman Filter (DQ-MEKF) to achieve tightly coupled position and attitude estimation, effectively mitigating drift errors that occur when the sensor origin does not coincide with the target's Center of Mass (CoM). General Rodrigues parameters (GRP) are known for the minimal representation of the attitude of a rigid body with three angles, similar to Euler angles [22]. Unlike redundant representations such as quaternion or rotation matrix, GRP can represent attitude with minimal number of parameters while providing an effective way of avoiding singularities, such as gimbal-lock phenomenon of Euler angles. From this perspective, Generalized Rodrigues Parameters (GRP) were used to represent the local error-quaternion between the target and chaser, from which a multiplicative quaternion-error approach was derived. This formulation ensures the maintenance of the quaternion unit-norm constraint within the Unscented Kalman Filter (UKF) framework [13],

Active vision sensors, including Time-of-Flight (ToF) and Light Detection and Ranging (LiDAR), have the advantage of rapid acquisition of 3D depth information [23], as these sensors operate by projecting infrared (IR) light onto objects, measuring the time of flight, and converting it into distance data, removing the necessity of implementing the stereo algorithm to obtain depth information. Due to these advantages, there has been a rapid increase in research using active vision sensors to estimate the relative pose of non-cooperative targets [24]. Estimating relative pose using 3D point clouds from the ToF sensor involves a successful registration of consecutive frames, with the iterative closest point (ICP) being one of the most widely used algorithms [25]. The goal of this algorithm is to derive the optimal transformation matrix, $T(R, \boldsymbol{t})$, by minimizing the distance between two point sets through an iterative procedure. Throughout this procedure, an appropriate initial guess is required to find the optimal transformation within the minimum number of iterations. To this end, the integration of the ICP with a Kalman Filter (KF) in a closed-loop configuration has been proposed. In this setup, the posterior states of the KF serve as the initial coarse alignment for the ICP algorithm during the next time step, removing the ambiguity in pose estimation for symmetric objects [6], [26]. The authors in [26] also proposed the integration of the ICP with the KF, specifically applying the DQ-MEKF to estimate the pose of the chaser spacecraft. Specifically, the authors in [6] designed a filter applying a simplified model of orbital relative motion, the Clohessy-Wiltshire (CW) equation [27], to propagate the relative position of a target. They also estimated the relative attitude by incorporating the center of mass position and moment of inertia information based on a known CAD model. Even with this ICP-KF configuration, registration errors may still occur, which is due to stability issues of the KF caused by sporadic measurements from ICP, hence the accuracy of pose estimation is significantly influenced.

An alternative approach to provide an initial guess to the ICP algorithm is to employ a two-stage registration method, that is, a coarse registration from matching features of the point clouds is performed first, which delivers a proper initial guess to the ICP registration to improve the accuracy of pose tracking [28], [29], [30], [31]. In contrast to the ICP-KF combination, both registration algorithms utilize the sampled ToF data at each step independently of the previous steps. Consequently, the accumulation of errors caused by the local minimum of the ICP algorithm can be avoided. However, the computation of local features using point clouds is based on the geometric relationship between a query point and its neighborhood. This results in an increased computational load proportional to the size of the data, making it less efficient for the on-board application. In order to reduce the complexity of local feature computation, [31] has proposed to use sparse sampling of point clouds for feature detection. However, this approach has the limitation of potentially losing geometric information before supplying the initial alignment to the ICP.

This paper presents a new approach for relative pose estimation of a non-cooperative target using point clouds from a ToF camera. To this end, a feature-based coarse registration method, named Corner Random Sample Consensus (CSAC), is combined with the well-known ICP algorithm to provide dependable initial guesses, hence ensuring both accuracy and robustness. Since this approach avoids computing on a large number of point clouds for full feature registration, but efficiently extracting 3D corner features to be used in the coarse registration, subsequently, it fits for real-time applications. Furthermore, to account for the dynamic motion of the target and mitigate the effects of measurement noise, an EKF is integrated with the proposed pose estimation algorithm in a loosely coupled manner, not only to continuously provide the target pose information for a moving target, but also to reduce measurement noises due to the motion.

The main contributions of the paper include:

1) The proposed method improves 3D corner feature computation by projecting 3D point clouds into 2D images for efficient line feature extraction and reprojecting them as 3D lines, thereby reducing computational cost and enabling real-time performance.
2) For pose determination, CSAC has been developed, employing a feature-based coarse registration. This provides reliable initial guesses for ICP, hence enhancing the accuracy of pose estimation.
3) The framework for pose estimation is proposed by integrating pose tracking method along with an extended Kalman filter.

The paper is organized as follows. Section II provides a detailed explanation of the proposed method, while Sect. III presents theoretical results under controlled conditions through numerical simulations. Section IV-B describes the experimental setup and performance verification results of the proposed method. Finally, Section V concludes the paper.

II. PROPOSED METHOD

The pose estimation framework using point clouds from a ToF camera consists of two stages: feature extraction and pose estimation, as illustrated in Figure 1.

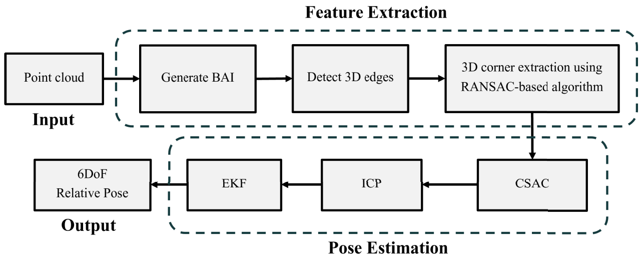


FIGURE 1. Diagram of the proposed method for pose estimation of non-cooperative target.

In the first stage, 3D corner features are efficiently extracted using 2D-3D relationship, which are then used in the second stage for pose estimation. The pose estimation stage integrates a two-stage registration process with an EKF, consisting of coarse registration using CSAC and refined registration using ICP.

A. FEATURE POINTS EXTRACTION

In general, various types of feature, such as corners, edges, and textures, can be adopted to characterize 2D images; however, extracting 3D features from 3D point clouds heavily relies on geometric analysis since it only contains distance information related to the shape of the object. Applying this analysis to every point in the data set requires enormous computational resources and is not suitable for real-time applications. To address this challenge, this paper proposes an efficient 3D feature extraction method. The proposed feature extraction method focuses on extracting global features to avoid intensive geometric analysis on the frames of point clouds.

1) 3D EDGE

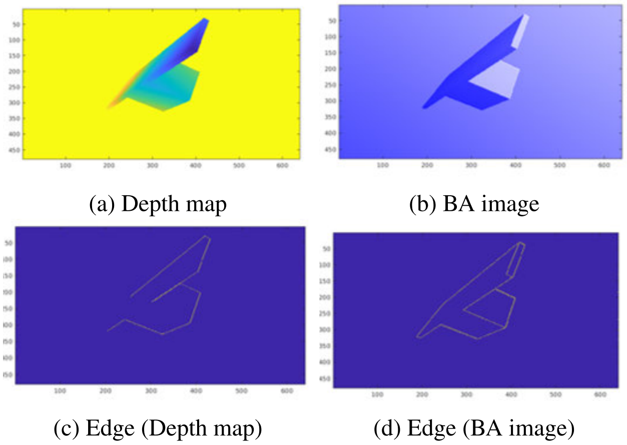
Feature extraction using 3D point clouds requires to analyze the spatial relationships between multiple query points and their neighboring points, which involves substantial computational overhead [29], [32] including the creation of index structures and the processes of pixelwise Euclidean distance calculations. Due to this computational overhead, it may be unsuitable for real-time applications to use 3D point clouds for feature extraction. Rather, converting 3D point clouds into a structured 2D representation is regarded as an efficient solution, as this can maintain predictable computational overhead regardless of whether the point clouds are structured or unstructured. A commonly adopted technique for converting point clouds into 2D images involves generating depth maps by which each pixel represents the distance information from the camera focal point, as shown in Fig. 2.

Various feature extraction operators for 2D images have already been proposed [33], [34], [35] and are widely used due to their low computational cost and proven effectiveness. These features are determined by significant differences in adjacent pixel values, which clearly visible along object boundaries separated from the background. But in some cases, they may not be as distinct within the object. Specifically, when pixel values represent distance information, as in the case of a depth map, the inner edges of an object may not appear distinctly since the difference variation across the edges could be negligible, as shown in Fig. 2c. To address this limitation, the frames of point clouds were rather transformed into Bearing Angle Images (BAI) [30] where, pixel values encode the geometric characteristics between neighboring points instead of distance information. That is, the BAI computes the Bearing Angle (BA) between the line segment connecting two consecutive points and the trace line originating from the camera, converting this angle into pixel values, as in Eq. (1), thereby enhancing the detectable variations for feature extraction.

$$BA_i = \cos^{-1} \frac{\rho_i - \rho_{i-1} \cos \Delta\psi}{\sqrt{\rho_i^2 - \rho_{i-1}^2 - 2\rho_i \rho_{i-1} \cos \Delta\psi}} \tag{1}$$

where the symbols ρ_i and ρ_{i-1} represents depth values of adjacent pixel along a selected scan direction while the symbol $\Delta\psi$ denotes the angle increment of trace lines with respect to the camera, each corresponding to the adjacent pixels. As depicted in Fig. 2b, this transformation can emphasize the intensity variations in feature regions. It follows that the comparison of applying the Canny operator to the depth map and the BAI reveals that the internal edges are more prominently highlighted, as shown in Fig. 2d.

In this study, 3D corner points are chosen as the desired feature points, by which the distinctive geometric properties of the object, or pose of the object can be easily recovered. Mathematically, a point can be calculated as the intersection of lines. For this sake, the set of edge pixels obtained from the Canny operator applied to the BAI should be first converted into mathematical lines. Extracting line parameters can be efficiently achieved using the Hough transform from 2D edge pixels, but it typically results in a 2D line representation. Instead, in this study, a 2D-3D reprojection is employed to converted edge pixels into 3D edge pixels, followed by the extraction of 3D lines through the manipulation of these 3D edges. This allows preserving the full geometric diversity of the 3D point clouds so that it can enhance the accuracy of 3D line extraction. In other words, projecting individual 2D edge pixels into 3D space generates a dense set of 3D edge pixels that fully capture variations in depth and orientation. This enables robust 3D line extraction by leveraging the distribution of numerous 3D points, enabling more accurate geometric analysis while minimizing information loss caused by linear simplification in 2D space.



(a) Depth map

(b) BA image

(c) Edge (Depth map)

(d) Edge (BA image)

FIGURE 2. Edge detection comparison from images depth and BAI using Canny.

To this end, suppose the depth information associated with a given edge pixel is known, a 3D edge pixel location $\boldsymbol{P} = (P_x, P_y, P_z)$ can be calculated by using the Pinhole camera model as illustrated in Fig. 3. The pinhole camera model is expressed in Eq. (2),

The transformation relationship can be expressed as follows:

$$\begin{bmatrix} u \\ v \\ 1 \\ \frac{1}{z} \end{bmatrix} = \frac{1}{z} \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \tag{2}$$

where u and v represent the pixel coordinates on the image plane, f denotes the focal length, and c_x and c_y are the principal points on the I_x and I_y axes, respectively. The reprojected 3D edge pixels using Eq. (2) is depicted in Figure 4. This approach is more efficient than the geometric analysis method described in [36], since it is less affected by the structure and size of the point cloud.

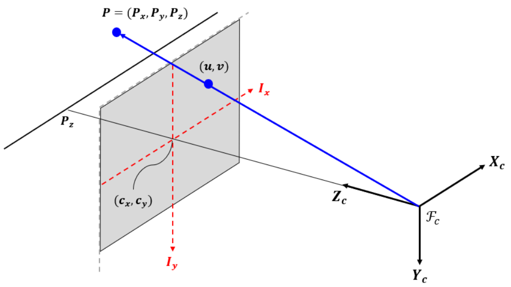


FIGURE 3. Relationship between depth and point clouds aspect on camera model.

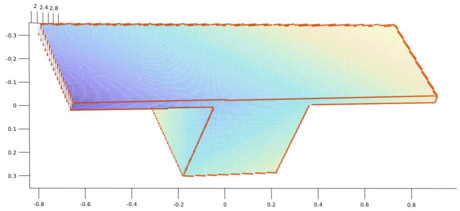


FIGURE 4. Re-mapped edge point clouds obtained from BAI.

2) CORNER EXTRACTION

To identify 3D corners, it is essential to determine line parameters to locate their intersection points. One common method for extracting line parameters is the Hough Transform (HT), which maps spatial data into a parameter space and determines line parameters through a voting scheme. However, this approach lacks robustness to noise and outliers, making the extraction of reliable information challenging [37]. In contrast, Random Sample Consensus (RANSAC)-based methods are inherently robust against noise and outliers and can be applied to various geometric models. RANSAC operates by formulating hypotheses and validating them through random sampling, making it particularly effective in handling noisy data and outliers. Therefore, an efficient RANSAC-based line segmentation algorithm was developed, and its details are presented in Algorithm 1. This algorithm works as follows: Sampled two points randomly on 3D edge (PC_e) to define a candidate line model. The algorithm then evaluates the fitness of hypothesis by calculating the distances of all remaining points in PC_e to the line and classifies points within a predefined threshold as inliers. The model with the highest number of inliers is stored as the best fit for each iteration of Algorithm 1. At the end of each iteration, the inliers are removed from PC_e to avoid redundant calculations in subsequent iterations. This step significantly reduces computational overhead, enhancing the efficiency of RANSAC-based methods and enabling their application in real-time systems [38]. The lines obtained using the line parameters derived from the above algorithm are shown in Figure 5 and can be expressed as follows:

$$l_n = [\boldsymbol{d}_n^T, \boldsymbol{b}_n^T]^T \tag{3}$$

where, $\boldsymbol{d} \in \mathbb{R}^{3\times1}$ is a direction vector and $\boldsymbol{b} \in \mathbb{R}^{3\times1}$ represents a point that belongs to the 3D edge and the subscript n is the number of lines found using Algorithm 1.

Algorithm 1 Line Parameter Identification Using RANSAC

```
Input: 3D edge ($PC_e$)
Output: Parameters of the best fit 3D line
while length($PC_e$) > 0 do
    Randomly sample 2 points from $PC_e$
    Compute the line defined by these points
    Calculate the distances of all other points in $PC_e$ to the line
    if distance $\leq \epsilon_{line}$ then
        Classify as inliers
    end if
    Store the model with the highest number of inliers
    Remove inliers from $PC_e$
end while
return Parameters of lines
```

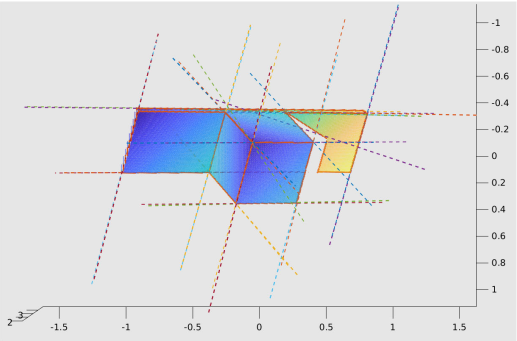


FIGURE 5. Segmented lines using Algorithm 1.

To determine the 3D corner, which is the intersection of ℓ_i and $\ell_j : (i \neq j)$, the following linear system is adopted:

$$[\boldsymbol{d}_i, \ -\boldsymbol{d}_j]\boldsymbol{c} = \boldsymbol{b}_j - \boldsymbol{b}_i. \tag{4}$$

Here, $\boldsymbol{c}$, the solution of Eq. (4), consists of scalar coefficients used to compute the Euclidean coordinates of the candidate 3D corner points, $\boldsymbol{c}_{ck}$ ($k = 1$ or 2), which can be obtained using the following equation:

$$\begin{aligned} \boldsymbol{c}_{c1} &= \boldsymbol{b}_i + \boldsymbol{c}(1)\boldsymbol{d}_i, \\ \boldsymbol{c}_{c2} &= \boldsymbol{b}_j + \boldsymbol{c}(2)\boldsymbol{d}_j. \end{aligned} \tag{5}$$

It is important to note that the intersections of the lines computed using Eq.(5) may be located not only within the target but also outside its boundaries, as depicted in Figure 6. Furthermore, in cases where the two lines do not intersect, $\boldsymbol{c}_{c1}$ and $\boldsymbol{c}_{c2}$ will yield distinct values. Therefore, for a 3D corner candidate to be formally classified as a 3D corner, it must satisfy two necessary conditions.

1) To verify whether the two lines intersect, the similarity between 3D corner candidates is evaluated using the predefined threshold ϵ_c as follows:

$$\|\boldsymbol{c}_{c1} - \boldsymbol{c}_{c2}\| < \epsilon_c. \tag{6}$$

2) To ensure that a 3D corner candidate belongs to a 3D edge, the 10 nearest neighboring points within the 3D

edge that are closest to a definitive 3D corner candidate are selected. Next, the average Euclidean distance d_{nn} between the 3D corner candidate and these neighboring points is computed and compared to the threshold ϵ_{nn}. This relationship is expressed by the following equation:

$$d_{nn} < \epsilon_{nn}. \tag{7}$$

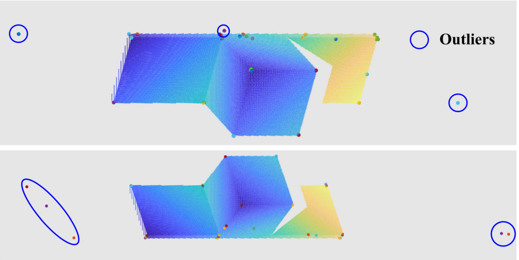


FIGURE 6. Extracted corner including candidates.

Finally, 3D corner candidates that is passed those two conditions are determined as 3D corner.

B. POSE ESTIMATION

The pose estimation process based on 3D data involves a registration procedure that estimates the transformation between consecutive point clouds. Typically, the ICP algorithm is widely used for registration; however, its accuracy and computational efficiency heavily depend on the initial guess. To address this limitation, this study employs a two-stage registration approach, consisting of coarse registration followed by refined registration, to enhance the robustness of the alignment. To achieve this, CSAC was developed by combining the 3D corners extracted in Section II-A with the RANSAC algorithm. CSAC aims to improve registration performance by providing an appropriate initial transformation for ICP. In particular, to enhance registration accuracy and improve computational efficiency, the method was designed to converge within a maximum of three iterations. This approach is feasible because, as demonstrated in Section IV-B, the extraction speed of 3D corners reaches 30 Hz, ensuring rapid processing. Furthermore, to implement a relative navigation filter, the 6-DoF relative incremental pose estimated by CSAC-ICP was integrated into an Extended Kalman Filter-based to configure the pose estimation scheme.

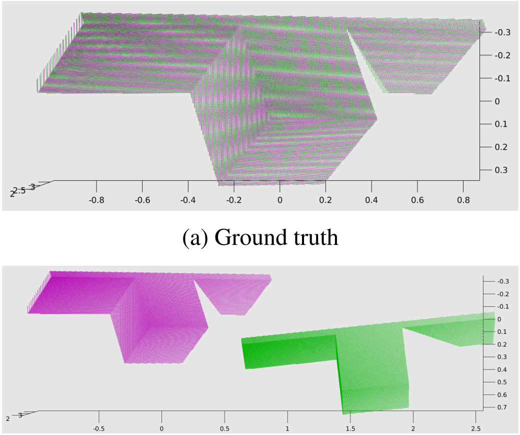
1) ICP

This section provides an overview of the fundamental principles of the ICP algorithm. It is one of the most widely used methods for registration and is designed to estimate the rigid transformation between two 3D point clouds. Given two point clouds obtained from different viewpoints—a target point cloud $\mathcal{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n]$ and a source point cloud $\mathcal{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$, where $\boldsymbol{a}_i,\ \boldsymbol{b}_i \in \mathbb{R}^{3\times1}$— ICP first establishes correspondences by identifying the Nearest Neighbors (NN) between the points in the two datasets. Once correspondences are established, ICP performs an optimization process to minimize the total distance between corresponding points, which is formulated as follows:

$$T_{ICP}(R,\ \boldsymbol{t}) = \sum_{i=1}^{N} \|R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}_i\|^2 \tag{8}$$

where R represents the rotation matrix, and $\boldsymbol{t}$ is the translation vector.

This process continues until the Mean Square Error (MSE) falls below a predefined threshold, ensuring an accurate transformation T_{ICP}. Despite its effectiveness, ICP has several critical limitations when used as a standalone registration method. One of the primary limitations of ICP is its strong dependence on the initial alignment, as its performance is highly sensitive to the initial guess.



(a) Ground truth

(b) ICP local convergence

FIGURE 7. Result of registration using ICP. Magenta: Target state, Green: Estimated transformation.

If the initial guess is less accurate, the ICP may converge to a local minimum, leading to a transformation that deviates significantly from the optimal registration result, as illustrated in Figure 7b. Additionally, ICP often requires a large number of iterations, which significantly increases computational overhead, making it unsuitable for real-time applications crucial for relative navigation. To mitigate these limitations, this study introduces a coarse registration technique that provides an appropriate initial alignment for ICP. However, since ICP itself is computationally demanding, the efficiency of the coarse registration process must also be carefully considered.

2) CSAC

This section discusses CSAC, proposed to address the limitations of ICP described in Section II-B1, as detailed in Algorithm 2. In each iteration of CSAC, four points are randomly selected from the extracted 3D corners, and correspondence matching is performed to estimate the transformation matrix between consecutive 3D datasets.

Although ICP performs iterative optimization to minimize the MSE only in the vicinity of the initial transformation, making it susceptible to local minima, CSAC is designed to avoid this limitation by computing the transformation using representative points that characterize the entire scene of the target point cloud.

Algorithm 2 CSAC

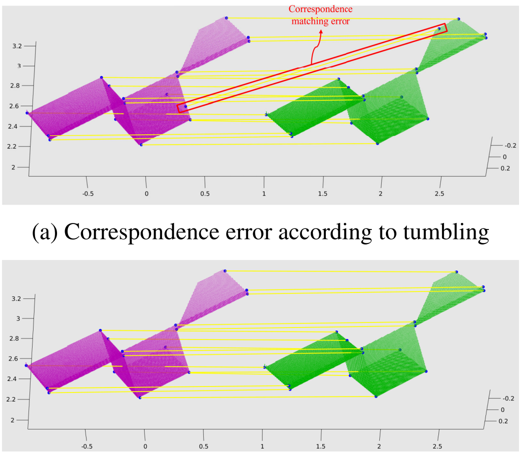
```
Input: Set of 3D corners extracted from consecutive scenes
Estimate correspondences between the 3D corners
Eliminate invalid correspondences using GC
for each unique combination of 4 distinct corners do
    Compute the transformation matrix $T$
    Apply $\boldsymbol{T}$ to the entire point cloud
    Calculate the MSE for the transformed point cloud
end for
Select the optimal transformation matrix $T_{\text{CSAC}}$ that minimizes the MSE
Output: $T_{\text{CSAC}}$
```

The accuracy of CSAC strongly depends on the correctness of the correspondences between 3D corners, but due to the dynamic motion of the target, corners that were previously within the line of sight may become occluded, leading to correspondence errors, as illustrated in Figure 8.

The accuracy of CSAC strongly depends on the correctness of the correspondences between 3D corners, but due to the dynamic motion of the target, corners that were previously within the line of sight may become occluded, leading to correspondence errors, as illustrated in Figure 8a. However, since the input corners for CSAC are spatially well-separated, such errors become more apparent and can be effectively identified and eliminated using a geometric consistency-based method, as shown in Figure 8b. Finally, the transformation matrix T_{CSAC} estimated by CSAC is used as the initial estimate for ICP, completing the two-stage registration process.

3) EKF

The CSAC algorithm is expected to improve pose estimation by successfully providing an initial alignment to ICP. However, if the feature points of the object are occluded and thus not detected, the accuracy of the calculated pose may degrade. To address this issue and improve the accuracy of pose estimation, this section describes a Kalman filtering approach based on the relative kinematics of both position and orientation of the target. For this sake, the two-stage registration algorithm is supplemented with the Extended Kalman Filter (EKF) as illustrated in Figure 9. The prior estimate, denoted as $(\cdot)_k^-$, is propagated through time integration, while the output of CSAC-ICP is fused with the prior estimate $(\cdot)_{k-1}^+$ to obtain the measurement $(\cdot)_m$. This measurement is then incorporated into the EKF measurement update step to compute the posterior estimate in the k-th time step.



(a) Correspondence error according to tumbling

(b) Correction using geometric constraint

FIGURE 8. Corner correspondence error correction using geometric constraint.

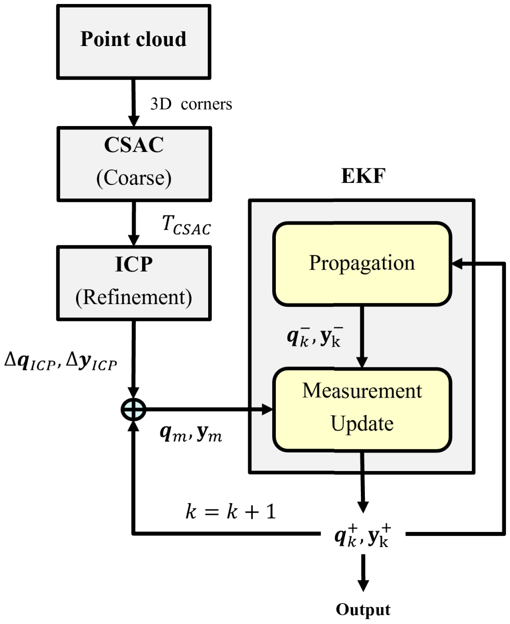

Point cloud
3D corners
CSAC
(Coarse)
T_{CSAC}
ICP
(Refinement)
$\Delta \boldsymbol{q}_{ICP}, \Delta \boldsymbol{y}_{ICP}$
EKF
Propagation
$\boldsymbol{q}_k^-, \mathbf{y}_k^-$
Measurement
Update
$\boldsymbol{q}_m, \boldsymbol{y}_m$
$k = k + 1$
$\boldsymbol{q}_k^+, \mathbf{y}_k^+$
Output

FIGURE 9. Pose tracking process flowchart.

The state vector of the EKF is defined as:

$$\boldsymbol{x}_k = [\boldsymbol{y}_k^T,\ \dot{\boldsymbol{y}}_k^T,\ q_{0,k},\ \boldsymbol{q}_k^T,\ \omega_k^T]^T \tag{9}$$

where $\boldsymbol{y}$ and $\dot{\boldsymbol{y}}$ are the relative position and velocity, respectively. q_0 and $\boldsymbol{q}$ represent the scalar and vector parts of a unit quaternion, respectively, which satisfy the condition of $q_0^2 + |\boldsymbol{q}|^2 = 1$. ω denotes the angular velocity of the target.

The prediction of the state is given by [39]:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{y}_k + \dot{\boldsymbol{y}}_k \Delta t \\ \dot{\boldsymbol{y}}_k \\ q_{0,k+1} \\ \boldsymbol{q}_{k+1} \\ \omega_k \end{bmatrix} \tag{10}$$

where Δt is the time step and $q_{0,k+1}$ and $\boldsymbol{q}_{k+1}$ are computed as:

$$\begin{aligned} q_{0,k+1} &= q_{0,k} \cos\left(\|\omega_k\| \frac{dt}{2}\right) \\ &\quad + \frac{\boldsymbol{q}_k^T \omega_k}{\|\omega_k\|} \sin\left(\|\omega_k\| \frac{dt}{2}\right), \end{aligned} \tag{11}$$

$$\begin{aligned} \boldsymbol{q}_{k+1} &= \boldsymbol{q}_k \cos\left(\|\omega_k\| \frac{dt}{2}\right) \\ &\quad + \frac{1}{\|\omega_k\|}\left(\omega_k^\wedge \boldsymbol{q}_k - q_{0,k}\omega_k\right) \sin\left(\|\omega_k\| \frac{dt}{2}\right), \end{aligned} \tag{12}$$

where the skew-symmetric matrix $\omega_k^\wedge$ is used, which is defined by

$$\omega_k^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{13}$$

Let z_k be the measurement that represents the relative position and orientation of the target, defined by

$$z_k = \left[\boldsymbol{y}_m^T,\ \boldsymbol{q}_m^T\right]^T. \tag{14}$$

Since the ICP only provides information on pose increments, it is assumed that the best full measurement of position and orientation can be obtained from the posterior estimate at each step of the EKF.

$$\begin{aligned} \boldsymbol{y}_m &= \boldsymbol{y}_{k-1}^+ + \Delta \boldsymbol{y}_{ICP} \\ \boldsymbol{q}_m &= \boldsymbol{q}_{k-1}^+ \otimes \Delta \boldsymbol{q}_{ICP} \end{aligned} \tag{15}$$

where the symbol $\otimes$ denotes the Kronecker product [40] for the quaternion multiplication. In this case, the linear observation model simply becomes

$$z_k = H_k \boldsymbol{x}_k \tag{16}$$

where the sensitivity matrix H_k is defined as

$$H_k = \left[\begin{array}{cc|ccc} \boldsymbol{I}_3 & \boldsymbol{0}_3 & 0 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \hline \boldsymbol{0}_3 & \boldsymbol{0}_3 & 0 & \boldsymbol{I}_3 & \boldsymbol{0}_3 \end{array}\right]. \tag{17}$$

Note that $\boldsymbol{I}_3$ is the three-dimensional identity matrix and $\boldsymbol{0}_3$ is the three-dimensional zero matrix. It follows from the standard Kalman filter scheme that the Kalman gain is calculated as follows,

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \tag{18}$$

where P_k is the covariance matrix and R_k is the measurement noise covariance associated with ICP.

Finally, the posterior estimate of the EKF at the current step is obtained by multiplying the Kalman gain with the innovation of the relative position as well as orientation as follows,

$$\begin{aligned} \boldsymbol{y}_k^+ &= \boldsymbol{y}_k^- + K_k\left(\boldsymbol{y}_k^- - \boldsymbol{y}_m\right) \\ \boldsymbol{q}_k^+ &= \boldsymbol{q}_k^- \otimes K_k\left(\boldsymbol{q}_k^- \otimes \boldsymbol{q}_m^{-1}\right). \end{aligned} \tag{19}$$

III. NUMERICAL SIMULATION

A. SIMULATION SETUP

To validate the proposed algorithms, a simulator based on the open-source Robot Operating System (ROS) and Gazebo was developed. Gazebo incorporates a physics engine, enabling the simulation of 6 DoF motion for non-cooperative targets as well as a ToF camera. The proposed algorithm was implemented in Matlab, and communication between Matlab and Gazebo was facilitated via ROS. The simulation applied the following assumptions: The relative position of the target according to the chaser remains constant. The target is considered to be a rigid body. The satellite's shape is assumed to be typical, with two solar panels attached to its base. First, the performance of the feature extraction algorithm was evaluated. The performance of CSAC was then assessed across three cases in consecutive scenes. The target's motion was assumed to be x-axis rotation, y-axis rotation, and tumbling, as shown in Table 1. Each simulation was conducted for 100 seconds, 240 seconds, and 120 seconds, respectively, considering the angular velocity of the target.

TABLE 1. Initial relative parameters of case studies.

Parameter	Case 1	Case 2	Case 3
$\Delta \boldsymbol{y}$ (m)	(0, 6, 0)	(0, 7, 0)	(0, 8, 0)
$\Delta \boldsymbol{q}$ (°)	(0, 0, 0)	(0, 0, 0)	(0, 20, 30)
$\Delta \dot{\boldsymbol{y}}$ (m/s)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
Angular Rate (°/s)	(3.5, 0, 0)	(0, 1.5, 0)	(0, 0, 3)

B. RESULTS

The dashed lines in Figure 10 represent the lines obtained by applying Algorithm 1 to the point cloud with Gaussian noise added, following a normal distribution to account for measurement noise. The white points indicate the 3D corners extracted using the method introduced in Section II-A2. This demonstrates that the feature extraction algorithm proposed in Section II operates robustly against measurement noise.

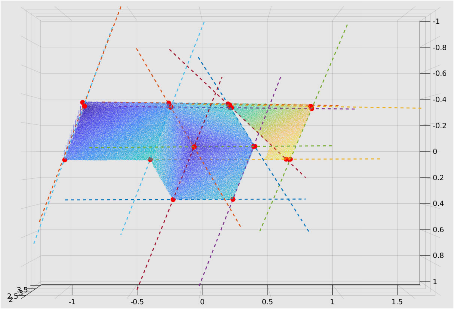
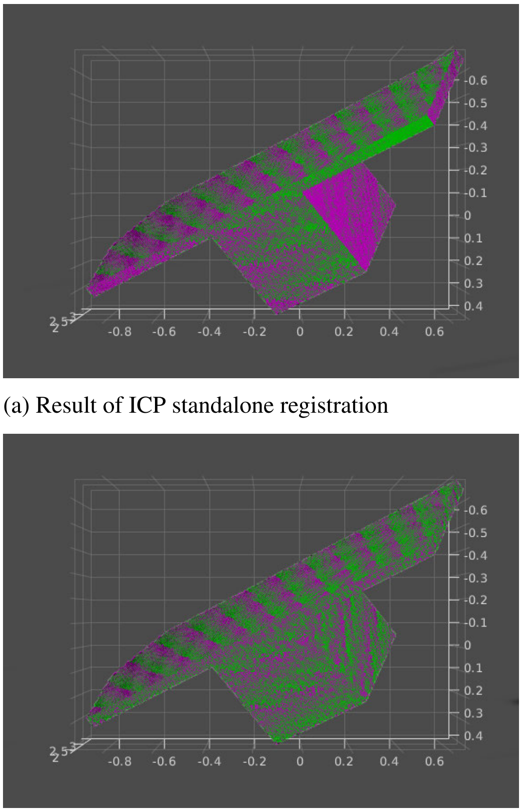


FIGURE 10. Lines and corners extracted using proposed method in this paper.

Figure 11 shows the registration results of the ICP standalone and the proposed CSAC-ICP. The green and pink points represent the consecutive point clouds registered using the ICP standalone and the CSAC-ICP, respectively. The



(a) Result of ICP standalone registration

(b) Result of CSAC-ICP registration

FIGURE 11. Results of the registration in simulation.

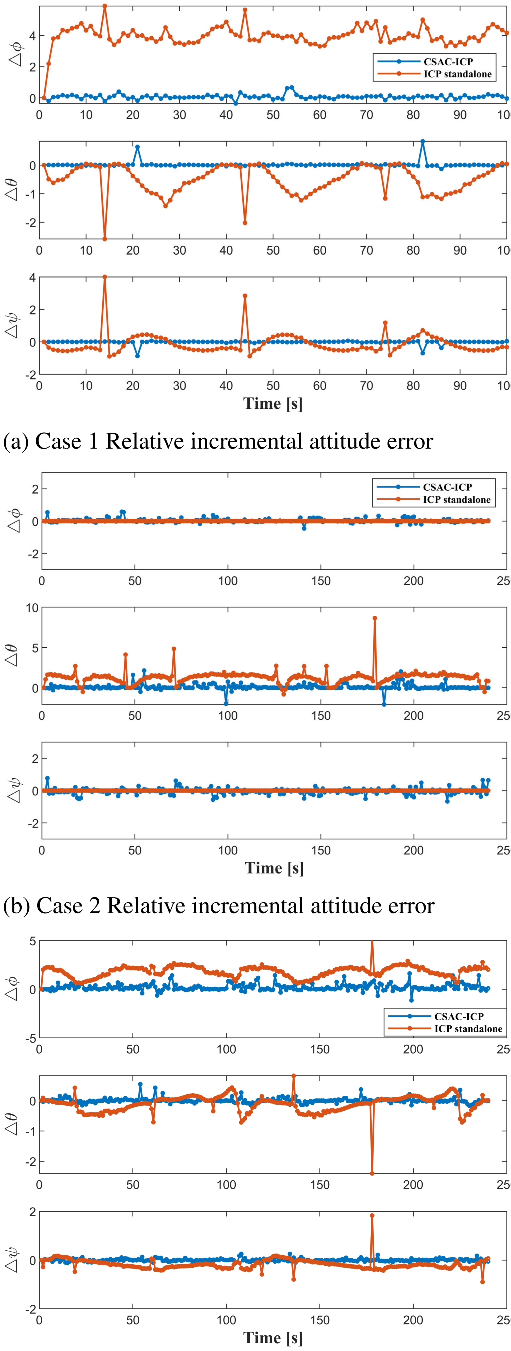
results were obtained by setting the maximum iteration count to fifty for the ICP standalone and five for the CSAC-ICP, demonstrating the higher efficiency of the CSAC-ICP approach. In addition, the average registration time for each method is summarized in Table 2.

TABLE 2. Average registration time.

Method	ICP standalone	CSAC-ICP
Mean time [s]	1.70	0.35

Additionally, the RMSE values for registration were 0.012 and 0.003, respectively, indicating improved accuracy with the CSAC-ICP. This enhancement in accuracy can also be visually confirmed through the overlap shown in Figures 11a and 11b. Additionally, the relative pose increment errors measured by ICP standalone and CSAC-ICP for each case are presented in Figure 12.

The red and blue '-o' markers in Figure 12 represent the pose increment values measured by ICP standalone and CSAC-ICP, respectively. In all cases, it can be observed that the incremental attitude error of ICP closely approximates the target's angular rate, indicating a failure in registration within the given timeframe. In contrast, for CSAC-ICP, which includes feature extraction, successful registration was achieved.



(a) Case 1 Relative incremental attitude error

(b) Case 2 Relative incremental attitude error

(c) Case 3 Relative incremental attitude error

FIGURE 12. Relative position and attitude for each case: (a) Case 1, (b) Case 2, (c) Case 3.

IV. EXPERIMENT

A. SETUP

To evaluate the performance of the proposed pose estimation method, experiments were conducted using the setup illustrated in Figure 13.

The relative translational and rotational motion of a tumbling non-cooperative target in orbit were simulated using a single-axis linear motor and a UFACTORY xArm 6 robotic manipulator, respectively. The chaser satellite was equipped

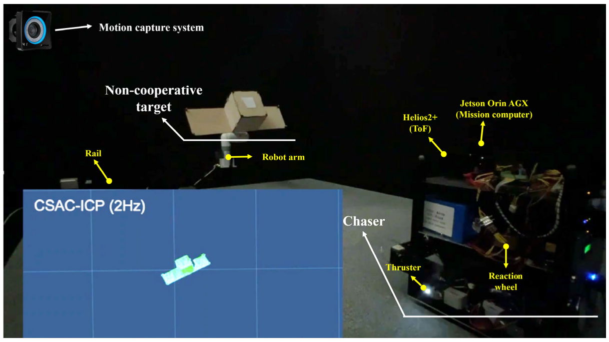
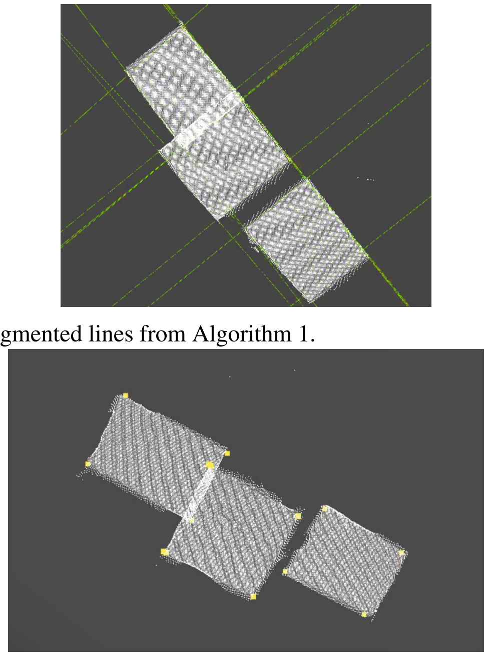


FIGURE 13. The experimental setup consists of the UFACTORY Lite 6, Helios 2+, and a satellite mock-up.

with a Helios 2+ ToF camera from LUCID Vision Labs and a mission computer, NVIDIA Jetson Orin AGX. The target satellite mockup was designed as a cubic structure with solar panels attached to its two lateral sides. The ground truth pose of the target was measured using a motion capture system.

For relative navigation in an independent system, the proposed pose estimation method was developed in C++ to run on the mission computer. The implementation utilized the Point Cloud Library (PCL) [41]. Communication between the ToF camera and the mission computer was established via the ROS.



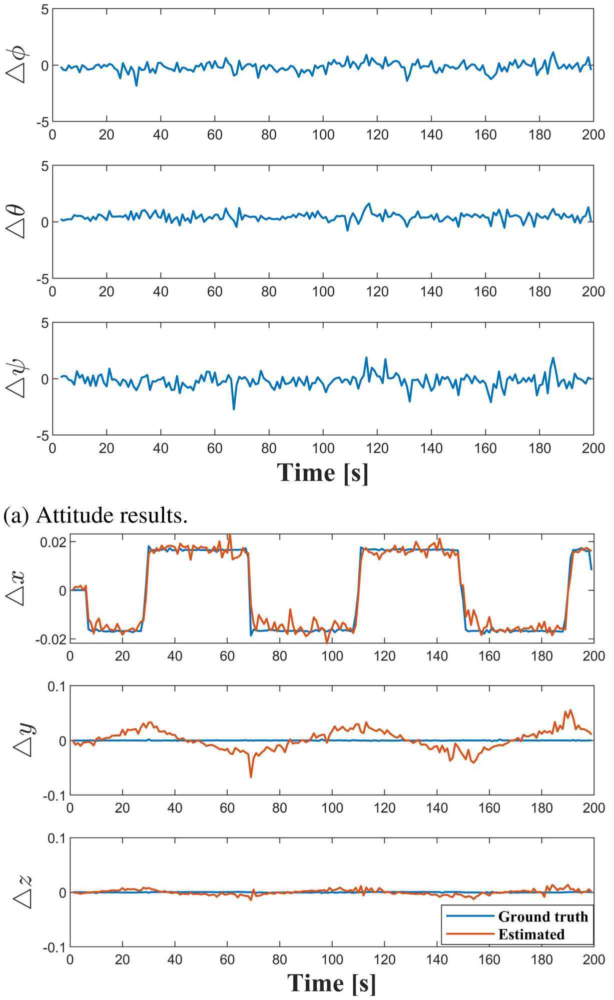
(a) Segmented lines from Algorithm 1.

(b) Extracted 3D corners.

FIGURE 14. (a) Segmented lines obtained using Algorithm 1. (b) 3D corners extracted using the proposed feature extraction method.

B. RESULTS

The results of applying the algorithm proposed in Section II-A to the point cloud acquired using the Helios2+ are presented in Figure 14. The yellow dashed lines in Figure 14a represent the lines generated by applying Algorithm 1, while Figure 14b illustrates the extracted 3D corners.



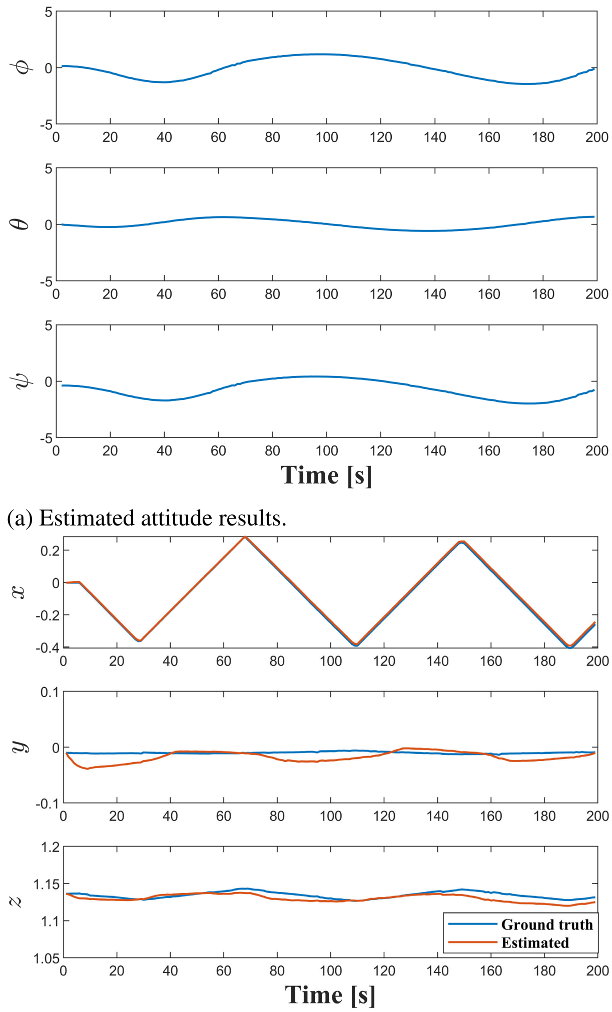
(a) Attitude results.

(b) Position results.

FIGURE 15. Measured incremental relative pose: (a) Attitude results and (b) Position results.

The experimental results confirmed that the average extraction time for 3D corners was 0.028 seconds, which demonstrates the efficiency of the proposed method.

The relative motion of the target mockup with respect to the chaser was simulated using a robotic arm and a linear rail. The experiments were conducted over a total duration of 200 seconds, with a time step of $dt = 0.5$ seconds. During the experiments, the chaser's coordinate frame was assumed to align with that of the ToF camera. The incremental relative pose measured by CSAC-ICP is shown in Figure 15. The errors in the incremental relative attitude are depicted in Figure 15a, with maximum errors of 1.98°, 1.59°, and 3.52° for ϕ, θ, and ψ, respectively, and RMSE values of 0.35°, 0.28°, and 0.6°. The results for the incremental relative position are presented in Figure 15b, where the blue solid line represents the ground truth, and the red solid line represents the measured values. The motion along the x-axis relative to



(a) Estimated attitude results.

(b) Estimated position results.

FIGURE 16. Estimated experimental results: (a) attitude and (b) position.

the chaser was evaluated, with position errors confirmed to be within 0.05 m for each axis.

The relative attitude errors estimated by the CSAC-ICP-EKF are shown in Figure 16a, with errors within 3 degrees for ϕ, θ, and ψ. Figure 16b shows the relative position results, where the blue solid line represents the ground truth, and the red solid line represents the estimated values. As a result, the relative position errors were confirmed to be within 0.02 m for the x, y, and z axes.

V. CONCLUSION

Relative pose estimation of non-cooperative space targets is a critical technology for Autonomous Redenzvous and docking. With recent advancements in 3D sensing, the fundamental limitation of 2D sensors in providing distance information has been effectively addressed. Moreover, in low-illumination mission environments such as umbra and penumbra, passive cameras become inoperable, whereas active cameras are expected to function reliably even under such conditions. However, due to surface reflectance characteristics of satellites, measurements from active sensors may contain noise, necessitating algorithm design that accounts for such disturbances. This study proposes a relative pose estimator using a ToF camera. To this end, a global registration technique named CSAC is developed, and a method for extracting global features from 3D point clouds at 30 FPS through reprocessing is presented. By providing a robust initial transformation, CSAC enables a two-stage registration framework that significantly enhances the performance of ICP. This registration framework is integrated with an EKF to construct the complete pose estimator. Through case studies conducted in a SILS environment and experiments using real sensors and mission computers, it is demonstrated that accurate pose estimation can be achieved without prior knowledge of the target's shape or dynamic information.

REFERENCES

[1] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations," *Prog. Aerosp. Sci.*, vol. 93, pp. 53–72, Aug. 2017.

[2] I. Kawano, M. Mokuno, T. Kasai, and T. Suzuki, "Result of autonomous rendezvous docking experiment of engineering test satellite-VII," *J. Spacecraft Rockets*, vol. 38, no. 1, pp. 105–111, Jan. 2001.

[3] T. Rumford, "Demonstration of autonomous rendezvous technology (DART) project summary," *Proc. SPIE*, vol. 5088, pp. 10–19, Aug. 2003.

[4] R. Friend, "Orbital express program summary and mission overview," *Proc. SPIE*, vol. 6958, pp. 11–21, Apr. 2008.

[5] R. Delage, "An overview of ATV integrated mission analysis and mission preparation," in *Proc. 54th Int. Astron. Congr. Int. Astron. Fed., Int. Acad. Astronaut., Int. Inst. Space Law*, Sep. 2003, pp. 1–7.

[6] F. Aghili, M. Kuryllo, G. Okouneva, and C. English, "Fault-tolerant position/attitude estimation of free-floating space objects using a laser range sensor," *IEEE Sensors J.*, vol. 11, no. 1, pp. 176–185, Jan. 2011.

[7] G. Di Mauro, M. Lawn, and R. Bevilacqua, "Survey on guidance navigation and control requirements for spacecraft formation-flying missions," *J. Guid., Control, Dyn.*, vol. 41, no. 3, pp. 581–602, Mar. 2018.

[8] L. Zhang, D.-M. Wu, and Y. Ren, "Pose measurement for non-cooperative target based on visual information," *IEEE Access*, vol. 7, pp. 106179–106194, 2019.

[9] H. Han, H. Kim, and H. Bang, "Monocular pose estimation of an uncooperative spacecraft using convexity defect features," *Sensors*, vol. 22, no. 21, p. 8541, Nov. 2022.

[10] L. Chang, J. Liu, Z. Chen, J. Bai, and L. Shu, "Stereo vision-based relative position and attitude estimation of non-cooperative spacecraft," *Aerospace*, vol. 8, no. 8, p. 230, Aug. 2021.

[11] K. D. Mallampati, J. Lee, and S. Sung, "VIO algorithm design using integrated spatial feature geometry for precise spacecraft pose determination," *Int. J. Aeronaut. Space Sci.*, vol. 26, no. 5, pp. 2785–2800, Oct. 2025.

[12] Q. Fu, J. Wang, H. Yu, I. Ali, F. Guo, Y. He, and H. Zhang, "PL-VINS: Real-time monocular visual-inertial SLAM with point and line features," 2020, *arXiv:2009.07462*.

[13] D.-R. Lee and H. Pernicka, "Vision-based relative state estimation using the unscented Kalman filter," *Int. J. Aeronaut. Space Sci.*, vol. 12, no. 1, pp. 24–36, Mar. 2011.

[14] Y. Na, H. Bang, and S.-H. Mok, "Vision-based relative navigation using dual quaternion for spacecraft proximity operations," *Int. J. Aeronaut. Space Sci.*, vol. 20, no. 4, pp. 1010–1023, Dec. 2019.

[15] S. Park and D. Jung, "Vision-based tracking of a ground-moving target with UAV," *Int. J. Aeronaut. Space Sci.*, vol. 20, no. 2, pp. 467–482, Jun. 2019.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Inf. Process. Lett.*, vol. 2, no. 1, pp. 18–21, Mar. 1973.

[18] D. Viswanathan, "Features from accelerated segment test (FAST)," in *Proc. 10th Workshop Image Anal. Multimedia Interact. Services*, May 2009, pp. 6–8.

[19] J. Salvi, X. Armangué, and J. Batlle, "A comparative review of camera calibrating methods with accuracy evaluation," *Pattern Recognit.*, vol. 35, no. 7, pp. 1617–1635, Jul. 2002.

[20] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[22] M. D. Shuster, "A survey of attitude representations," *J. Astron. Sci.*, vol. 41, no. 4, pp. 439–517, 1993.

[23] H. Matsumura and C. Premachandra, "Deep-learning-based stair detection using 3D point cloud data for preventing walking accidents of the visually impaired," *IEEE Access*, vol. 10, pp. 56249–56255, 2022.

[24] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Uncooperative pose estimation with a LiDAR-based system," *Acta Astronautica*, vol. 110, pp. 287–297, May 2015.

[25] P. Besl and N. McKay, "Method for registration of 3-D shapes," *Proc. SPIE*, vol. 1611, pp. 586–606, May 1992.

[26] A. Sveier, T. A. Myhre, and O. Egeland, "Pose estimation with dual quaternions and iterative closest point," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 1913–1920.

[27] W. H. Clohessy and R. S. Wiltshire, "Terminal guidance system for satellite rendezvous," *J. Aerosp. Sci.*, vol. 27, no. 9, pp. 653–658, Sep. 1960.

[28] J. O. Woods and J. A. Christian, "LiDAR-based relative navigation with respect to non-cooperative objects," *Acta Astronautica*, vol. 126, pp. 298–311, Sep. 2016.

[29] Y. He, B. Liang, J. He, and S. Li, "Non-cooperative spacecraft pose tracking based on point clouds feature," *Acta Astronautica*, vol. 139, pp. 213–221, Oct. 2017.

[30] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 4164–4169.

[31] Y. He, J. Yang, X. Hou, S. Pang, and J. Chen, "ICP registration with DCA descriptor for 3D point clouds," *Opt. Exp.*, vol. 29, no. 13, pp. 20423–20439, Jun. 2021.

[32] Y. He, J. Yang, K. Xiao, C. Sun, and J. Chen, "Pose tracking of spacecraft based on point cloud DCA features," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5834–5843, Mar. 2022.

[33] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[34] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 1–6.

[35] R. Shenbagavalli, "Classification of soil textures based on laws features extracted from preprocessing images on sequential and random windows," *Bonfring Int. J. Adv. Image Process.*, vol. 1, no. 1, pp. 15–18, Dec. 2011.

[36] S. M. Ahmed, Y. Z. Tan, C. M. Chew, A. A. Mamun, and F. S. Wong, "Edge and corner detection for unorganized 3D point clouds with application to robotic welding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7350–7355.

[37] C. Dalitz, T. Schramke, and M. Jeltsch, "Iterative Hough transform for line detection in 3D point clouds," *Image Process. Line*, vol. 7, pp. 184–196, Jul. 2017.

[38] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 726–740, Jun. 1987.

[39] H. Gómez Martínez, G. Giorgi, and B. Eissfeller, "Pose estimation and tracking of non-cooperative rocket bodies using time-of-flight cameras," *Acta Astronautica*, vol. 139, pp. 165–175, Oct. 2017.

[40] R. Bellman, *Introduction to Matrix Analysis*, 2nd ed., Philadelphia, PA, USA: SIAM, 1997. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611971170

[41] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

SANGDO PARK received the master's degree from Korea Aerospace University (KAU), Goyang-si, South Korea, in 2025. His research interests include the fields of vision-based relative navigation for on-orbit servicing.

MINSIK OH received the B.S. degree in avionics and information engineering from Korea Aerospace University (KAU), Goyang-si, South Korea, in 2024. He is currently pursuing the M.S. degree. His research interests include GNC system integration and open-source space simulation development.

HEOKJUNE YOU received the master's degree from Korea Aerospace University (KAU), Goyang-si, South Korea, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include the fields of space robotics and control.

DONGWON JUNG (Member, IEEE) received the B.S. and M.S. degrees in aerospace engineering from Seoul National University, Republic of Korea, in 1998 and 2000, respectively, and the Ph.D. degree in aerospace engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2007. From 2010 to 2014, he was with Korea Aerospace Research Institute (KARI) as a Senior Researcher with the Division of Korea Space Launch Vehicle-2 Research and Development Program Office. He is currently an Associate Professor with the Department of Smart Drone Convergence, Korea Aerospace University. His research interests include design and development of avionics systems for uncrewed aerial vehicles (UAV), nonlinear guidance and trajectory optimization for autonomous vehicles, path/motion planning for multiple autonomous agents, and hybrid estimation using vision and inertial sensors.